KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Clinical Decision Support

Interpretability and Applications in Patient Monitoring

Lieven Billiet

Supervisors:
Prof. dr. ir. S. Van Huffel
dr. ir. V. Van Belle

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

October 2018

Clinical Decision Support

Interpretability and Applications in Patient Monitoring

Lieven BILLIET

Examination committee:
Prof. dr. A. Bultheel, chair
Prof. dr. ir. S. Van Huffel, supervisor
dr. ir. V. Van Belle, supervisor
Prof. dr. B. Vanwanseele
Prof. dr. R. Westhovens MD
Prof. dr. ir. M. B. Blaschko
Prof. dr. ir. M. Verleysen
(Université catholique de Louvain)
Prof. dr. P. Lisboa
(Liverpool John Moores University)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

October 2018

© 2018 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Lieven Billiet, Kasteelpark Arenberg 10 box 2446, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

All we have to decide is what to do with the time that is given to us.

J.R.R. Tolkien, *The Lord of the Rings*

Preface

Once upon a time, er was eens... Zo beginnen vele verhalen. Die 'eens' is echter vaak nogal ondefinieerbaar, zoals ook in het geval van een doctoraat. Het eindresultaat ervan ligt voor u, maar wat is het begin? Ligt dat in de nieuwsgierigheid die van jongsaf werd gestimuleerd en de vele opgedane indrukken doorheen de jaren? Ligt ze in de plaats en omstandigheden van de verdere studies hier aan de KU Leuven en elders? Of is het eenvoudig een mogelijkheid waarvan gebruikt werd gemaakt toen ze werd aangeboden? Vermoedelijk is het een beetje een combinatie van al die factoren. Feit is dat ik ongeveer vier jaar geleden de kans kreeg om een doctoraat te beginnen hier aan de faculteit.

Daarvoor wil ik in het bijzonder mijn promotor, Sabine Van Huffel, bedanken. Sabine, boven alles, dank je wel voor je vertrouwen, dat vaak groter was dan wat ik zelf kon opbrengen. Ik heb het als een uitzonderlijk genoegen ervaren om begeleid te worden door een promotor die niet enkel geïnteresseerd is in output en technische verwezenlijkingen, maar vooral iemand die zich een missie stelt en zich van daaruit oprecht bekommert om de mens achter de doctorandus. Dat Biomed kon uitgroeien tot wat het nu is en dat dit doctoraat tot een mooi einde kan komen is slechts mogelijk door uw visie en steun.

Om dezelfde reden wil ik ook een dankbaar woord richten tot mijn co-promotor, Vanya Van Belle. Vanya, bedankt voor je gigantische inzet en bereidheid om steeds weer te verifiëren en in discussie te gaan, of het nu over code of tekst ging. Ondanks mijn tegenstribbelen af en toe heb je de kwaliteit van het werk ontegensprekelijk sterk verbeterd. Jouw originele ideeën en gedrevenheid zijn mij tot voorbeeld en inspiratie geweest.

Of course, I would also like to thank the other jury members for their presence and willingness to represent the academic community on this occasion. In particular, I am grateful to professor Vanwanseele and professor Verleysen for their constructive feedback as members of the Supervisory Committee over the last four years. Furthermore, I value the presence of professor Blaschko and professor Westhovens representing the multidisciplinary scope of my work. A

special word of thanks goes to professor Lisboa for joining us from Liverpool. Finally, I thank professor Bultheel for accepting the Faculty's invitation of chairing the defense.

Een groot deel van dit werk heeft betrekking op de doelstellingen van het project rond het 'Sensor-based Platform for the Accurate and Remote monitoring of Kine(ma)tics Linked to E-health' (SPARKLE). Ik neem de vrijheid u te wijzen op dit mooie acroniem, zij het wat moeilijk te memoriseren, dat zowel ingenieus als ietwat geforceerd kan overkomen. Hoe dan ook, mijn dankbaarheid gaat ook uit naar mijn 'sparkling' collaboranten. Thijs, bedankt voor je sterke visie en drijvende kracht. Stefan, bedankt voor de uren van overleg en gezamenlijk Shimmer-bashen. Verder ook mijn dank aan Jeroen, Jonas, professor Dankaerts, dokter de Vlam, professor Geurts, professor Puers en professor Vanden Abeele alsook de andere eerder vermelde SPARKLE-leden.

Enkele vakantiejobs daargelaten is dit doctoraat eveneens mijn eerste lange werkervaring. Dat brengt automatisch de vraag naar de betekenis van werk met zich mee. Hoe past dit in de rest van mijn persoonlijk leven? De sleutel tot een antwoord zit grotendeels in hoe ikzelf, maar ook familie en vrienden ermee omgaan. Daarom wil ik ook graag een woord van dank richten tot mijn ouders, moeke en vake, voor hun stimulans en ondersteuning, vanaf mijn prille opvoeding tot dit punt toe. Jullie hielpen me door te zetten, ook op momenten dat het moeilijk ging. Dat geldt uiteraard ook voor Rusangiza, Ingabire en Machteld, mijn broer en zussen. In de loop van de jaren kwam daar de oprechte interesse van mijn schoonfamilie bij, waarvoor zeker ook hartelijk dank! Het is mooi te merken hoezeer ik bij jullie allen een thuis mag vinden. Het zou me verder te ver leiden om alle betrokken vrienden bij naam te noemen: bezoekjes, vele gesprekken, huwelijken, geboortes, reizen, GCL-bijeenkomsten, koorrepetities en -concerten,... noem maar op. Bedankt voor al jullie eigen manieren van mee op weg gaan!

A second key to understanding the meaning of work as inherent part of life lies undoubtedly with my colleagues. I realize I am exceptionally lucky to be able to have spent these four years with all of you. I benefited a lot from our discussions and I can still be amazed by your knowledge and profound way of thinking. Thanks for the support and new directions in times when I got stuck! Yet, another important thing about biomedians is their ability to talk about very, uhm, *diverse*, topics, as in, Life, the Universe and Everything with additional servings of plant life, religion, gender fluidity, baking and much more. Thank you for your friendship, the parties (less fluo please), karaokes (courtesy to UltraStar DX), football and other sports games (keep up the winning tradition for the next sports days, but for goodness' sake, change the t-shirt), survival runs (none is left behind), house warmings, baby visits (Biomed's future is guaranteed), bar visits (no comments), LAN party (I've built a new computer in the meantime, so less lag), curve fever hackaton (which we won), fabulous

breakfasts (Happiness still around?), laser shooting, coffee stops (with cookies), board and card games (waiting for Ahrr?!), bachelor parties, the escape room (I still insist our room was faulty), publication pancakes (surprisingly rare, not because of the lack of publications though) and countless receptions for any occasion that arose (in search for that famous Ballmer peak, no doubt). So, thank you all: Abi, Adrian, Alex Caicedo, Alex Suarez, Alexander Bertrand, Amalia, Amir, Arun, Bharath, Bori, Carolina, Claudio, Devi, Dorien, Dries, Griet, Jasper, Javier, Jesus, John, Kaat, Kirsten, Laure, Margot, Mario, Martijn, Matthieu, Michiel, Milica, Neetha, Nico, Nicolas, Ninah, Ofelie, Otto, Rob, Siba, Steven, Stijn, Thomas, Tim, Vladimir, Ying, Yissel. In particular, Jonathan and Simon, sharing my office, thank you for putting up with my old-fashioned phrasings ('Zeg niet...') and other eccentricities. Your ideas and presence were and remain invaluable!

Le plus important vient à la fin, comme toujours. Ces dernières années, ma meilleure amie et co-aventurière est devenue de plus en plus l'amour de ma vie, ma fiancée, mon épouse pour le meilleur et pour le pire, pendant un doctorat et après, devant Dieu et tout le monde. Chère Béné, de tout coeur: un grand merci de garder confiance en moi, pour ton soutien, ta compréhension, ta compassion et patience, ton amour incessant. Voilà pourquoi j'aimerais dédier ce doctorat à toi et à notre petit miracle ou petite merveille déjà si présent(e) dans notre foyer. Je ne peux imaginer de plus grande bénédiction que votre présence dans ma vie.

Abstract

Current clinical practice heavily relies on technology to support decisions. In particular, machine learning is ever more employed in decision support systems. This can be attributed to information overload, the fact that it becomes impossible for a clinician to take into account all available information. The drawback of this approach is that such decision support systems are often black boxes, yielding no insight into the reason of their decisions. In medical settings however, where trust and accountability are important issues, such systems should preferably be interpretable. In contrast, other domains almost fully rely on observation or subjective patient-reported questionnaires to quantify a medical situation. This is the case with assessment of physical capacity in patients suffering from chronic musculoskeletal conditions. With the advent of wearable technology this quantification can also be performed in an objective way to yield complementary information. This requires automatic activity recognition and assessment of the more challenging, but informative transitory activities, preferably in the home environment.

With that situation in mind, the research in this manuscript focuses on two themes: interpretable decision support and patient monitoring using wearables. It also connects them by developing interpretable models for activity assessment. A first set of objectives is connected to interpretable classification systems. A second set is linked to the development of activity recognition and assessment algorithms within the scope of the SPARKLE project.

The first part of this work develops and improves two algorithms to extract interpretable medical scoring systems for binary classification from data. Jointly, they are called Interval Coded Scoring (ICS). The resulting models are piecewise constant over the intervals of selected variables. User interaction is possible at several points to steer the training process with expert knowledge and make a desired trade-off between performance and model simplicity.
A first existing approach, `lpICS`, is extended to support interactions. It consists of a total variation regularized classification problem cast in the framework of Support Vector Machines, solved via Linear Programming. However, its

time complexity and its uniqueness properties can be improved. Therefore, preselection is introduced as an interval-informed way to decrease the data dimensionality. Also, a second approach, `enICS`, is developed. It is based on elastic net cast as a dual Support Vector Machine and can be solved independently of the data dimensionality. However, it does not support variable interactions. Both algorithms are validated on several publicly available biomedical datasets, proving a performance that is similar to several standard machine learning algorithms.
ICS is implemented in a Matlab toolbox offering an interactive interface for model setup, training, risk estimation and visualization.

The second part designs and implements two pattern-based approaches for activity recognition using data from a single accelerometer mounted on the upper arm. They are both based on the assumption that pattern matching is more suitable for recognition of transitory activities, whereas sliding windows with statistical features are more prevalent in literature for repetitive activities of longer duration. One approach improves pattern matching recognition via the combination of dynamic time warping and common statistical features. It consists of an automatic segmentation approach followed by a multilayer random forest classification framework to reject false positives belonging to the rejection class. The other approach exploits the multilinear structure of the data. Higher Order Discriminant Analysis is applied on pattern-matched data to extract discriminatory features from the automatically segmented data. Both methods have been validated on acquired patient datasets in a protocol setting suitable for a home environment. They are proven to significantly outperform approaches based on either simple pattern matching or statistical features alone.
Moreover, the second part also studies assessment of activity recognition. Two informative parameter extraction methods are developed. Furthermore, three different ways to implement an objective complement to the BASFI questionnaire, via classification with ICS or regression, are presented. To end, ICS is applied to create a scoring system for comparison with the current standard in evaluation of disease activity.
Finally, the work describes a Matlab and Python toolbox for data visualization and annotation.

Beknopte samenvatting

De huidige medische praktijk maakt steeds meer gebruik van technologie bij het nemen van beslissingen. Zo is er bijvoorbeeld een sterke opkomst van beslissingsondersteuningssystemen gebaseerd op automatisch leren, een onderdeel van kunstmatige intelligentie. Dit heeft ondermeer te maken met wat men informatieovervloed noemt, namelijk dat de arts de hoeveelheid beschikbare informatie niet meer kan overzien. Deze evolutie heeft als nadeel dat de gebruikte systemen vaak niet interpreteerbaar zijn. Dat is nochtans vaak een belangrijke factor bij medische beslissingen omwille van vertrouwen en mogelijke aansprakelijkheid. Daar staat tegenover dat andere domeinen nauwelijks gebruik maken van technologie, maar in plaats daarvan volledig steunen op observatie en subjectieve patiëntgerapporteerde vragenlijsten. Dit geldt bijvoorbeeld voor het inschatten van de functionele capaciteit bij patiënten die lijden aan chronische musculoskeletaire aandoeningen. De komst van draagbare sensors laat toe om hier een objectieve complementaire techniek voor te ontwikkelen. Zo'n aanpak vereist automatische herkenning en beoordeling van de uitdagende maar zeer informatieve overgangsactiviteiten, bij voorkeur in de thuisomgeving.

Vertrekkend van deze situatie focust dit manuscript op deze twee thema's: interpreteerbare beslissingsondersteuning en patiëntopvolging met behulp van draagbare sensors. Ze verbindt die ook door het ontwerp van interpreteerbare systemen voor activiteitsbeoordeling. Een eerste deel van het manuscript handelt daarom over interpreteerbare classificatiesystemen. Het tweede deel richt zich op de ontwikkeling van automatische activiteitsherkennings- en beoordelingssystemen.

Het eerste deel ontwerpt en verbetert twee algoritmes om medische scoresystemen voor binaire classificatie te trainen uit beschikbare data. Gezamenlijk vallen ze binnen het raamwerk van Interval-geCodeerde Systemen (ICS). ICS-modellen zijn stuksgewijs constant over het bereik van de geselecteerde datavariabelen. Een gebruiker kan op verschillende punten in het trainingsproces tussenkomen om expertkennis toe te voegen of een afweging te maken tussen de eenvoud en prestatie van een model.

Het eerste algoritme is `lpICS`. Het bestaande raamwerk wordt uitgebreid met ondersteuning voor interacties tussen variabelen. De aanpak is gebaseerd op Support Vector Machines, maar past die aan met regularizatie van de totale variantie en verkrijgt de oplossing via Lineaire Programmering. De tijdscomplexiteit en uniciteit kunnen echter nog verbeterd worden. Daarom wordt preselectie aangebracht als een manier om de dimensionaliteit van het probleem te verlagen. Daarnaast ontwikkelt dit werk een tweede ICS-methode, `enICS`, geïnspireerd op een elastisch-netprobleem dat wordt opgelost als de duale formulering van een Support Vector Machine. Daardoor is de complexiteit van het probleem onafhankelijk van de dimensionaliteit wat tot een grote verbetering in uitvoeringssnelheid leidt. Daar staat tegenover dat interacties niet ondersteund worden. Beide ICS-algoritmes worden geëvalueerd op publieke biomedische datasets. De vergelijking bewijst dat de interpreteerbaarheid geen nadelige gevolgen heeft voor de kwaliteit van de voorspelling.
ICS is geïmplementeerd in een Matlabpakket met een interactieve interface voor probleemformulering, training, risicobepaling en visualisatie.

Het tweede deel ontwerpt en implementeert twee patroongebaseerde methodes voor activiteitsherkenning gebruik makend van data van een enkele versnellingssensor op de bovenarm. Beide vertrekken van de hypothese dat patroonherkenning verkiesbaar is voor de herkenning van overgangsactiviteiten. Dit contrasteert met repetitieve activiteiten die in de literatuur vaak via glijdende vensters en statistische kenmerken herkend worden. Een eerste techniek verbetert de standaardmethode voor patroonherkenning door dynamische tijdsvervorming te combineren met de gebruikelijke statistische kenmerken. Eerst dient de data automatisch gesegmenteerd te worden, waarna een meerlaags raamwerk met een random forest verkeerdelijk gedetecteerde activiteiten verwerpt en de rest tracht in te delen in de juiste klasse. De tweede techniek maakt gebruik van de multilineaire structuur van de data als extra informatie. Het toepassen van Hogere-Orde-Discriminantanalyse op patroonvervormde data maakt het mogelijk om activiteitsonderscheidende kenmerken te verkrijgen. Validatie van beide methodes in een protocolsituatie toont aan dat ze significant betere voorspellingen leveren dan alternatieven die enkel eenvoudiger patroonherkenning of statistische kenmerken aanwenden. Bovendien is deze protocolstructuur perfect verplaatsbaar naar de thuisomgeving.
Verder bestudeert het tweede deel van het manuscript ook de beoordeling van activiteiten. Daarvoor worden eerst twee verzamelingen van informatieve parameters ontwikkeld. Daarnaast worden ook drie verschillende manieren onderzocht om een objectief alternatief of complement te genereren voor de BASFI-vragenlijst. Dit kan zowel via classificatie met ICS als via regressie. Bovendien wordt ICS ook toegepast voor het aanmaken van een scoresysteem ter vergelijking met de huidige standaard voor de evaluatie van ziekteactiviteit. Tot slot beschrijft het werk ook een Matlab- en Pythonpakket voor datavisualisatie en -annotering.

List of Abbreviations

Acc	Accuracy
Acc_a	Actual Accuracy
Acc_p	Pure (closed world) Accuracy
AI	Artificial Intelligence
AS	Ankylosing Spondylitis
ASAS	Assessment of SpondyloArthritis international Society
ASDAS	AS Disease Activity Score
AUC	Area Under the ROC Curve
axSpA	axial Spondyloarthritis
BASDAI	Bath AS Disease Activity Index
BASFI	Bath AS Functional Index
CDSS	Clinical Decision Support System
cMSC	chronic Musculoskeletal Condition
CPD	Canonical Polyadic Decomposition
CRP	C-Reactive Protein
CV	Cross-validation
CWT	Continuous Wavelet Transform
DT	Decision Tree
DTPR	Detection True Positive Rate
DTW	Dynamic Time Warping
ECG	Electrocardiogram
EHR	Electronic Health Record
EMG	Electromyography
EN	Elastic Net
ESR	Erythrocyte Sedimentation rate
`enICS`	elastic net Interval Coded Scoring
FACIT	Functional Assessment of Chronic Illness Therapy

FD	False Detections
fMRI	functional Magnetic Resonance Imaging
GDP	Gross Domestic Product
GPS	Global Positioning System
HLA-B27	Human Leucocyte Antigen B27
HMM	Hidden Markov Model
HODA	Higher Order Discriminant Analysis
iBASFI	instrumented BASFI
ICS	Interval Coded Scoring
ICS-	ICS without interactions
IMU	Inertial Measurement Unit
kNN	k Nearest Neighbours
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
LoSo	Leave one Subject out (cross-validation)
LP	Linear Programming
`lpICS`	linear programming Interval Coded Scoring
LS-SVM	Least Squares-Support Vector Machine
MAE	Mean Absolute Error
MASES	Maastricht AS Enthesitis Score
MET	Metabolic Equivalent
MLSVD	Multilinear Singular Value Decomposition
MRI	Magnetic Resonance Imaging
NA	Not Applicable
NB	Naive Bayes
NFEA	Novel toolbox for Feature Extraction and Applications
NN	Neural Network
NSAID	Non-Steroidal Anti-Inflammatory Drug
PCA	Principal Component Analysis
PGA	Patient Global Assessment
QP	Quadratic Programming
RBF	Radial Basis Function
RF	Random Forest
RGB	Red, Green, Blue (colour space)

RMSE	Root Mean Squared Error
ROC	Receiver Operator Characteristic
stB	standardized Beta
SDC	Sørensen-Dice Coefficient
SJC	Swollen Joint Count
SNR	Signal-to-Noise Ratio
SpA	Spondyloarthritis
SPARKLE	Sensor-based Platform for the Accurate and Remote monitoring of Kine(ma)tics Linked to E-health
STDAE	STandard Deviation of Absolute Error
SVD	Singular Value Decomposition
SVEN	Support Vector Elastic Net
SVM	Support Vector Machine
TJC	Tender Joint Count
TNF-α	Tumor Necrosis Factor α
TUG	Timed Up and Go
XOR	Exclusive Or

List of Symbols

General notation conventions

a, b, c, ...	Scalars
i, p, k, l, ...	Index variables
x	Variable
x^p	p^{th} variable
$\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, ...	Vectors
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, ...	Matrices
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, ...	Tensors
A, B, C, ...	Sets
N, M, L	Numbers, constant values
a_i	i^{th} element of vector $\mathbf{a}$
$\mathbf{a}^i$	i^{th} column of matrix $\mathbf{A}$
$\mathbf{a}_i$	i^{th} row of matrix $\mathbf{A}$
$\mathbb{R}$	Set of real numbers
$\mathbb{B}$	Set of binary numbers $\{0, 1\}$
$\mathbb{R}^N$	Set of N-dimensional real vectors
$\mathbb{B}^{N_1 \times N_2}$	Set of N_1 by N_2 binary matrices
ϵ, ϵ_0, ϵ_1, ...	Small values
γ, λ	Hyperparameters
$\varphi(\cdot)$	Data transformation function
θ	Angle
$\#$	Cardinality
$\otimes$	Outer (Kronecker) product operator

Specific symbols

$\mathbf{D}$	Matrix defining differences between features
$\boldsymbol{\varepsilon}$	Vector of slack variables
ε_i	Slack for the ith constraint
$\mathbf{f}$	Vector of absolute coefficient differences $\lvert\mathbf{Dw}\rvert$ in ICS
$\varphi^p(\cdot)$	Data transformation for the pth variable
$\varphi^{p_1,p_2}(\cdot)$	Data transformation for the interaction (x^{p_1}, x^{p_2})
$I(\cdot)$	Indicator function, $I(\texttt{true}) = 1$, 0 otherwise
$\mathcal{N}(\mu, \sigma)$	Normal distribution with mean μ, standard deviation σ
N_c	Number of categories for a categorical variable
N_d	Dimensionality of a dataset
N_{ds}	Number of defined differences via $\mathbf{D}$
N_r	Number of intervals for a variable
N_r^p	Number of intervals for the pth variable
N_s	Dimensionality of the expanded feature space
R	Rank of a matrix or tensor
R^2	R squared metric
R^2_{adj}	Adjusted R squared metric
σ_n^2	Noise variance
σ_s^2	Signal variance
$\mathbf{s}$	Vector of integer score points, outcome of ICS
$\boldsymbol{\tau}$	Thresholds for defining intervals on variables
$\boldsymbol{\tau}^p$	Thresholds for the pth variable
τ_k^p	k^{th} threshold for the pth variable
$\mathbf{w}$	Coefficient vector
$\boldsymbol{\omega}$	Coefficient vector in ICS preselection
$\mathbf{w}^p$, $\boldsymbol{\omega}^p$	Coefficients for the pth variable
$\mathbf{w}^{p_1,p_2}$, $\boldsymbol{\omega}^{p_1,p_2}$	Coefficients for the interaction (x^{p_1}, x^{p_2})
$\boldsymbol{\Sigma}$	Covariance matrix
$\mathbf{X}$	Dataset with observations as rows
$\boldsymbol{\chi}$	Diagonal matrix of weights
$\mathbf{y}$	Labels for dataset $\mathbf{X}$
$\mathbf{Z}$	Binary feature matrix
$\mathbf{Z}^p$	Part of $\mathbf{Z}$ corresponding to variable x^p
$\mathbf{Z}^{p_1,p_2}$	Part of $\mathbf{Z}$ corresponding to interaction (x^{p_1}, x^{p_2})
$\tilde{\mathbf{Z}}$, $\tilde{\mathbf{w}}$, $\tilde{\boldsymbol{\chi}}$	As defined above, but specific to `lpICS`
$\hat{\mathbf{Z}}$, $\hat{\mathbf{w}}$, $\hat{\boldsymbol{\chi}}$	As defined above, but specific to `enICS`

Contents

Abstract v

Beknopte samenvatting vii

List of Abbreviations ix

List of Symbols xiii

Contents xv

List of Figures xxiii

List of Tables xxvii

1 Introduction 1

1.1 Problem statement 1

1.2 The SPARKLE project 3

1.3 Research objectives 5

1.4 Outline of the thesis 6

1.5 A note on notation 8

I Interval Coded Scoring Systems 11

2 Decision support 13

2.1 Introduction . . . 13
2.2 Data-based decision support . . . 14
2.2.1 Regression techniques . . . 14
2.2.2 Classification techniques . . . 15
2.2.3 Pros and cons . . . 16
2.3 Knowledge-based decision support . . . 17
2.3.1 Pros and cons . . . 18
2.4 Data-driven interpretability . . . 18
2.5 Datasets . . . 19

3 (Semi-)automatic extraction of interpretable scoring systems 21

3.1 Rationale and general approach . . . 22
3.2 The Interval Coded Scoring algorithm . . . 23
3.2.1 Data transformation . . . 24
3.2.2 The `lpICS` core algorithm . . . 25
3.2.3 Iterative reweighting . . . 28
3.2.4 Overview of the ICS algorithm . . . 29
3.2.5 Visualization and assessment . . . 31
3.3 Sensitivity study . . . 34
3.3.1 Resampling . . . 34
3.3.2 Influence of noise . . . 35
3.3.3 Influence of non-informative variables . . . 36
3.3.4 Influence of training set size . . . 37
3.3.5 Execution time . . . 38
3.4 Applications . . . 39

3.4.1 Results 39
3.5 Discussion 43
3.6 Conclusion 45

4 Extensions for larger problems **47**
4.1 Methods 47
4.1.1 Preselection 47
4.1.2 Uniqueness: a quadratic formulation 49
4.1.3 Uniqueness and speed: elastic net formulation 51
4.2 Experiments and applications 57
4.2.1 Execution time 57
4.2.2 Model comparison 58
4.2.3 Comparison with standard techniques 60
4.3 Discussion 65
4.4 Conclusion 69

II Activity recognition and assessment from wearable sensors **71**

5 Activity monitoring **73**
5.1 Chronic musculoskeletal conditions 73
5.1.1 Cause and symptoms of axial spondyloarthritis 73
5.1.2 Treatment 74
5.1.3 A functional approach 76
5.2 Activity recognition 76
5.2.1 Wearable sensors 77
5.2.2 Sensor setup 79
5.2.3 Activity recognition methods 80
5.3 Activity assessment 85

5.4 Linear discriminant analysis and performance metrics 87
5.4.1 Linear Discriminant Analysis 87
5.4.2 Performance metrics 88
5.5 Datasets 89
5.5.1 SenseWear protocol data 90
5.5.2 Shimmer protocol data 92
5.5.3 Motor control parameter data 93
5.5.4 Timed Up-and-Go data 95

6 Activity recognition from statistical features complemented with patterns 97

6.1 Features and methods 97
6.1.1 Statistical features 98
6.1.2 Pattern-derived features 99
6.1.3 Segmentation 100
6.1.4 Algorithms 101
6.2 Results 105
6.2.1 Patterns and features in a closed world 105
6.2.2 Patterns and features with segmentation 108
6.3 Discussion 110
6.4 Conclusion 112

7 Activity recognition using patterns 115

7.1 Introduction: a simple step detector 115
7.1.1 Matching via correlation 116
7.1.2 Scalability 117
7.1.3 Limitations 119
7.2 Advanced methods 121
7.2.1 Exploiting data structure 121

7.2.2 Algorithms and experiments 126
7.3 Results 133
7.3.1 Recognition 133
7.3.2 Segmentation 134
7.3.3 Recognition and segmentation 135
7.4 Discussion 140
7.5 Conclusion 142

8 Activity assessment 145
8.1 Parameter extraction 146
8.1.1 Motor control parameters 146
8.1.2 Timed Up and Go 148
8.2 instrumented BASFI (iBASFI) 152
8.2.1 A stepwise approach 152
8.2.2 Modelling with elastic net 154
8.2.3 Approximation as classification 160
8.3 Example: a scoring system for patient assessment 163
8.3.1 Data and methods 165
8.3.2 Results and discussion 166
8.4 Conclusion 168

9 Conclusions and future work 169
9.1 Overview 169
9.1.1 Interpretable classification 170
9.1.2 Activity recognition 171
9.1.3 Activity assessment 173
9.2 Contributions 174
9.3 Future research perspectives 175

9.3.1 Interval Coded Scoring 175
9.3.2 Activity recognition and assessment from wearable sensors 177
9.3.3 Other considerations 178

Appendices **180**

A Interval Coded Scoring Toolbox **181**

A.1 Setup 181
A.1.1 Prerequisites 181
A.1.2 Installation 182
A.2 Setting up a problem 182
A.3 Automatic or Interactive ICS 184
A.4 Preselection 185
A.5 Visualization and Assessment 186
A.5.1 Visualization 186
A.5.2 Assessment 187
A.6 Applying a model on new data 189
A.7 Demos 189
A.7.1 Fisher's Iris dataset 189
A.7.2 Ionosphere dataset 190
A.7.3 Synthetic dataset 191
A.8 Implementation details 191
A.8.1 Top-level functions 192
A.8.2 Visualization functions 192
A.8.3 Core functions 193
A.8.4 Problem parameters 194

B Bath Ankylosing Spondylitis Functional Index **195**

C SegmentLab toolbox **197**

C.1 A brief overview of the interface 197

C.2 Data representation 199

C.2.1 Data conversion 200

C.3 Visualization 203

C.4 Test annotations 205

C.5 Extensions 206

C.5.1 Video synchronization 206

C.5.2 Activity recognition 208

C.6 Python implementation 210

C.6.1 Package requirements 210

C.6.2 Loading, saving and conversion 210

C.6.3 Signals and annotations 212

Bibliography **215**

Curriculum **233**

List of publications **235**

List of Figures

1.1 The SPARKLE project flow 4

1.2 Overview of the thesis structure. 9

3.1 ICS model simplification 27

3.2 Application of ICS on a synthetic data set 32

3.3 ICS execution time 39

4.1 Illustration of the data transformation for `enICS` 54

4.2 Execution time of `lpICS` vs `enICS` 58

4.3 Multiple comparison of ICS AUC with established algorithms. . 63

4.4 Multiple comparison of ICS calibration error with established algorithms. 65

5.1 Flowchart for the diagnosis of spondyloarthritis 75

5.2 Example of a wearable single-sensor setup 78

5.3 Dynamic Time Warping: alignment of two sequences. 84

5.4 An example of Dynamic Time Warping. 85

5.5 Example of a mounted sensor and a patient resp. sitting down and picking up a pen 91

5.6 Positioning of the ViMove units. 94

6.1 Flowchart for the closed world algorithms. 102

6.2 Flowchart of the full activity recognition approach including segmentation. 103

7.1 The averaged step template. 116
7.2 Detail of the step detection results 117
7.3 Correlation step matching of the first 500 seconds. 118
7.4 Step detection using the Continuous Wavelet Transform 120
7.5 An example of the Canonical Polyadic Decomposition 123
7.6 An example of a Tucker decomposition 124
7.7 Graphical summary of the HODA procedure 125
7.8 Overview of the data representation 129
7.9 Classification accuracy for each patient using HODA features. . 133
7.10 A comparison of the classification accuracy using LDA on simple and HODA features. 134
7.11 Segmentation performance for all patients. 135

8.1 Examples of the computation of the motor control parameters. 147
8.2 An example of segmented TUG trials of a patient. 149
8.3 An example of the outcome of the peakfinding algorithm for a patient TUG trial. 150
8.4 Elastic net traces showing the influence of λ on the RMSE and the number of variables. 156
8.5 The 25 most important features based on their selection as a function of λ. 157
8.6 Bland-Altman plots for the best duration model, a simple feature model and the optimal feature model. 158
8.7 Grayscale representation of the absolute correlation coefficient between all selected variables in the optimal model. 159
8.8 ICS model for the classification approach of iBASFI. 163
8.9 A comparison of objective ICS and subjective BASFI. 164

A.1 The interactive weight selection interace 184
A.2 The preselection interface 186
A.3 Visualization interface for ICS models 187
A.4 The model assessment interface. 188

C.1 The starting window of segmentLab. 198
C.2 Interface to convert Excel or text files to segmentLab-readable Matlab files. 200
C.3 Interface to describe a file layout configuration. 201
C.4 The signal visibility panel, the scroll settings panel and the (test) visualization panel. 203
C.5 The interface to edit signal names and bindings. 204
C.6 The segmentLab panel to segment and label activities. 206
C.7 A fully functional example of segmentLab including video synchronization and automatically recognized activities. 207
C.8 The Python interface to import Excel or text files. 211
C.9 The Python Ticks/Test definition panel. 212
C.10 The full Python segmentLab interface. 213

List of Tables

2.1 The CHA_2DS_2-VASc score 17

2.2 The CHA_2DS_2-VASc score risk estimation 18

3.1 Performance assessment of the synthetic data model. 33

3.2 Resampling results for ICS with (ICS) and without (ICS-) interactions as ICS-|ICS. 35

3.3 Influence of noise on the detected effects and test AUC for ICS. 36

3.4 Selected effects and test AUC for an increasing number of additional variables. 37

3.5 Influence of the training set size on the detected effects and the test AUC for ICS. 37

3.6 Rules for the mushroom data set leading to perfect classification 40

3.7 The effect of `lpICS` reweighting 41

3.8 Results of the application of ICS on biomedical datasets 42

4.1 Test result of ICS approaches on the arrhythmia database. . . . 59

4.2 Selected effects of ICS approaches on the arrhythmia database. 60

4.3 Comparison of ICS test performance and complexity with established algorithms on biomedical datasets. 61

4.4 Average ranks of the classifier AUC values over all datasets. . . 63

4.5 Comparison of ICS risk assessment with established algorithms on biomedical datasets. 64

4.6 Average ranks of the classifier calibration errors over all datasets. 65

5.1 Demographic, anthropometric and disease-specific measures for the SenseWear patient population. 90

5.2 List of Sensewear setup activities and their abbreviations 91

5.3 Demographic, anthropometric and disease-specific measures for the Shimmer patient population. 92

5.4 List of Shimmer setup activities and their abbreviations 93

5.5 Activities performed in the motor control parameter study. . . 94

5.6 Demographic and anthropometric measures for the TUG subject population. 95

6.1 LoSo accuracies of linSt, purePat, linPat and linPatSt for all patients. 106

6.2 Pooled confusion matrix over all patients for linPatSt. 107

6.3 LoSo recognition performance including segmentation. 109

6.4 Per-activity segmentation performance 110

7.1 Overview of the six activities performed in the experimental protocol 127

7.2 Confusion matrix of the classified activities using HODA features pooled over all 39 patients. 134

7.3 Results for the combination of segmentation and classification. 136

7.3 Continued from previous page. 137

7.4 Performance for the combined segmentation and classification, grouped by activity. 138

7.5 A summary of the performance difference between SenseWear and Shimmer datasets. 139

8.1 Mean and standard deviation of ground truth duration, number of steps, mean absolute error and standard deviation in duration and number of steps for patients and control subjects performing TUG. 151

8.2 Results of the stepwise feature selection for all activities. 154

8.3 Disease activity-related variables used in the ASDAS models. . 164

8.4 Disease activity variables used for the construction of an ICS ASDAS equivalent. 165

8.5 10-fold cross-validation results for disease activity classification with `lpICS`. 167

8.6 10-fold cross-validation results for disease activity classification with `enICS`. 167

A.1 ICS performance indicators. 188

> Little by little, though, the road was opening up before him and he was moving along it, wisely ignorant.
>
> J. Nadal sj, *Dialogi pro Societate*

Chapter 1

Introduction

1.1 Problem statement

Chronic musculoskeletal conditions (cMSCs) have been identified as a major burden on society with important indirect costs. This raises concerns for a further increase due to ageing populations and changes in lifestyle. One of these cMSCs, low back pain, is highly prevalent, affecting 4 to 33% of the population, influencing their labour ability and well-being. Many cases of low back pain are never fully resolved, but about 10% of diagnoses are specific, that is, with a suspected pathological cause [174]. Spondyloarthritis (SpA) is such a pathology, relatively rare, but well-researched.

Although disease activity can be quantified with conventional means, functional capacity is often assessed subjectively. For example, with patient-reported questionnaires such as the Bath Ankylosis Spondylitis Functional Index (BASFI). Performance-based tests offer an objective alternative or complement, but they often depend on a clinical observer.

With the advent of wearable technology, automatic objective functional assessment becomes feasible, even in a home environment. Yet, before activities can be assessed, they should be recognized. Hence, it requires the implementation of activity recognition and assessment algorithms. Many recognition approaches are already available. In general, they can be divided in sliding window approaches and pattern matching [110]. Sliding windows extract statistical parameters from a fixed signal window [17]. This is advanced over the signal to detect different activities. It is mostly used for repetitive activities [89]. In contrast, pattern matching uses an extracted training pattern and matches this to the signal to detect occurrences of an activity. This captures transitory

behaviour such as sit-stand transitions [58]. Yet, the challenge of capturing and assessing transitory behaviour with a pattern remains open, particularly when taking into account the high variability due to the disease [4, 72]. The SPARKLE project was set up with the problem of activity recognition and assessment in mind, as will be described in the next Section.

Activity assessment fits into the general scope of decision support. Over the last decades, the impact of technology on our lives has increased, particularly in healthcare. New techniques have become available: new devices for imaging or monitoring, new analyses to extract knowledge from measurements, new efficient ways to structure and combine data. Above all, the amount of available patient data has grown exponentially in the medical field [117, 145].

This data explosion creates the problem that even a trained observer cannot process all available information [152]. Or, as with SpA, that the information cannot easily be quantified objectively. Hence, clinical decision support systems (CDSS) have been developed. They exploit the additional information available in the data via existing and new modelling techniques. Many stem from the machine learning community, leveraged by the increasingly available computational power. Typical examples of (potential) CDSS include automatic tumour delineation and classification, expert systems to prescribe drugs taking into account a patient's medical history, personalized diagnosis or prognosis systems or even population-wide analysis of the genome to quantify a person's risk for several genetically-related afflictions.

Yet, in a medical context, as in the given examples, decisions can have far-reaching consequences on both health and well-being. A a result, clinical practice remains hesitant towards adopting the technology until it has exceedingly proven to be superior to existing established approaches. This hesitation stems from the fact that many CDSS are black box systems. They make it hard to understand how a certain decision has been made, whether the knowledge base is trustworthy and what criteria or trade-offs have been considered [148]. Therefore, alternative interpretable approaches should be developed.

However, this is not only a matter of trust or accountability. Ultimately, the aim is to model an underlying process. Understanding the model and being able to match it to what one knows of the modelled process or disease also aims to increase the model's validity, that is, it ensures its output indeed quantifies the process under observation.

1.2 The SPARKLE project

The Sensor-Based Platform for the Accurate and Remote monitoring of Kine(ma)tics Linked to E-health (SPARKLE) is an interdisciplinary project addressing the problem of monitoring patients suffering from cMSCs. The research described in this manuscript aims to contribute to its goals. It is composed of the following partners:

- University Hospitals Leuven, Division of Rheumatology. Prof. R. Westhovens, dr. K. de Vlam, dr. T. Swinnen.
- KU Leuven, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics. Prof. S. Van Huffel, L. Billiet.
- KU Leuven, Department of Electrical Engineering (ESAT), Division of Microelectronics and Sensors (MICAS). Prof. R. Puers, J. Lecoutere, M. Baijot.
- KU Leuven, Faculty of Movement and Rehabilitation Sciences (FaBeR), Biomechanics Laboratory (BIOMECH). Prof. B. Vanwanseele, S. Seerden.
- KU Leuven, Faculty of Movement and Rehabilitation Sciences, Division of Musculoskeletal Rehabilitation (REHAB). Prof. W. Dankaerts.
- KU Leuven, Faculty of Engineering Technology (Group T), eMedia Lab. Prof. L. Geurts, prof. V. Vanden Abeele, J. Geuens.

The aims of the project are summarized in the project's flow in Figure 1.1. It covers multiple areas. On the hardware side, it aims to develop a small, unobtrusive, wireless monitoring system consisting mainly of inertial sensors, but possibly also incorporating electromyography (EMG). Moreover, it also aims to develop a biomechanical model for estimation of kine(ma)tics and loading, from established motion laboratory data and via the newly developed system. Another project output is a patient study on gamification strategies to increase adherence to therapy. This will be complemented by a mobile application for patient motivation and feedback.

This thesis focuses on STADIUS's contribution: the development of algorithms for activity recognition and assessment of functional capacity of patients suffering from spondyloarthritis using wearables. A more thorough discussion of SpA and current approaches for recognition and assessment can be found in Chapter 5.

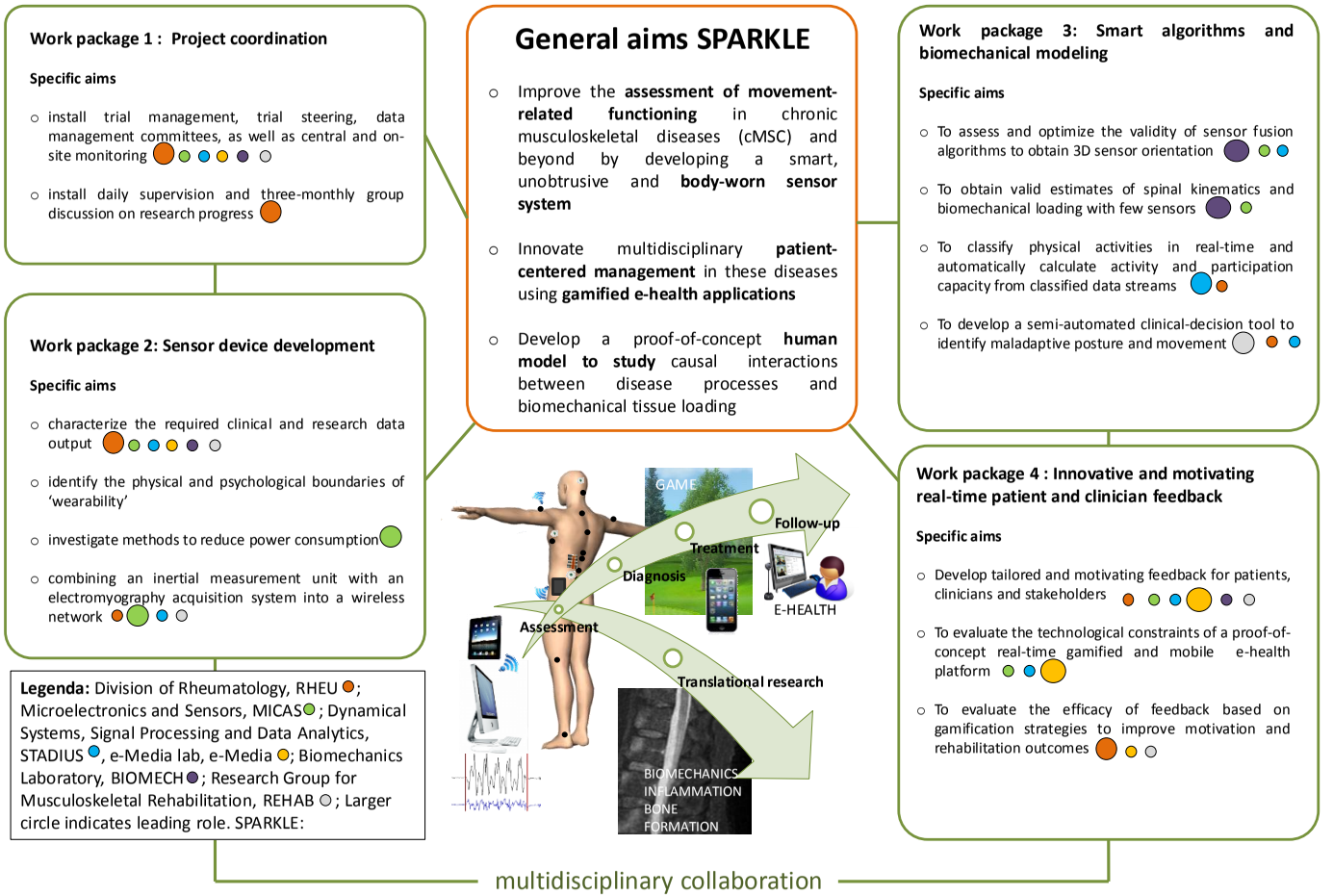


Figure 1.1: The SPARKLE project flow

1.3 Research objectives

This thesis consists of two parts, mirroring the two main aims touched upon in the problem description: interpretable clinical decision support and cMSC patient monitoring. Part I develops and implements a framework for interpretable classification and risk assessment, aimed at, but not restricted to, biomedical applications. Part II aims at developing algorithms for classification and assessment of physical activity and apply them on cMSC patient data as a proof-of-concept. This is summarized in eight objectives below.

Part I: Interval Coded Scoring Systems

Objective 1 The framework of Interval Coded Scoring (ICS) [168] will be expanded as a method to infer interpretable models for binary classification from data with optional guidance via expert knowledge. In particular, variable interactions should be supported. The visualization is expanded and the assessment module is enhanced to also quantify risk calibration quality. Chapter 3 covers this objective.

Objective 2 The basic ICS formulation suffers from non-uniqueness and limited scalability to high-dimensional datasets. New approaches to solve these issues should be investigated. This is covered in Chapter 4

Objective 3 Interpretation presumably comes at the cost of reduced performance. ICS should be validated and compared with current established machine learning methods. Moreover, the statistical significance of the differences should be investigated, bearing in mind the clinical significance. This is discussed in Chapters 3 and 4.

Objective 4 ICS is applicable to classification problems in many fields. To promote its use, a toolbox with an easy interface for problem formulation, solving, expert intervention, visualization, assessment and application to new data should be implemented. Its content is discussed throughout Chapters 3 and 4. A user manual is available in Appendix A.

Part II: Activity recognition and assessment

Objective 5 Window-based statistical features are commonly used for long term monitoring. Yet, patterns are hypothesized to be more suitable for recognizing transitory activities. Hence, it should be assessed whether statistical features offer additive value when combined with patterns as a more suitable approach to detection of transitory activities. This is addressed in Chapter 6.

Objective 6 As an alternative to adding more information, the structure of the data could be exploited instead of relying on statistical features. Chapter 7 reports on the impact of this approach.

Objective 7 Activity recognition is complemented with assessment. This involves the extraction and selection of informative quality features. Moreover, these should be incorporated in a classification or regression model to yield an objective measure as an alternative to the current patient-reported outcome. This is covered in Chapter 8.

Objective 8 Clinical data from wearable sensors need to be segmented and annotated. A toolbox should be developed for easy customizable visualization, organization, annotation and labelling of the data. The annotation accuracy can be improved via video synchronization. This toolbox is presented in Appendix C.

1.4 Outline of the thesis

As mentioned before, the thesis consists of two parts with a total of eight chapters and three appendices. They will be shortly discussed below. Their content and mutual links are also depicted graphically in Figure 1.2.

Chapter 1: Introduction
This is the current chapter, it is not mentioned in the graphical representation.

PART I: Interval Coded Scoring Systems

Chapter 2: Interpretable decision support
This Chapter introduces the state of the art and current practices for decision support, highlighting the need for interpretability. It discusses scoring systems as a historical way of structuring knowledge and constrasts them with more recent advanced machine learning techniques.

Chapter 3: Semi-automatic extraction of interpretable scoring systems
Here, the basic principles of Interval Coded Scoring are explained. The Chapter shows how scoring systems can be extracted from data based on total variation minimization combined with a binary expansion of the data. A semi-automatic iterative training procedure allows to tune the model. Also, a sensitivity study is conducted and ICS is applied on several validation sets.

Chapter 4: Extensions for larger problems
This Chapter starts from the observation that the version of ICS as introduced in the previous chapter suffers from non-uniqueness and scales badly to high-dimensional datasets. It proposes several approaches to deal with these problems and conducts a statistical comparison of all ICS approaches with established classifiers.

PART II: Activity recognition and Assessment from wearable sensors

Chapter 5: Activity monitoring
The first Chapter of the second part is an introduction to activity monitoring. It discusses spondyloarthritis, the disease under study, highlights the advantage of wearables and summarizes the state of the art on activity recognition and assessment. Finally, it also describes the datasets collected as part of the SPARKLE project.

Chapter 6: Activity recognition using patterns and statistical features
This Chapter starts from the ubiquity of statistical features for activity recognition in the literature and the question whether they are also suitable for the subgroup of transitory activities. It investigates how they can be combined with pattern detection to enhance the classifier's performance and assesses whether the combination outperforms the unimodal approaches.

Chapter 7: Activity recognition using patterns
As an alternative to the previous Chapter, the goal is to recognize activities solely based on patterns which are considered well-suited to characterize the transitory activities under study. The Chapter describes an experimental validation of the combination of Dynamic Time Warping (DTW) to capture variability in the data and Higher Order Discriminant Analysis (HODA) to take structure into account.

Chapter 8: Activity assessment
The last Chapter of part II investigates several ways to quantify the quality of activities. On the one hand, it studies several feature sets capturing various aspects of the performed activities. On the other hand, it introduces three models as an objective alternative to BASFI. They can all be referred to as instrumented BASFI (iBASFI). Two models are regression-based, the other one is an ICS classification model.

Chapter 9: Conclusion
The final Chapter summarizes the findings of this thesis, highlights its contributions and discusses several options for further research.

APPENDICES
The thesis is further complemented with three Appendices that contain additional background or implementation details.

Appendix A: Interval Coded Scoring toolbox
The first Appendix discusses the Matlab toolbox used to train and evaluate ICS models. Hence, its content depends heavily on Chapter 2 and 3.

Appendix B: Bath Ankylosing Spondylitis Functional Index
One of the items discussed in Chapter 5 is the use of a questionnaire as part of current practice. The Bath Ankylosing Spondylitis Functional Index (BASFI) is an important example in this thesis and has therefore been included as appendix.

Appendix C: SegmentLab toolbox
The data for activity recognition and assessment in the second part of the thesis was annotated using a dedicated Matlab/Python toolbox. More details on this toolbox can be found in this Appendix.

1.5 A note on notation

This manuscript uses standard notations for linear algebra. A scalar value is denoted by a lowercase character (a, b, $c, \ldots$). Small scalar values, denoted by ϵ are quantified with the scientific notation, e.g. $\epsilon = 5\text{E-}7$. A bold lowercase character indicates a vector ($\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}, \ldots$). A matrix is represented as bold uppercase ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}, \ldots$). A tensor is written in calligraphic font ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}, \ldots$).

In particular, a variable will be referred to as x, a data matrix as $\mathbf{X}$, a vector of class labels as $\mathbf{y}$ and a vector of data coefficients as $\mathbf{w}$ or occasionally as $\boldsymbol{\omega}$.

Superscripts are used to indicate the index into a set of variables or vectors. For example x^1 is the first variable, $\mathbf{a}^2$ is the second data vector. Its use as exponentiation should be clear from the context. Subscripts are used as index in a vector: $\mathbf{a}_1^3$ is the first element of the third vector $\mathbf{a}$.

The set of the real numbers is rendered $\mathbb{R}$, as usual. Similarly, $\mathbb{B}$ represents the set of binary numbers. For example, $\mathbf{A} \in \mathbb{B}^{3\times 2}$ is a binary matrix with three rows and two columns.

Finally, γ and λ are used as hyperparameters and $\varphi(\cdot)$ denotes a data transformation.

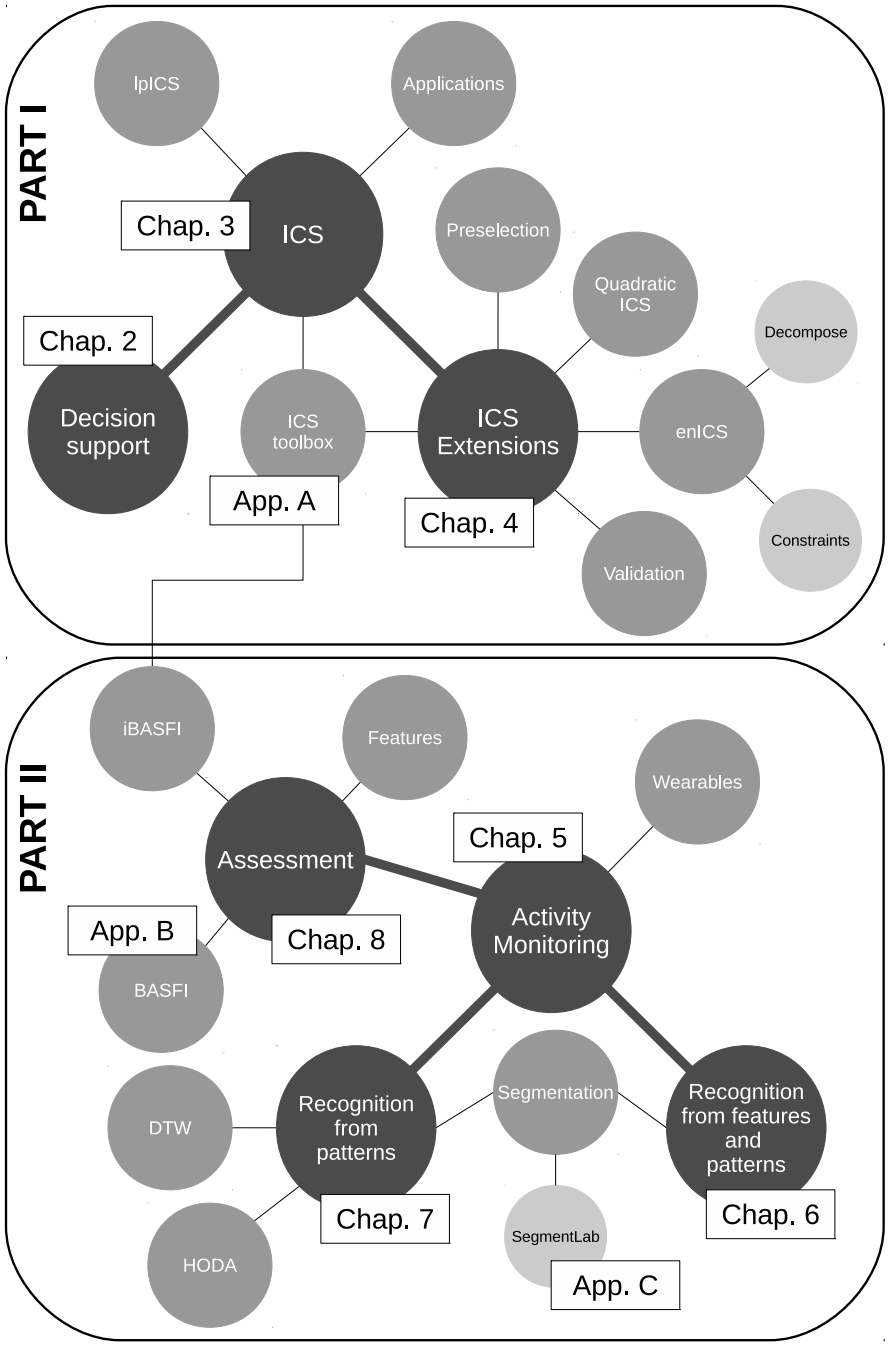


Figure 1.2: Overview of the thesis structure.

Part I

Interval Coded Scoring Systems

> 'We can get Hex to reverse the thaumic flow in the cthonic matrix of the optimized bi-direction octagonate, can't we?' 'Well, if I may express a preference, something that isn't complete gibberish would be more welcome next time, please.'
>
> T. Pratchett et al, *The Science of Discworld*

Chapter 2

Decision support

This Chapter sketches a general overview of decision support. It serves as an introduction to the other Chapters in Part I. It shortly describes what decision support entails and why it is important. Then, it discusses two common approaches: data-based and knowledge-based, followed by techniques combining their strengths, each with example applications. The Chapter concludes with an overview of several biomedical datasets that will be used for validation of the developed algorithms.

2.1 Introduction

According to the World Health Organization, 'health is a key determinant of economic growth, labour force productivity and poverty reduction. At the same time, health is increasingly viewed as a human right' [52]. Furthermore, the most recent World Happiness Report points to health problems as the major cause of reduced well-being in the United States [142]. As a result, it should not surprise that in 2015 a world-wide total of 7.2 trillion USD, around 10% of the GDP, was spent on health [176]. This shows that healthcare is of paramount importance, both socially and economically.

Diagnosis and treatment hinge on decisions made by clinicians based on their experience and knowledge obtained through extensive training. Given their potential impact, the importance of these decisions should not be underestimated and continuous improvement efforts should be made. An early-adopted way to improve healthcare quality was experimental research, gathering and transferring knowledge through written language. This lead eventually to regulated evidence-based practice in many disciplines [115]. Yet, with the current rate of knowledge

discovery it becomes hard for individuals to keep an overview. This phenomenon is called *information overload*. It entails that the available information cannot be processed effectively by an observer (e.g. a clinician) [152].

Therefore, the last decades have seen a wide growth of the application of Machine Learning and Data Science in healthcare, giving rise to the field of *health informatics*. The term itself was probably first used in the seventies [134]. It can be described to be at 'the crossroad of information science, computer science, medicine, and healthcare, with a wide range of application areas including nursing, clinical care, public health, and biomedicine' [94]. Largely, it has gained importance due to the increase of computational power, but also due to the enormous amount of medical data that has become available in an accessible digital format. Hence, computer systems have been designed to serve as *Clinical Decision Support Systems* (CDSS). For example, models based on the Electronic Health Record (EHR) have been used for early prediction of late-onset neonatal sepsis [108] and hand, foot or mouth diseases [97]. Several studies have shown that such systems indeed increase practitioner performance and improve patient outcome, while decreasing the costs at the same time [40, 59, 82]. Meanwhile, different systems have been developed. One can roughly distinguish two types: data-based and knowledge-based decision support systems [20].

2.2 Data-based decision support

This type of CDSS is the most exemplary of the application of general machine learning techniques on (bio)medical problems. The underlying idea is harvesting the information contained in past data. Although some techniques can use the full database of previous records, the more usual approach is to extract a model that summarizes the most important discriminating information for the problem at hand using a wide variety of available techniques.

2.2.1 Regression techniques

These techniques try to predict a continuous variable. Given a data matrix $\mathbf{X}$ and a corresponding label vector $\mathbf{y}$, one of the best known formulations is least-squares linear regression. It simply fits a linear model $\mathbf{X}\mathbf{w} + b$ with weight vector $\mathbf{w}$ and offset b by minimizing the quadratic error with regard to the label vector $\mathbf{y}$ i.e. $\min_w \|\mathbf{X}\mathbf{w} + b - \mathbf{y}\|_2^2$. However, the optimization problem is often required to satisfy certain additional conditions, which can be obtained by introducing penalty terms. This is termed regularized least squares. One such condition is sparsity, a measure of model simplicity. The goal is to obtain a well-performing model by focusing on as few variables as possible. In other

words, the aim is to have as many zeros in $\mathbf{w}$ as possible. It has become very popular with the advent of the Least Absolute Shrinkage and Selection Operator (LASSO) [164] and elastic net [186]. Both add an ℓ_1-norm penalty on $\mathbf{w}$ to promote sparsity as a placeholder for the ideal ℓ_0-norm which has undesirable optimization properties. The elastic net formulation imposes an additional ridge regression ℓ_2-norm penalty to obtain a better-conditioned result. Summarizing, the regularized optimization problems can be formulated as in Equation 2.1, with hyperparameters λ to make a trade-off between approximation error and sparsity. Alternatively, the penalties could also be imposed as constraints.

$$\begin{aligned}
&\min_{\mathbf{w},b} \|\mathbf{X}\mathbf{w} + b - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 && \text{(ridge regression)} \\
&\min_{\mathbf{w},b} \|\mathbf{X}\mathbf{w} + b - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_1 && \text{(LASSO)} \\
&\min_{\mathbf{w},b} \|\mathbf{X}\mathbf{w} + b - \mathbf{y}\|_2^2 + \lambda_1\|\mathbf{w}\|_2^2 + \lambda_2\|\mathbf{w}\|_1 && \text{(elastic net)}
\end{aligned} \tag{2.1}$$

The use of regression as a CDSS has largely been restricted to logistic regression for likelihood estimation [155]. Still, elastic net and LASSO have been used, if not as CDSS, to gain insight in several biomedical problems and as feature selection technique. At its introduction, the elastic net was immediately applied to microarray data due to its ability to manage high dimensional data [186]. Similarly, [147] used it to identify neuroimaging and proteomic biomarkers for Alzheimer's disease. More recently, [141] applied the elastic net to estimate functional connectivity from fMRI data, whereas [64] exploited it to detect mobility parameters progression of early multiple sclerosis and to discriminate patients from a control population based on Timed-Up-and-Go (TUG) tests.

2.2.2 Classification techniques

Classification methods try to distinguish several data classes in either a binary or multiclass setting. Here, known observations represent the classes. The model inferred from this data aims at generalization to predict the class of new observations. Well-known approaches include Support Vector Machines (SVM) [172], Naive Bayes (NB) classifiers [93], decision trees (DT) [135], random forests (RF) [70] and neural networks (NN) [27].

Classification methods can be divided in two main groups [77]. Firstly, discriminative methods. The model describes a boundary between classes. Binary SVMs are a typical example. For the linear SVM, the decision boundary is given by $\mathbf{x}^T\mathbf{w} + b = 0$. This is a hyperplane dividing the space in a positive and a negative segment. The coefficients $\mathbf{w}$ are computed such that the margin between the classes is maximal. A full formulation of the optimization problem involved will be given in Chapter 3.

The second group are generative methods. The aim is to model the data distribution generating the data. In this way, such methods can assess the likelihood of a new sample belonging to any of the classes. Naive Bayes classifiers fall into this category, most often approximating the classes by multivariate gaussian distributions. As another example, regression methods are considered generative as well.

All methods mentioned so far have been applied to solve biomedical problems and serve as CDSSs. For example, all of them except NN were compared for the early detection of late-onset neonatal sepsis [108]. RFs have been used for classification of microarray data [50]. Another application involves screening for ECG arrhythmia using, among others, SVMs [112]. Further examples include NNs for severity assessment of breast cancer [101] and SVMs for detection of mild cognitive impairment and Alzheimer disease from multimodal brain imaging [146]. A final example is the so-called 'doctor AI' system [41]. It is a generally applicable system using a recursive NN to predict a diagnosis category or medication class from more than a thousand possibilities in the EHR.

2.2.3 Pros and cons

The machine learning approaches outlined above have some clear advantages. They incorporate experience based on previous cases, refining it to its essence, either in a data model or a class boundary. Training is mostly automatic without the need for manual intervention. Moreover, the methods have solid mathematical and statistical foundations built on decades of extensive research and have been successfully applied in many different fields. Finally, they also boast impressive results in almost all healthcare domains, sometimes surpassing clinical experts.

The downside of many of these models is their lack of transparency and interpretability. Whereas one can still understand the outcome of a logistic regression model or a decision tree, this is much harder for nonlinear SVMs or NNs. Particularly in case of CDSSs aimed at providing a diagnosis or prognosis or even a treatment decision, such black box characteristics are far from appealing. On the one hand, the clinician has to trust the model. He or she treats patients according to its guidance, and does not always understand how the algorithm reached its conclusion. Hence, there is no easy way to check if the decision corresponds with personal or more general clinical experience. On the other hand, such black box decision-making CDSSs create a problem of (legal) responsibility. If a clinician does not fully understand the algorithm or its conclusion, can he or she be held responsible for carrying out the treatment?

2.3 Knowledge-based decision support

Structured ways to summarize human knowledge rather than extracting it from data predate the data-based systems although the distinction is blurry at times. The first computerized attempts were the *expert systems* based on early artificial intelligence. Researchers tried to represent human knowledge using, often incomplete, sets of rules. Decisions and explanations for new data could be obtained from inference engines. This led to systems such as MYCIN for the selection of antibiotics or INTERNIST-I for diagnosis in internal medicine [55]. Many of these systems could output their inference trace when requested. This allowed clinicians to see how the engine obtained its decision by reviewing which rules had been applied.

Another approach to summarize knowledge favours manual applicability. Such *medical scoring systems* have been used for decades and continue to be used and developed until today. Although conceptually similar to expert systems, they are usually much simpler and have a more limited focus. They consist of several simple logic rules e.g. whether the patient's age is higher than a threshold or whether the patient suffers from a specific symptom. Each rule has a corresponding number of points that should be added if it holds true. Adding all these points yields a score, which can be linked to an empirical risk and possible treatment options. Examples of scoring systems include Alvarado for appendicitis [8], CHA_2DS_2-VASc for atrial fibrillation [96], Glasgow for pancreatitis [118], CURB-65 for pneumonia [78] and ASDAS for disease activity in ankylosing spondylitis [100]. Tables 2.1 and 2.2 display the CHA_2DS_2-VASc score as an example.

It is applied as follows. Consider a female patient of 67 years with hypertension. Some years back, she suffered a stroke. Table 2.1 indicates that her total score equals 5: one point for hypertension, two points for a prior stroke, one point for

Table 2.1: The CHA_2DS_2-VASc score [96].

Risk factor	**Points**
Congestive heart failure/Left Ventricular dysfunction	1
Hypertension	1
Age $\geq$ 75	2
Diabetes mellitus	1
Prior **S**troke/Transient ischemic attack/Thromboembolism	2
Vascular disease	1
Age 65-74	1
Sex **c**ategory (i.e. female)	1

Table 2.2: Risk estimation for the CHA_2DS_2-VASc score taken from cohort studies as reported on Medscape [2].

Score	Annual stroke risk
0	0%
1	1.3%
2	2.2%
3	3.2%
4	4.0%
5	6.7%
6	9.8%
7	9.6%
8	6.7%
9	15.2%

being aged 65-74 and one point for being female. According to Table 2.2 she runs an empirical risk of 6.7% of suffering another stroke.

2.3.1 Pros and cons

The advantage of knowledge-based systems is the fact that they are constructed based on accepted clinical knowledge and they are interpretable, particularly in case of scoring systems. However, they are often crude rules of thumb, albeit sometimes statistically verified post hoc. This is particularly true for expert and scoring systems that have been constructed through questionnaires and structured discussions.

2.4 Data-driven interpretability

Some other approaches have tried to combine aspects of both data and knowledge-driven techniques. The majority focuses on the extraction of an interpretable view on either the data or an existing model. For example, several studies investigated the construction of rule systems from data. Sometimes, rules are generated directly, e.g. using Bayesian techniques [91]. Alternatively, one can derive them from an SVM [19, 111].

A related approach is the construction of scoring systems from data using sparse integer programming [167]. This approach is similar to Interval Coded Scoring

as introduced in the next Chapter. However, it does not consider variable interactions and only supports linear effects.

The main drawback of all approaches discussed so far is their restriction to simple models consisting of logical clauses, decision tree structures or linear effects without interactions. Moreover, they are purely data-driven, no prior expert knowledge is incorporated in their default application.

Fuzzy cognitive maps are an interesting exception. The technique updates an expert-constructed cognitive map, that is, a neural network, via fuzzy rule extraction techniques. It has been applied to radiotherapy treatment planning selection [125]. Yet, although the final network is somewhat interpretable, the strongest contribution of the approach appears on the level of integration of different kinds of information, including quantified expert knowledge, rather than on the interpretable representation of the decision process or final model.

2.5 Datasets

The following Chapters will introduce techniques to derive scoring systems from data in a semi-automatic way. This Section describes machine learning datasets from the University of California, Irvine (UCI) that will be used to validate the newly developed techniques and compare them to established approaches. Nearly all of the datasets considered are biomedical. Nevertheless, many other areas could also benefit from interpretable models e.g. process management, behaviour studies or modelling of physics systems.

Acute inflammations [44] The set contains 120 observations with 6 variables, one continuous and five binary. It was originally intended to test a new expert system for diagnosis of two diseases of the urinary system, namely acute inflammations of urinary bladder and acute nephritis. As such, the set defines two classification problems: each disease versus a control group. The set contains 30 healthy controls, 40 patients only suffering from inflammation, 31 only suffering from nephritis and 19 suffering from both. The two binary classification problems are relatively balanced: 59/120 diseased (inflammation) and 50/120 diseased (nephritis). This dataset was perfectly classified by a rule-based expert system.

Breast cancer diagnosis [107] The Wisconsin diagnostic breast cancer database (WDBC) consists of 569 cell observations, 357 of which are benign and 212 malignant, with 30 continuous features. In its original publication paper, it has been used for classification using Linear Programming with an estimated accuracy of 97.5%.

Cardiotocography [14] This database consists of 21 variables extracted from 2126 foetal electrocardiograms. 18 variables can be considered continuous, one categorical and two ordinal. The labelling distinguishes between 1655 normal, 295 suspected pathological and 176 pathological records. This study limits the problem to a binary classification (normal vs suspected/pathological).

Chronic kidney disease [139] This dataset with 400 observations and 24 variables contains many missing values, about 10% of the data. Therefore, the dataset was reduced. Firstly, the variables missing in more than 10% of the patients were eliminated. Then, the remaining patients with missing values were removed. This resulted in a dataset of 354 observations and 12 variables. Alternatively, imputation could be attempted, but that is outside the scope of the current work. Eight of the remaining variables are binary, four are continuous. The remaining observations contain 220 healthy controls and 134 patients.

Indian liver patient data [136] The set contains 583 observations of nine continuous variables and one binary variable. Four observations were removed due to missing data. The goal is to distinguish 414 healthy controls from 165 liver patients.

Mushrooms [143] The mushroom set includes descriptions of 23 species of the Agaricus and Lepiota family. The aim is to classify them as either edible or poisonous based on 19 categorical attributes, one binary and one ordinal. 8124 samples were provided with a balanced class distribution of 51.8% edible and 48.2% poisonous. In the past, perfect classification has been obtained by an expert system consisting of only four rules [53].

Vertebral column [45] This data set consists of 310 observations with 6 real-valued biomechanical attributes. Class labels distinguish 100 'Normal' from 210 'Abnormal' patients. Normally, it is a three class problem with the 'Abnormal' class consisting of patients suffering from disk hernia or spondylolisthesis, but they have been pooled here to arrive at a binary classification problem.

Anyone can foretell the future. And everyone does it, for it is simple. It is no great art to foretell it. The art is in foretelling it accurately.

A. Sapkowski, *The Tower of Swallows*

Chapter 3

(Semi-)automatic extraction of interpretable scoring systems

The previous Chapter described current trends in decision support. It also mentioned the importance of interpretability, particularly when dealing with clinical decisions. It showed how simple medical scoring systems have been used for decades, but are now often outperformed by data-driven black box machine learning techniques. This Chapter introduces Interval Coded Scoring (ICS). It produces data-derived medical scoring systems guided by expert knowledge through an interactive training phase. It supports non-linear models and variable interactions. Despite the apparent complexity, it presents the final model in a colour-coded interpretable and insightful representation. Hence, it is suitable for both manual and computerized use. The foundations of this work were first presented by Van Belle et al. [168]. This Chapter's contributions consist of the introduction of variable interactions and a sensitivity study. A first section describes the rationale and the main principles behind ICS. Next, it introduces the ICS algorithm including its extension to interactions, as well as its interpretable visualization. Then, stability and time complexity are discussed based on a sensitivity study. In the next Section several applications are discussed. Finally, a discussion elaborates on the methods and findings in this Chapter. The content of this Chapter is based on earlier publications [24, 26].

3.1 Rationale and general approach

Clinical decision support can be improved by building models based on available data. As explained before, these models are often not interpretable. In contrast, ICS is an approach to derive interpretable classification systems. It is based on the principles of scoring systems, explained in Chapter 2, Section 2.3. The system tries to reach a conclusion based on the available variables. Suppose one wants to estimate the risk of heart failure based on age, Body Mass Index (BMI) and IQ. Broadly describing a simplified process, ICS obtains a system according to the following steps:

1. All variables are divided into a number of intervals. For example, age can be split into ranges < 25, $25 - 55$ and > 55 while BMI and IQ have their own intervals. The intervals are defined using a threshold vector $\boldsymbol{\tau}$ based on the amount of data and the data distribution, as will be explained in detail later.

2. An optimization algorithm uses cross-validation to attribute a number of score points to each interval of every variable.

3. The previous stage can lead to simplification of the model. If neighbouring intervals of the same variable have the same score, the intervals can be merged. For example, if age intervals < 25 and $25 - 55$ each contribute two score points, the new interval ≤ 55 contributes two points as well.

4. The model is refined iteratively: if neighbouring intervals have *almost* the same number of points, they can also be merged and the scores can be recomputed. This guarantees that only the most relevant differences are taken into account. At this point, a user could define what should be considered relevant, although it can also be decided automatically. This process is called *iterative reweighting* and continues until convergence.

5. If at some time during the reweighting the score for all intervals of a variable is zero, that variable can be removed. For example, IQ might not be informative for the risk on heart failure.

6. In the end, all non-informative variables have been discarded and intervals have been merged to reach the required level of simplicity for a given performance. This yields a scoring system.

7. Finally, the score obtained by summing the number of points over all variables can be linked to the empirical risk on heart failure via logistic regression on the training data.

In practice, the score points are not attributed as integers, but rounded at the end of the ICS process for computational reasons. Although the process can be performed fully automatically, the user can intervene at several points.

- While defining an ICS problem, he or she can define a maximum number of intervals for each variable.
- The user can provide prior information about which variables should be included and which variable interactions should be considered.
- Iterative reweighting provides feedback at each iteration. Possible simplifications may come at a cost of classification performance, judged via cross-validation results. The user can intervene and impose a desired trade-off between simplicity and performance.

The next Sections will define ICS in a more rigorous mathematical way.

3.2 The Interval Coded Scoring algorithm

In essence, an ICS system is a general binary classifier drawing on concepts from sparse optimization. It starts from a data matrix $\mathbf{X} \in \mathbb{R}^{N \times N_d}$ with N observations of dimensionality N_d, paired with a label vector $\mathbf{y}$ containing class labels $y_i \in \{-1, 1\}$ $(i = 1, \ldots, N)$. In the end, it produces a model $\varphi(\mathbf{X})\mathbf{s}$. $\mathbf{s}$ is a vector of rounded coefficients, yielding integer score points. Because of discarded variables, it will be sparse. $\varphi(\cdot)$ is the data transformation converting the original variables in $\mathbf{X}$ to an interval representation $\mathbf{Z}$.

ICS has two concrete implementations. This Chapter discusses the first one, the other one is discussed in the next Chapter. They follow the same approach expressed in Section 3.1 but they differ in three building blocks:

- The data transformation $\varphi(\cdot)$ and, consequently, the transformed data matrix $\mathbf{Z}$
- The ICS core algorithm computing the coefficient vector $\mathbf{w}$ for the intervals
- The exact iterative reweighting procedure

The symbols used so far refer to the general concepts used by ICS. The concrete implementation in this Chapter is denoted by a tilde, yielding e.g. $\tilde{\varphi}(\cdot)$, $\tilde{\mathbf{Z}}$ and $\tilde{\mathbf{w}}$. The three building blocks will be discussed first. Afterwards, an overview algorithm will be presented and the visualization and assessment will be discussed.

3.2.1 Data transformation

The variables in data matrix $\mathbf{X}$ can be binary, ordinal, categorical or continuous. A new binary dataset $\tilde{\mathbf{Z}}$ is constructed by dividing the range of the original variables in a number of intervals through thresholds τ. These intervals are defined as follows:

- **A binary variable** value x^p is expanded to two binary values $\tilde{z}_1^p$ and $\tilde{z}_2^p$ such that $\tilde{z}_1^p = 1$ if $x^p = 0$, 0 otherwise, and $\tilde{z}_2^p = 1$ if $x^p = 1$, 0 otherwise.
- **A categorical variable** value x^p is expanded into a number of binary values $z_1^p, \ldots, z_{N_c}^p$, with N_c the number of categories. All binary values are zero, except for the one corresponding to the category indicated by x^p.
- **A continuous variable** value x^p results in N_r binary values through thresholds τ_k^p with $k = 1, \ldots, N_r - 1$. N_r can be set beforehand for each variable x^p. The thresholds are calculated as to divide the range of x^p into N_r equal percentiles (intervals) with the restriction that any percentile must contain at least five data points. Otherwise, the number of intervals is reduced. For each of the intervals, a new binary variable z_k^p is created. All values $z_k^p = 0, \forall\ k$, except the one corresponding to the interval in which x^p lies.
- **Ordinal variable** values are treated as continuous data, except that thresholds can only be integer values.
- **Interactions** can be modelled by combining the principles above. This is an extension of the original ICS [168]. For example, the interaction between heart rate and energy readings from accelerometry can indicate whether an increased heart rate is due to physical activity or not. Imagine an observation of these two continuous variables $[x^1, x^2]$, with N_r^1 and N_r^2 intervals, respectively. This single observation would be expanded to a binary matrix: $[x^1, x^2] \in \mathbb{R}^2 \rightarrow \mathbf{B} \in \mathbb{B}^{N_r^1 \times N_r^2}$. Then, $\mathbf{B}$ is vectorized to a single transformed data vector encoding the rows and columns. All values of $\mathbf{B}$ are zero, except the one corresponding to the row representing the interval containing the current value of x^1 and the column representing the interval containing the current value of x^2.

Combining all variables for one observation yields a new (binary) observation vector: $\mathbf{x}_i \in \mathbb{R}^{N_d} \rightarrow \mathbf{z}_i \in \mathbb{B}^{N_s}$, where N_s is the total number of intervals over all variables and interactions. The data matrix $\mathbf{X}$ is thus transformed into a binary data matrix $\tilde{\mathbf{Z}} = \tilde{\varphi}(\mathbf{X}, \boldsymbol{\tau})$ using the thresholds $\boldsymbol{\tau}$. The transformation $\tilde{\varphi}$ supports both main effects (only a single variable x^p is affected) and interactions (x^{p_1}, x^{p_2}) as explained above.

Mathematically, the transformation of a single observation $\mathbf{x}_i$ is given as:

$$
\begin{aligned}
&\tilde{\mathbf{z}}_i = [\tilde{\mathbf{z}}_i^1, \tilde{\mathbf{z}}_i^2, \ldots, \tilde{\mathbf{z}}_i^{N_d}, \tilde{\mathbf{z}}_i^{1,2}, \ldots, \tilde{\mathbf{z}}_i^{N_d-1,N_d}] \\
&\text{where} \quad \tilde{\mathbf{z}}_i^p = \tilde{\varphi}^p(x_i^p) \quad \text{with} \\
&\qquad \tilde{\varphi}^p(x_i^p) = [I(x_i^p < \tau_1^p), I(\tau_1^p \le x_i^p < \tau_2^p), \ldots, I(\tau_{N_p-1}^p \le x_i^p)] \\
&\quad \text{and} \quad \tilde{\mathbf{z}}_i^{p_1,p_2} = \text{vec}(\tilde{\varphi}^{p_1,2}(x_i^{p_1}, x_i^{p_2})) \quad \text{with} \\
&\qquad \tilde{\varphi}_{k,l}^{p_1,p_2}(x_i^{p_1}, x_i^{p_2}) = I(\tau_{k-1}^{p_1} \le x_i^{p_1} < \tau_k^{p_1} \quad \& \quad \tau_{l-1}^{p_2} \le x_i^{p_2} < \tau_l^{p_2}) \\
&\qquad\qquad k \in \{1, \ldots, N_{p_1}\}, l \in \{1, \ldots, N_{p_2}\} \\
&\qquad\qquad \text{defining} \quad \tau_0^{p_1} = \tau_0^{p_2} = -\infty \quad \text{and} \quad \tau_{N_{p_1}}^{p_1} = \tau_{N_{p_2}}^{p_1} = \infty
\end{aligned}
\tag{3.1}
$$

In this equation, I is an indicator function, yielding 0 or 1 when its argument is `false` or `true`, respectively. N_p refers to the number of intervals for variable x^p. Summarizing, every observation x_i^p of a variable x^p is expanded into a binary vector $\tilde{\mathbf{z}}_i^p$ containing only a single 1 indicating the interval to which the value of x^p belongs. Two-way interactions (x^{p_1}, x^{p_2}) yield a binary matrix containing only a single 1. The `vec` in the fourth line in Equation 3.1 indicates vectorization of this matrix to obtain a vector. Finally, transformations for several variable values are simply concatenated to obtain one sparse vector $\tilde{\mathbf{z}}_i$, a row of the matrix $\tilde{\mathbf{Z}}$. Which main effects and interactions are considered initially can be set by the user.

Although the mapping function has only been defined for main effects and two-way interactions it can easily be generalized to higher-order interactions and has been implemented as such. In practice, such higher-order interactions rarely occur in biomedical datasets.

3.2.2 The `lpICS` core algorithm

The core algorithm uses the data transformation explained above to calculate an initial estimate for the coefficient vector $\tilde{\mathbf{w}}$. It is based on the soft-margin Support Vector Machine (SVM) formulation [172], given as follows:

$$
\begin{aligned}
&\min_{\mathbf{w},b,\varepsilon} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma \sum_{i=1}^{N} \varepsilon_i \\
&\text{s.t.:} \quad \begin{cases} y_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b) \ge 1 - \varepsilon_i, & \forall\ i = 1, \ldots, N \\ \varepsilon_i \ge 0, & \forall\ i = 1, \ldots, N \end{cases}
\end{aligned}
\tag{3.2}
$$

In this formulation, $(\mathbf{x}_i, y_i)$ are pairs of observations and labels, $\mathbf{w}$ and b the coefficients and bias of the model, respectively, and ε_i slack variables used for the regularization controlled by the hyperparameter γ. Equation 3.2 is the

standard formulation. To simplify notation, it can also be rendered in matrix form:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\varepsilon}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma\boldsymbol{\varepsilon}^T\mathbf{1} \\ \text{s.t.:} \quad & \begin{cases} \mathbf{Y}(\mathbf{Z}\mathbf{w} + b) \geq \mathbf{1} - \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} \geq \mathbf{0} \end{cases} \end{aligned} \tag{3.3}$$

Here, $\mathbf{Y} \in \mathbb{R}^{N\times N}$ is a diagonal matrix containing the labels y_i, $\mathbf{Z} = \varphi(\mathbf{X})$, the transformed data matrix and $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ is the vector of slack variables ε_i. This equation is the basis of the ICS core algorithm developed in this Chapter.

A first step toward ICS is application of the specific data transformation defined in Equation 3.1. This can be expressed explicitly by exchanging $\mathbf{Z}$ and, consequently $\mathbf{w}$ by $\tilde{\mathbf{Z}} \in \mathbb{B}^{N\times N_s}$ and $\tilde{\mathbf{w}} \in \mathbb{R}^{N_s}$, respectively.

However, the aim of ICS is different from SVM. The SVM Equation 3.3 tries to maximize the classification margin (minimizing $\mathbf{w}^T\mathbf{w}$) while minimizing the classification error expressed by the slack vector. In general, each coefficient w_i in $\mathbf{w}$ will be different. This is not desirable for the ICS coefficients $\tilde{\mathbf{w}}$. Recall that they correspond to intervals of the variables. A scoring system grasping the general trends rather than dataset-specific details would try to have as few intervals as possible. This can be achieved by neighbouring intervals having the same score. The principle is illustrated in Figure 3.1 for a single variable x^p with 18 intervals. It has a coefficient vector $\mathbf{w}^p$, a sub-vector of $\mathbf{w}$. The full line could be the output of general SVM training: all intervals have a different score. However, the main variations shown by the dashed line are the most important. Firstly, a simpler model is more likely to capture general tendencies rather than sample-specific changes. Secondly, it is simpler to understand. The training process can take this into account by forcing a piecewise constant $\tilde{\mathbf{w}}$, or, equivalently, a sparse vector of differences of $\tilde{\mathbf{w}}$. Thus, the maximal margin objective can be replaced by *Total Variation Minimization* [140] via the ℓ_1 objective term $\|\mathbf{D}\tilde{\mathbf{w}}\|_1$. The binary matrix $\mathbf{D} \in \mathbb{R}^{N_{ds}\times N_s}$ encodes N_{ds} differences between coefficients of adjacent intervals of the same variable x^p or variable interaction (x^{p_1}, x^{p_2}). In the latter case, both the horizontal and vertical differences in the non-vectorized matrix shape are taken into account. So, all rows of $\mathbf{D}$ contain a single 1 and a single -1. The only exceptions are rows corresponding to the first interval of every variable or variable interaction. They don't code differences but directly minimize the coefficient attributed to obtain the lowest possible offset. Due to this construction, the sparsity-inducing ℓ_1 norm is applied on the difference of the weights but also minimizes their offset.

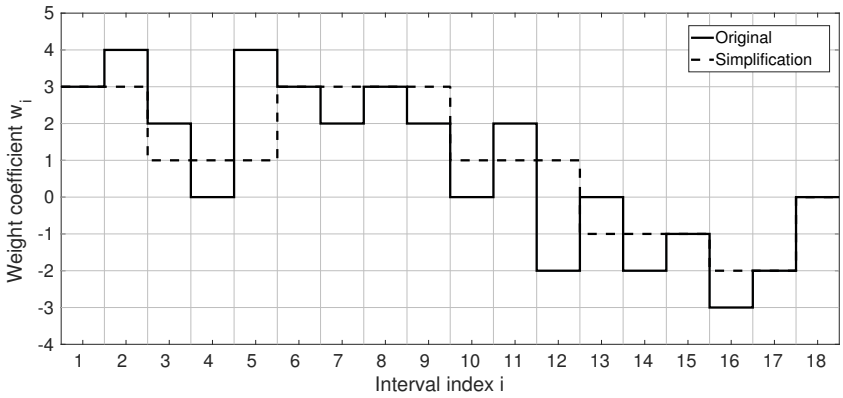


Figure 3.1: ICS model simplification. Standard optimization leads to different coefficients for every interval, shown as the full line. Sparsity of the differences between coefficients of the intervals leads to a simplified coefficient vector capturing only the major effects, shown as a dashed line.

Taking the data transformation and changed objective into account, the ICS approach can be stated in matrix form as:

$$\begin{aligned} &\min_{\tilde{\mathbf{w}},b,\boldsymbol{\varepsilon}} \quad \|\mathbf{D}\tilde{\mathbf{w}}\|_1 + \gamma\boldsymbol{\varepsilon}^T\mathbf{1} \\ &s.t. \quad \begin{cases} \mathbf{Y}(\tilde{\mathbf{Z}}\tilde{\mathbf{w}} + b) \geq \mathbf{1} - \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} \geq \mathbf{0} \end{cases} \end{aligned} \tag{3.4}$$

Similarly to the original SVM formulation, it tries to fit a model $\tilde{\mathbf{Z}}\tilde{\mathbf{w}} + b$ on the class labels $\mathbf{y}$ in the diagonal matrix $\mathbf{Y}$. The constraints enforce correct classification, allowing slack $\boldsymbol{\varepsilon}$ with respect to the classification boundary. Equation 3.4 can be restated and solved efficiently as a linear programming (LP) problem. Therefore, it is further referred to as `lpICS`.

3.2.2.1 Solving `lpICS`

To restate Equation 3.4 as LP problem a new vector $\mathbf{f} = |\mathbf{D}\tilde{\mathbf{w}}| \in \mathbb{R}^{N_{ds}}$ is introduced. Then, the problem becomes:

$$\begin{aligned} \min_{\tilde{\mathbf{w}},b,\mathbf{f},\boldsymbol{\varepsilon}} \quad & \mathbf{f}^T\mathbf{1} + \gamma\boldsymbol{\varepsilon}^T\mathbf{1} \\ \text{s.t.} \quad & \begin{cases} \mathbf{D}\tilde{\mathbf{w}} - \mathbf{f} \leq \mathbf{0} \\ -\mathbf{D}\tilde{\mathbf{w}} - \mathbf{f} \leq \mathbf{0} \\ \mathbf{Y}(\tilde{\mathbf{Z}}\tilde{\mathbf{w}} + b) \geq \mathbf{1} - \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} \geq \mathbf{0} \end{cases} \end{aligned} \tag{3.5}$$

The two constraints involving $\mathbf{f}$ enforce it to contain the absolute values of $\mathbf{D}\tilde{\mathbf{w}}$. If an element of the latter is positive, the first constraint enforces the corresponding element in $\mathbf{f}$ to be positive as well and its value should be equal or higher. Because of the objective function, the lowest possible value will be selected such that the constraint is tight. Hence, it will be equal to the element of $\mathbf{D}\tilde{\mathbf{w}}$. The second constraint will be trivially satisfied. In contrast, if the difference element is negative, the second constraint will also enforce the corresponding element in $\mathbf{f}$ to be positive. Due to the objective function, its value should be minimal and is therefore fixed on the absolute value of the difference element under consideration. In this case, the first constraint will be trivially satisfied. Overall, this results in $\mathbf{f} = |\mathbf{D}\tilde{\mathbf{w}}|$.

Because the formulation in Equation 3.5 contains a linear objective function and linear inequality constraints it can be solved via linear programming. The total number of variables is given by $\#\tilde{\mathbf{w}}+\#b+\#\mathbf{f}+\#\boldsymbol{\varepsilon} = N_s+1+N_{ds}+N$, although $\mathbf{f}$ is of course defined through the values of $\tilde{\mathbf{w}}$. The number of constraints sums up to $N + 2N_{ds} + N = 2(N + N_{ds})$, but fulfilling one of the absolute value constraints, combined with the objective, automatically fulfils the other as well.

The data construction and ℓ_1 penalty lead to the problem being sparse, which can be exploited by most recent LP solvers. Normally, Matlab's solver is used, but IBM's CPLEX is selected if available: it drastically improves execution time.

3.2.3 Iterative reweighting

Iterative reweighting is applied as a way to further simplify the model [168]. The example in Figure 3.1 is often not achievable using a single step of total variation minimization. It can be achieved by iterative reweighting. The general idea is that small differences are penalized to force them to be smoothed out in a next iteration. It is implemented as a repetitive execution of the `lpICS` formulation in Equation 3.4. $\mathbf{D}\tilde{\mathbf{w}}$ should be replaced by $\tilde{\boldsymbol{\chi}}\mathbf{D}\tilde{\mathbf{w}}$. $\tilde{\boldsymbol{\chi}} \in \mathbb{R}^{N_{ds}\times N_{ds}}$ is a diagonal matrix containing weights $\tilde{\boldsymbol{\chi}}_{i,i}$, $i = 1, \ldots, N_{ds}$. The reweighting values are computed based on the current solution of the problem $\tilde{\mathbf{w}}$ as $\tilde{\boldsymbol{\chi}}_{i,i} = 1/(\epsilon_1 + a|(\mathbf{D}\tilde{\mathbf{w}})_i|)$. The reweighting is summarized in Algorithm 3.1.

Algorithm 3.1 Iterative reweighting in `lpICS`

1: Given: $\tilde{\mathbf{w}}$, a
2: **repeat**
3: $\quad \tilde{\mathbf{w}}^{\text{old}} = \tilde{\mathbf{w}}$
4: $\quad \tilde{\boldsymbol{\chi}}_{i,i} = \frac{1}{\epsilon_1 + a|(\mathbf{D}\tilde{\mathbf{w}}^{\text{old}})_i|}, \forall\ i = 1, \ldots, N_{ds}$
5: $\quad \tilde{\mathbf{w}} \leftarrow$ solve Equation 3.4 with $\mathbf{D}\tilde{\mathbf{w}}$ replaced by $\tilde{\boldsymbol{\chi}}\boldsymbol{D}\tilde{\boldsymbol{w}}$:

$$\min_{\tilde{\mathbf{w}}, b, \boldsymbol{\varepsilon}} \quad \|\tilde{\boldsymbol{\chi}}\mathbf{D}\tilde{\mathbf{w}}\|_1 + \gamma \boldsymbol{\varepsilon}^T \mathbf{1}$$
$$s.t. \quad \begin{cases} \mathbf{Y}(\tilde{\mathbf{Z}}\tilde{\mathbf{w}} + b) \geq \mathbf{1} - \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} \geq \mathbf{0} \end{cases}$$

6: **until** AVG$(|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}^{\text{old}}|) < \epsilon_2$ or #iter == 10

ϵ_1 and ϵ_2 are small values of 5E-4 and 1E-8, respectively. They help to avoid singularities and numerical issues. The value a can be selected in two different ways. On the one hand, the user can be supplied with cross-validation results on model simplicity and performance for all possible values. He or she can then pick a specific value. On the other hand, the value can also be selected automatically. In that case, the best model's Area under the ROC curve (AUC_{opt}) is determined via cross-validation. Then, the algorithm picks the value of alpha corresponding to the cross-validation performance closest to $c.\text{AUC}_{\text{opt}}$, where c is a cutoff parameter that can be set beforehand. Its default value is 0.95. The value of a will influence the trade-off between model complexity and performance. The loop continues until convergence, with a maximum of 10 iterations. If two neighbouring intervals have the same weight, the model is automatically simplified by joining them together. Yet, in extreme cases, this simplification of the variable might also result in its removal, namely, if all intervals have zero weight.

The impact of iterative reweighting on the performance and complexity of an ICS model will be demonstrated later in the application results in Section 3.4.1, Table 3.7.

3.2.4 Overview of the ICS algorithm

The general overarching ICS flow is outlined in Algorithm 3.2. Because this approach remains the same for `lpICS` and the other variant introduced in the next Chapter, the tildes have been left out. The variants only differ in the specific implementation of the data transformation, core algorithm and iterative reweighting.

Algorithm 3.2 General overview of ICS.

1: Construct $\tau^p \quad \forall\, x_p, \quad p \in [1, N_d]$
2: $\mathbf{Z} \leftarrow \varphi(\mathbf{X}, \tau)$
3: $\mathbf{w} \leftarrow \text{ICS}(\mathbf{Z})$
4: select a
5: $\mathbf{w} \leftarrow \text{ICS}(\mathbf{Z}, a)$ {see Algorithm 3.1}
6: $\mathbf{w}, \mathbf{Z} \leftarrow \text{nonEmptyVars}(\mathbf{w}, Z)$
7: **repeat**
8: $\quad n_{old} \leftarrow \#\text{vars}(\mathbf{w})$
9: $\quad$ select a
10: $\quad \mathbf{w} \leftarrow \text{ICS}(\mathbf{Z}, a)$ {see Algorithm 3.1}
11: $\quad \mathbf{w}, \mathbf{Z} \leftarrow \text{nonEmptyVars}(\mathbf{w}, \mathbf{Z})$
12: **until** $n_{old} == \#\text{vars}(\mathbf{w})$
13: $\mathbf{s} \leftarrow \text{scaleAndRound}(\mathbf{w})$
14: risk profile $\leftarrow \text{logRegress}(\mathbf{Z}.\mathbf{s}, \mathbf{y})$

- Line 1 and 2 represent the data transformation. The data are expanded to a binary representation $\mathbf{Z} \in \mathbb{B}^{N \times N_s}$ after construction of thresholds τ as explained in Section 3.2.1.
- Line 3 calls the core algorithm. It returns a real-valued sparse vector $\mathbf{w}$ containing weights for all binary variables in $\mathbf{Z}$. This was discussed in Section 3.2.2.
- The model so far can be tuned further in lines 4-5 through *iterative reweighting*. Its goal is to further simplify the model. To this end, a tuning parameter a needs to be selected, either manually, based on predictions of its outcome, or automatically. This was discussed in Section 3.2.3.
- If all weights for the expansion of an original variable x^p are zero, this variable can be eliminated. Hence, ICS has built-in feature selection. The simplification of the problem through removal of variables with zero coefficients is carried out in line 6.
- Lines 7-12 show a loop containing lines 3-6. Here, the model training is repeated with the selected variables until a stable model is obtained. Note the difference with the loop in Algorithm 3.1. There, the goals is the simplification of the model for every variable whereas here, the aim is to remove variables.
- A scoring systems consists of integer weights, but the ICS coefficient vector $\mathbf{w}$ is not necessarily integer-valued. Hence, line 13 converts it to integer score points $\mathbf{s}$ by scaling and rounding. Furthermore, this step simplifies the model by joining adjacent intervals with equal score points.

- The last line applies logistic regression to convert the scores to a risk profile, that is, it attributes a risk value to every score value.

Algorithm 3.2 yields a scoring system and associated risk profile. This information and an assessment of its performance should be communicated to the user via suitable visualization techniques and performance measures, as explained next.

3.2.5 Visualization and assessment

To discuss visualization and assessment of the model, a concrete example on synthetic data will be used. This dataset will also be used for the sensitivity study discussed further since the correct solution is known and specific effects can be tested. The model for data generation can be expressed as:

$$p = S(7x^1x^2 + 4(x^3)^2 + 3x^4 - 3) \tag{3.6}$$

in which S is the standard logistic (sigmoid) function and p the risk or probability of the observation x_i belonging to the target class. The model includes two main effects, quadratic in x^3 and linear in x^4, and one interaction involving x^1 and x^2. Data generation is done by randomly generating a pool of independent standard normally distributed data. Apart from the four required, additional non-informative variables can be added. The basic data set that will be used for the study consists of 250 observations for each class, involving seven variables (four required and three additional).

The results of applying ICS on the basic data set are presented in Figure 3.2. Two third of the data was used for training, one third for testing. The three top parts of the Figure represent the detected effects. The τ values are shown at the borders. The top effect involves x^1 and x^2. Notice the influence of the multiplication. The quadratic and linear effect were correctly detected as well, whereas the three spurious variables were correctly rejected. The number of scoring points is represented as text, but also coded using colours. Blue colours correspond to large negative numbers. The higher the score, the more the colour shifts towards yellow. This colour coding is the same for all detected effects.

The bottom part of the Figure is the Risk Profile. It maps the final score obtained by summing over all effects to the probability of belonging to the target class. This profile has its own colour scheme going from blue (no to low risk) to red (high risk).

This type of graphical visualization allows for easy application of the model to new data. First of all, it indicates which variables are important for the outcome and whether they influence the results as main effect or in an interaction.

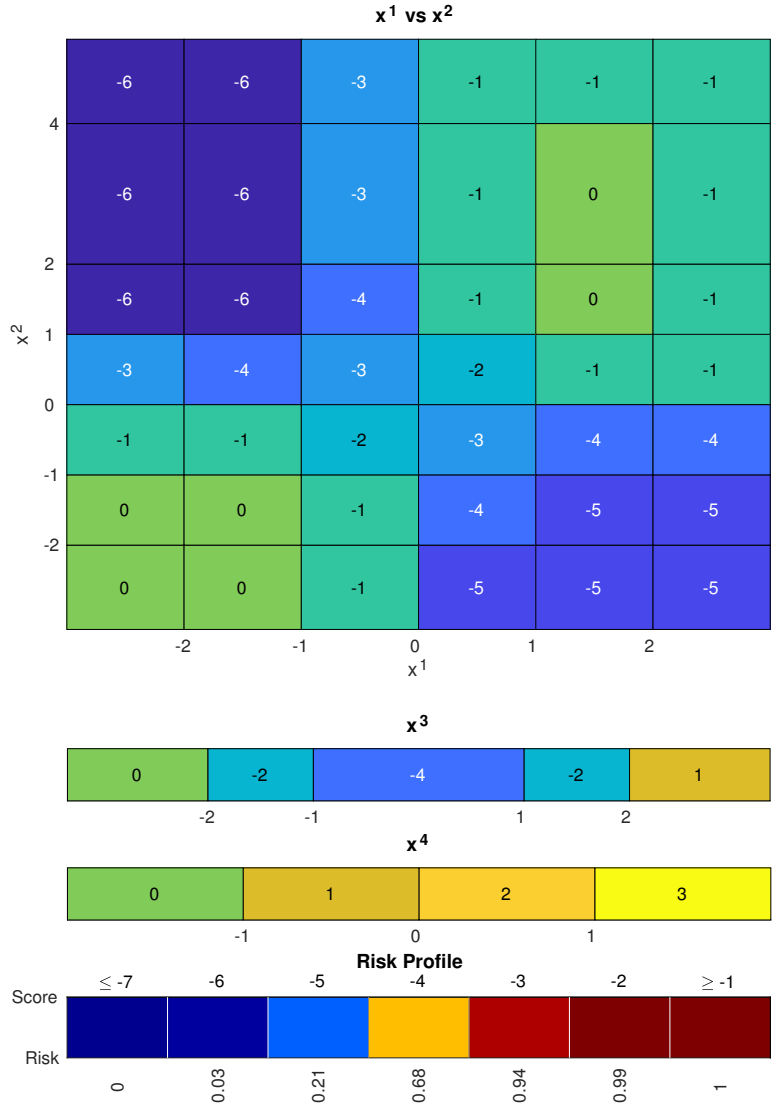


Figure 3.2: The results of the application of ICS to the basic synthetic data set. It shows an identified interaction between x^1 and x^2, a quadratic effect in x^3 and a linear effect in x^4. Each interval of each effect has been attributed a number of score points which are marked as well as coded by colour. At the bottom, the possible total scores are linked to empirical risks via the risk profile.

Table 3.1: Performance assessment of the synthetic data model in terms of accuracy (Acc), number of intervals, Area under the ROC curve (AUC) and Slope and Bias of the calibration curve for training and testing data.

	Training	**Testing**
Acc	0.870	0.865
AUC	0.942	0.938
Slope	0.992	0.974
Bias	0.004	0.001
#intervals	51	

Moreover, it also indicates which intervals are most important. If a new data sample would be generated, it would suffice to determine for each effect what interval the data is in, sum the corresponding points together and match this to a probability using the risk profile. For example, given an observation $[0.5,\ -0.12,\ 1.12,\ -0.34,\ 4.15,\ 8.36,\ -11.2]$, the interaction effect in (x^1, x^2) would yield -3, the quadratic effect in x^3 -2 and the linear in effect x^4 1. All other variables can be discarded. Hence, the score would be $-3 - 2 + 1 = -4$. Using the risk profile, this corresponds to a probability of 68% for the target class.

Assessment of the model involves three different aspects. Firstly, the actual performance can be judged either by accuracy or by the area under the ROC curve (AUC) [56]. The ROC curve is computed based on the obtained scoring system. The observations' scores are thresholded with a varying cutoff. This leads to different corresponding pairs of sensitivity and specificity when comparing to the known labels. These pairs are the points of the curve. The accuracy is subsequently obtained using the optimal cutoff, that is, the one corresponding to closest topleft point of the ROC curve. Secondly, the complexity of the model should be taken into account, because it is closely connected to its interpretability. The total number of intervals was chosen as a measure of simplicity in this work. Finally, the ability to correctly predict the probability should be judged. This can be done by means of a calibration curve, plotting empirical risk versus predicted risk. An ideal fit would have slope one and zero bias, hence these two values can be used as quality indicators for the risk profile. The results for the model displayed in Figure 3.2, indicating good performance, are shown in Table 3.1.

3.3 Sensitivity study

A model should not only be judged on its performance metrics, but also on its stability. If the data is changed only a little, e.g. due to the influence of noise, the model should remain mostly the same. To verify the impact of several influencing factors, a sensitivity study is performed on the synthetic data defined in Equation 3.6. It consists of five parts. The first part is a simple resampling via cross-validation (CV) of the model data, with and without allowing for variable interactions, to investigate the stability of the feature selection. This comes on top of the entire training process described in the previous Sections. This robustness-improving CV will then be applied on the training data for all subsequent experiments. Secondly, the influence of additive white noise will be investigated. Furthermore, the influence of the number of non-informative variables and the training set size are discussed. These last two in particular have an effect on the execution time, the last topic of the study. Unless stated otherwise, the model allows for interactions and is trained on two third of the data whereas one third will be used for testing.

3.3.1 Resampling

Resampling is performed to assess the basic stability of the model. If slightly different data is used, to what extent do the estimated effects change? The resampling is performed by 10-fold cross-validation in which each part of the data is used for testing in one fold, whereas it is used for training in all other folds. This procedure is carried out twice: once with the additional restriction that no interactions should be investigated (ICS-), and once including the option for interactions (ICS). Note that it is different from the cross-validation discussed during the ICS algorithm.

The discovered effects for the ten folds and corresponding test AUCs are shown in Table 3.2. ICS mostly succeeds at keeping the relevant effects included, both when applied with and without interactions. Of course, the interaction effect cannot be discovered with ICS- as these effects are not considered. Secondly, ICS- has less tendency to include non-informative effects. This was expected, since the number of effects to be considered and the average number of coefficients per effect are much larger for ICS. For both settings, some overfitting occurs (fold 8 for ICS-, folds 5 and 6 for ICS) as can be judged from spuriously detected effects. Unexpected selected effects in ICS often include main effects of variables 1 and 2. Taking into account the interaction between these variables, these effects can actually be informative. In other words, the main effects could be incorporated into the interaction effect. The same holds for some other effects, e.g. an interaction between variable 1 and variable 4 is considered relevant a few

Table 3.2: Resampling results for ICS with (ICS) and without (ICS-) interactions as ICS-|ICS. For each fold, the number of detected effects and test AUC (%) are indicated.

Fold	Effect [1, 2]	Effect 3	Effect 4	#Other Effects	Test AUC(%)
1	|✓	✓|✓	✓|✓	0|1	78|93
2	|✓	✓|✓	✓|✓	0|1	72|90
3	|✓	✓|✓	✓|✓	0|0	60|90
4	|	✓|✓	✓|✓	0|0	72|72
5	|✓	✓|✓	✓|✓	2|21	65|93
6	|✓	✓|✓	✓|✓	0|13	77|82
7	|✓	✓|✓	✓|✓	0|2	72|88
8	|✓	✓|✓	✓|✓	5|3	71|83
9	|✓	✓|✓	✓|✓	0|3	62|83
10	|✓	✓|✓	✓|✓	0|3	65|90

times. This still yields a good model. However, it shows that, when interactions are involved, the model is not uniquely defined.

Table 3.2 shows ICS obtains a better classifier than ICS-. Even when overfitting occurs, the variable coefficients are such that the model still yields good performance. Of course, this comes at the cost of a more complex model.

In conclusion, the results indicate that resampling can be used to improve robustness of model selection since only in a few folds detections were missed or spurious detections occurred. For a summary model, trained on all the (training) data, only the effects that occurred in more than 7 out of 10 folds will be included. The threshold for robustness is set arbitrarily for the moment. More elaborate techniques than thresholding should be used to give a statistical justification for the inclusion of an effect. Nevertheless, applying this principle for the data presented in Table 3.2 leads to inclusion of all correct effects. No spurious detected effects are included in ICS- and one main effect for variable 2 is included for ICS.

3.3.2 Influence of noise

This section investigates the impact of the noise level on the outcome model. The amount of additive white noise can be characterized by the Signal-to-Noise Ratio (SNR), defined as $\text{SNR} = \frac{\sigma_s^2}{\sigma_n^2}$, the ratio of the variances of the signal and

Table 3.3: Influence of noise on the detected effects and test AUC for ICS.

SNR	Effect [1,2]	Effect 3	Effect 4	#Other Effects	AUC
Inf	✓	✓	✓	2	91
50	✓	✓	✓	1	86
25	✓	✓	✓	2	86
10	✓	✓			80
5	✓	✓	✓		79
4	✓	✓	✓		76
3		✓	✓		63
2			✓		63
1.5		✓	✓		58

the noise. The influence of noise can be shown by comparing the models found for various SNRs. In this experiment, SNR $\in \{\infty, 50, 25, 10, 5, 4, 3, 2, 1.5\}$ will be considered. A Signal-to-Noise Ratio of ∞ corresponds to the noiseless case. Noise is added to $\mathbf{x}_i$ after setting y_i using the model described in Equation 3.6. Summarizing, the data matrices $\mathbf{Xn}$ for this experiment are generated as $\mathbf{Xn} = \mathbf{X} + \mathbf{S}$, where the elements of $\mathbf{X} \in \mathbb{R}^{500\times 7}$ are drawn from $\mathcal{N}(0, 1)$ and the elements of $\mathbf{S} \in \mathbb{R}^{500\times 7}$ are drawn from $\mathcal{N}(0, 1/\text{SNR})$. This generates a matrix $\mathbf{Xn}$ for each SNR value. The size of the matrices is explained in Section 3.2.5.

The sensitivity to noise is illustrated in Table 3.3. For high SNR, all relevant effects are detected, whereas for lower SNR, the interaction effect is lost. This is logical, since both variables are affected by the noise. Although the interaction effect is masked by the noise, ICS does not model the noise itself. The additional spurious detections for high SNR may seem surprising, but they involve variables 1 and 2, the variables also involved in the interaction. As such, they do contribute to the solution. The last column of the Table highlights the drop in performance when the noise level increases.

3.3.3 Influence of non-informative variables

Noisy conditions can also originate from a higher number of non-informative variables. Normally, ICS is able to exclude non-informative variables from the model. However, a variable can only be excluded if all of its intervals in the extended binary feature space have zero coefficients. This results in a requirement for a much sparser model when more and more non-informative variables are included. To quantify the influence of having a higher number of variables, ICS was applied on a dataset consisting of the four informative variables and an additional number of non-informative variables going from one

Table 3.4: Selected effects (interaction, two main effects, possibly other effects) and test AUC for an increasing number of additional variables (#vars).

#vars	1	2	3	4	5	6	7	8	9	10
Effect $[1,2]$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Effect 3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Effect 4	✓		✓	✓	✓		✓	✓	✓	✓
Other		1								1
AUC (%)	90	93	90	90	90	79	90	90	90	90

to ten. Table 3.4 shows the experiments most often yield a correct rejection of all non-informative variables with hardly any additionally selected effect whilst keeping the test AUC around 0.9. The two extra selected effects once again involve variables that are actually part of the model. No influence from the number of non-informative variables can be observed.

3.3.4 Influence of training set size

The performance of the model can be expected to correlate with the size of the training set. One would expect a model improvement when the training set size grows. To study this, an independent test set of 150 observations of each class is considered. The training set is enlarged gradually. Set sizes of 100, 200, 500, 1000, 1500 and 2000 with equal contribution of the two classes are considered.

The influence of the training set size is presented in Table 3.5. Even when only 100 data points are available, the three effects can be discovered. The one additional effect for a set size of 500 is related to variable x^1, which is indeed involved in the model. A closer look at the generated models for each case

Table 3.5: Influence of the training set size on the detected effects and the test AUC for ICS.

Size	Effect [1,2]	Effect 3	Effect 4	#Other Effects	AUC(%)
100	✓	✓	✓		85
200	✓	✓	✓		87
500	✓	✓	✓	1	95
1k	✓	✓	✓		94
1.5k	✓	✓	✓		94
2k	✓	✓	✓		94

reveals that when the set size grows, the number of binary features for some effects increases, particularly for the interaction. This signifies that although the correct effects are already discovered with less data, the scoring system becomes more refined when more data is added. This is due to the choice of τ. When more data is available, more and smaller intervals are considered, only limited by the user-defined maximum number of intervals. Hence, when the latter is reached, the model's maximal complexity is reached as well. The impact on performance is shown in the last column of the Table: the AUC improves with a growing data set, though even the coarsest model already has an AUC of 0.85. With the data set used for this study, the AUC saturates for a set size larger than 500 data points. No information is gained by having a larger data set. Note that these results depend on the complexity of the underlying model and the predefined maximum number of thresholds τ.

This correct detection of the effects for smaller data sets and the gradual improvement of the model until saturation when more data is available opens perspectives for large-size problems. As sometimes applied in other domains fixed-size approximations could be considered to reduce the computational load [158].

3.3.5 Execution time

The set size and the number of non-informative variables influence not only the performance of the model in terms of accuracy, but also the problem size. Depending on the method used to solve Equation 3.4, this can have an impact on execution time. To quantify this, 100 executions of the optimization problem given in Equation 3.4 were performed with the training sizes and additional variables as described in the previous sections. Furthermore, the impact of iterative reweighting is assessed by sketching the evolution for the non-weighted and the reweighted case.

Figure 3.3 shows an exponential increase in the execution time for the weighted and unweighted case for an increasing number of non-informative variables, as compared to only a nearly linear increase for the size of the data set. Hence, the impact of the number of non-informative variables is dominant over the impact of the training set size. This is due to the combinatorial expansion of the feature space implied by the mapping defined in Equation 3.1, whereas an increase in set size yields a same increase in data constraints in Equation 3.4. The linear programming problem is solved using a standard primal-dual approach. This explains the exponential and linear dependencies shown in Figure 3.3.

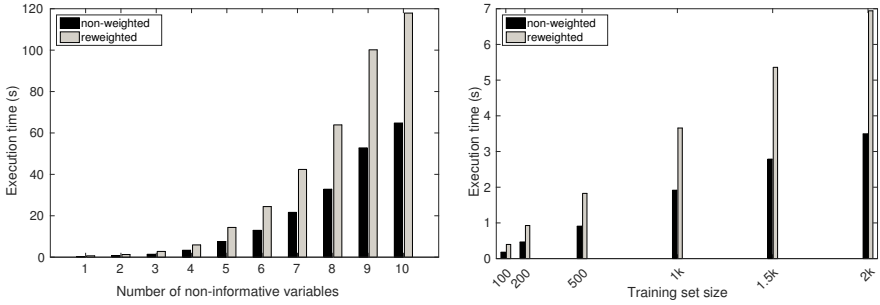


Figure 3.3: Average ICS core execution time as a function of the number of non-informative variables (left) and training set size (right).

3.4 Applications

This Section validates `lpICS` through its application on a range of real-life datasets taken from the Machine Learning repository of the University of California, Irvine (UCI) [95] introduced in Chapter 2, Section 2.5. A first test is performed on a dataset shown to be perfectly described by a small rule system to show ICS is capable of discovering these rules. Next, a short study demonstrates the impact of iterative reweighting. Finally, the results on various biomedical datasets are discussed. For these datasets, ICS will be run with its standard settings, that is, without user interaction to select the reweighting parameter a. Stratified random sampling selects two thirds of the data for training and one third for testing.

3.4.1 Results

Detection of rules on the Mushrooms dataset

Perfect classification has been obtained on this dataset using a simple rule system [53]. The obtained disjunctive rule set giving the solution of the mushroom set is described in Table 3.6. ICS correctly discovers the first two rules, leading to an AUC of 0.994 and an accuracy of 99.4%. The other two rules were not discovered for two reasons. Firstly, since the dataset is large, interactions were not considered which makes it impossible to discover conjunctive rules such as the last two in the Table. Moreover, even if interactions are considered, the default setting limits them to two-way interactions. Here, an explicit request for the three-way interaction should have been formulated. For this single interaction, taking into account the number of categories in the variables involved, it would lead to a 324-dimensional binary vector with a single

Table 3.6: Rules for the mushroom data set leading to perfect classification [53].

Rule
1. odor = NOT(almond OR anise OR none)
2. spore-print-color = green
3. odor = none AND stalk-surface-below-ring = scaly AND stalk-color-above-ring = brown
4. habitat = leaves AND cap-color = white

1. This is such a detail that it would be discarded. Secondly, the automatic reweighting select the parameter a using the cutoff on the AUC, as discussed in Section 3.2.3. Since it selects a model with a fraction of the best AUC, a perfect model cannot be obtained. The full rule set would involve six instead of two variables for an improvement of about 0.6% in classification accuracy.

The obtained result can be compared to other machine learning algorithms such as Naive Bayes, linear and nonlinear (RBF kernel) Least-Squares Support Vector Machines [158]. These have been chosen as they are widely accepted techniques. Decision Trees (DT) have been included as an alternative interpretable classifier. They all obtain close to optimal models in terms of AUC, depending on how well they are able to model a rule set.

- Naive Bayes and linear SVM perform well, but less than ICS, with AUCs of respectively 0.954 and 0.964.
- The nonlinear SVM finds an optimal model, but at the cost of interpretability.
- The decision tree model obtains a perfect separation with 8 splits involving seven variables.

Impact of reweighting

Before presenting the full validation on several datasets, an example will clarify the impact iterative reweighting can have. To this end, `lpICS` with and without reweighting was applied to the breast cancer dataset. Interactions were disabled and tuning was performed without user intervention, based on the default cutoff $c = 0.95$, discussed in Section 3.2.3. As described earlier, two third of the data was randomly selected as training data, the remaining third is used for assessment. Table 3.7 shows the results. Reweighting leads to a much simpler model. Due to its iterative nature, it allows many additional variables to be discarded. Furthermore, it also reduces the number of intervals per variable,

Table 3.7: The effect of `lpICS` reweighting on the breast cancer dataset. Complexity is reported as the number of selected variables and the total number of variable intervals. Training and test results are given as the Area Under the Curve (AUC), the accuracy (Acc) and the slope and bias of the calibration curve.

Without reweighting			
23 vars		68 intervals	
Training results			
AUC	1	Acc	1
Slope	1	Bias	0
Test results			
AUC	0.989	Acc	0.890
Slope	0.762	Bias	0.019

With reweighting			
2 vars		5 intervals	
Training results			
AUC	0.958	Acc	0.958
Slope	1	Bias	0
Test results			
AUC	0.933	Acc	0.916
Slope	0.893	Bias	0.030

albeit only moderately in this case: from an average of about 3 to an average of 2.5 per variable. The training and test results show that overfitting occurred without reweighting which is not surprising with the large number of variables. Training results are perfect, a typical indication of overfitting. This influences the test results, particularly leading to a low calibration slope. In contrast, applying reweighting leads to a much simpler model consisting of only two variables. A comparison of the training and test results show the accuracy and calibration slope are closer together, indicating a better generalization. The test AUC is lower than without reweighting, but the accuracy has improved. The simpler model also leads to a better calibration slope, although the bias increases with 0.011.

Biomedical datasets

The validation results of `lpICS` through application on seven biomedical datasets are displayed in Table 3.8. As for all tests in this Section, stratified random sampling selected two thirds of the data for training and one third for testing. No resampling as in the sensitivity study was performed.

The inflammation and nephritis case of the Acute Inflammation dataset prove very simple: with only six, resp. eight, intervals perfect classification and risk prediction is obtained. This was already expected since it had been shown a rule-based expert system could classify them perfectly [44]. Similarly as for the mushroom set, `lpICS` is able to correctly extract these simple rules underlying

Table 3.8: Results of the application of `lpICS` on the Acute inflammation ('Inflammation' and 'Nephritis' labels), Breast Cancer Diagnosis, Cardiotocography ('Cardio'), Chronic kidney disease ('Kidney'), Indian Liver Patient data ('Liver') and Vertebral Column ('Vertebral') datasets. Performance is measured by test accuracy ('Acc') and AUC on the test data. Risk prediction evaluation is quantified via the Slope and Bias of the calibration curve. Model complexity is captured by the number of Intervals and Variables(I (V)).

	Acc	**AUC**	**Slope**	**Bias**	**I(V)**
Inflammation	100	1	1	0	6(3)
Nephritis	100	1	1	0	8(4)
Breast Cancer	91.6	0.933	0.893	0.03	5(2)
Cardio	91.1	0.945	0.948	-0.007	11(5)
Kidney	91.5	0.932	0.947	0.053	4(2)
Liver	67.9	0.677	0.443	0.399	60(9)
Vertebral	82.5	0.782	0.977	-0.022	4(2)

the classification.

The Breast Cancer, Cardiotocography and Chronic Kidney disease data are modelled accurately as well: the accuracies are all higher than 91% and the AUC values are close to the optimal value of one as well. Still, the risk assessment could be improved. The more complex model for Cardiotocography is an advantage since it combines a good slope with a bias lower than one percent. For the two other sets, the simpler models lead to a systematic overestimation of the risk with three to five percent.

Results on the Indian Liver Patient Data are not very good. The high number of 60 intervals points to overfitting, nine out of ten variables were selected for the model. This is further backed up by the fact that the training AUC is 0.86, significantly higher than the test AUC of 0.677. As a result, the risk prediction also performs poorly with a slope of around 0.44 and a systematic overestimation of the risk with 40%.

Finally, the vertebral column data has been modelled decently, particularly when looking at the good slope and bias. Some additional experiments with manual guidance of the algorithm showed that a simpler model can be achieved using only one variable, but also with lower performance and worse risk prediction capabilities. Alternatively, a more complex model can be trained yielding higher accuracy and AUC, but this has a large negative impact on the slope and bias. This shows that a good trade-off has been found within the boundaries of ICS's modelling capabilities.

These results only show that ICS is capable of building good models for most of the biomedical problems discussed here. After introduction of an alternative ICS method, the next Chapter provides a full statistical comparison of both ICS approaches with established machine learning methods.

3.5 Discussion

Interval Coded Scoring aims at binary classification problems where interpretability is of primary importance. Its strength lies in the fact that it derives these models with a data-driven approach rooted in well-established sparse optimization theory rather than relying on expert knowledge only. Yet, it does not blindly analyse the data either. Domain experts can guide the training process in the reweighting stage. They are presented with an overview of model characteristics based on cross-validation on the training data. This allows them to determine the trade-off between complexity and performance. Hence, the main responsibility remains with the domain expert. Alternatively, a cut-off on the performance can be defined to automatically select a suboptimal, simpler model.

Another strength lies in the representation of the model. The interpretable colour-coded diagrams can easily be applied manually and provide a quick overview of the patterns in important effects. Furthermore, the proposed implementation offers a broad view on model performance. It focuses not only on standard performance measures as accuracy and AUC, but also assesses a model's ability for risk prediction through the calibration slope and bias, as well as the model complexity via the total number of variable intervals.

Compared to current scoring systems, more complex models are possible due to the introduction of interactions. ICS can be applied for interactions of any dimensionality in theory, but in standard use, one should limit it to binary interactions since they can be represented more easily, as shown in Figure 3.2. Moreover, as became clear in the applications, interactions are relatively rare in biomedical applications. However, they do appear in other fields. A well-known case is the XOR problem in logic circuits.

The application Section discussed how ICS is able to extract the essence of systems characterized by logic rules as was the case for the mushroom and acute inflammation data sets which primarily consist of categorical and binary variables. This easy characterization stems from the fact that it approaches variables differently based on their type. E.g. continuous or ordinal variables are discretized through intervals, whereas this is not necessary for binary and categorical data. This is an advantage over e.g. basic linear classifiers.

The interpretability of ICS comes with a price. Due to the discrete representation of the data, variable patterns are restricted to a certain level of granularity. For example, linear effects can only be detected up to the level of the intervals, that is, as piecewise constant. In contrast, on this level, any linear or nonlinear effect can be modelled, e.g. quadratic or even sinusoidal. This limited power in representation contrasts with more powerful techniques such as SVMs or neural networks known as universal approximators. One could of course argue that in a limit case, when enough data is available, infinitely many intervals could be defined, leading to a perfect representation. In practice, this would be unstable, make the problem infeasible to solve and lose the advantage of interpretability.

Another disadvantage that was touched upon is the computational cost. The LP formulation runs relatively slowly for larger problems and due to the data transformation the execution time grows exponentially with a growing number of variables. As seen in Section 3.2.2 both the number of variables and constraints depend on the defined differences between intervals, growing with the number of variables, so solving the primal or dual LP problem scales badly. This could be remedied by an approach similar to the kernel trick in SVMs, where a quadratic dual problem can be solved which only depends on the number of observations N. In that case, the dimension of the feature space would become irrelevant. However, the use of the ℓ_1 norm and the matrix D defining the differences lead to a more difficult dual formulation of the problem. For most common application purposes, this need not be a problem, as long as the data structures still fit in memory and time allows a slow execution.

A final drawback that was already mentioned is the non-uniqueness of ICS. This can be due to two reasons: the inherent non-uniqueness of the model itself and the mathematical non-uniqueness of Equation 3.4.

The *model uniqueness* issue cannot be solved, it is inherent to the model. ICS constructs additive models. As such, the same outcome can be achieved in multiple ways. For example, an interaction can sometimes be modelled as two main effects and any two main effects can always be modelled as an interaction. Hence, the model is not unique in the sense that an alternative model can be constructed with the same performance. However, in many cases this will be avoided due to the total variation penalty: an interaction leads to higher total variation and will therefore mostly be avoided if possible. A reason why the problem might still arise is the iterative nature of the training process. For example, an interaction is selected because a main effect was already removed in a previous step. Notice this problem only arises if interactions are considered.

Mathematical uniqueness depends on the convexity of Equation 3.4. The Equation potentially has multiple solutions, even when both total variation and performance as quantified by the slack variables is taken into account. Therefore, the system is not strictly convex and the solution is not always

unique, depending on the specific data. Imagine a problem with two strongly correlated variables, whose intervals carry approximately the same information. Equation 3.4 has no way of choosing one of the two. It could lead to a model based on one variable or the other, both could even be included with lower coefficients. Hence, in this case, mathematical uniqueness is not guaranteed. If one would aim at this uniqueness e.g. for perturbed repeated runs and comparisons of the resulting scoring systems, additional steps should be taken. This will be further discussed in the next Chapter.

To finish, it should be mentioned that the lack of uniqueness might seem unsatisfactory from a programming point of view. Yet, it leaves space for discussion with medical practice, where, in the end, the interpretation will take place.

3.6 Conclusion

This Chapter introduced Interval Coded Scoring as a technique to train interpretable models for binary classification problems. Although mathematically similar, it deviates from SVMs by its data transformation, penalization term and training process, which uses iterative reweighting to obtain a model as simple as possible. Obtained models can include interactions of any dimensionality, but only approximated in a piece-wise fashion. A trained model is represented graphically for easy interpretation and can be assessed on the basis of its performance, complexity and risk calibration.

A sensitivity study showed the ability of ICS to cope with gaussian noise and non-informative variables. Moreover, it was observed how the model detail can be improved if more data is available and how robustness can be increased by resampling of the data.

Several applications validated the viability of ICS for use in real-life situations. Rule systems can be extracted easily, but other datasets are mostly modelled correctly as well. Further improvement could be possible through manual guidance of the training based on domain knowledge.

Throughout the Chapter, model uniqueness was discussed, but this cannot be remedied. Also, execution time and mathematical non-uniqueness were encountered as potential problems. The next Chapter will investigate approaches to address these challenges by describing extensions to `lpICS`. Additionally, it will extend the application Section with these new approaches and include a comparison with other machine learning algorithms.

Magis: Originally a Latin adverb that meant 'more' or 'to a greater degree', it is now commonly used to denote a key element of Ignatian spirituality.

B. Geger sj, *What Magis Really Means and Why It Matters*

Chapter 4

Extensions for larger problems

This Chapter investigates solutions for the problems encountered in the previous Chapter. Large problems result in a high computational and memory load. Furthermore, the solution cannot ensure (mathematical) uniqueness. Therefore, this Chapter first proposes preselection as a dimensionality reduction scheme taking into account the interval structure of the data. Then, it shortly investigates a data transformation to an ℓ_2-form in an attempt to address the need for uniqueness, but in the current implementation, it increases the computational cost. Therefore, a second attempt starts from elastic net as an alternative formulation, leading to the approach called `enICS`*. Since it does not support interactions, two approaches are discussed in an attempt to include them: Dykstra decomposition and additional constraints. Next, an experimental section first compares the execution time and resulting models for* `lpICS` *and* `enICS`*, with and without preselection. Then, it extends the applications from the previous Chapter to the new methods and compares all ICS approaches to established classifiers. The Chapter ends with a discussion. Parts of this Chapter have been published earlier [25, 26].*

4.1 Methods

4.1.1 Preselection

The study in Chapter 3, Section 3.3.5 explained that `lpICS` is highly affected by a growing number of features. Although good models are still obtained, the execution time grows exponentially. Depending on the available solver, large problems can take a long time to finish. This is not necessarily an issue

when training a final system for use in an application since the evaluation time for new data is very fast. Yet, in case of exploratory research, the timing is more important. Moreover, if the dataset does not fit in the available memory, thrashing (moving data between the RAM and harddisk) further dramatically increases the execution time. Therefore, this Section looks at preselection, a feature selection prior to the execution of the full ICS algorithm, to reduce that size of the dataset.

This reduction of the dimensionality of the feature space not only reduces the training time, but also the risk of overfitting. It can be achieved in multiple ways. Firstly, the aim could be feature extraction, that is, a lower number of new features can be constructed from the existing set. In that case, a typical approach is Principal Component Analysis which tries to compress the variance into a lower-dimensional space by performing consecutive orthogonal projections [126].

However, if interpretability is important, feature selection is a more appropriate option: the least informative features should be discarded. Typically, the best features are discovered through either a filter or wrapper approach. In the former case, a stand-in function is used to judge a feature's discriminative ability. Many filters, though not all, are univariate, that is, discriminative interactions between features are not considered. Examples of filters include a statistical test or information criterion. In the latter case, subsets of features are tested through cross-validation with the intended classifier. This is a more computationally expensive, but also a more realistic estimate of features' discriminative abilities. Furthermore, it is not limited to individual features but judges subsets as a whole [37].

`lpICS` falls into a third category, embedded feature selection, since it selects features as part of its training process. The proposed preselection process is an additional procedure akin to a wrapper approach. Yet, although it takes into account the interval structure, it does not solve the full ICS optimization problem. Instead, it is performed as a four-step procedure:

1. The `lpICS` data transformation in Equation 3.1 (p. 25) is performed on the initial dataset $\mathbf{X}$. Both main effects and interactions can be modelled as desired. Using the computed $\tilde{\mathbf{Z}}$ and the known labels, a standard Least Squares Support Vector Machine classification problem with linear kernel yields coefficients $\boldsymbol{\omega}$.

2. A new data set $\mathbf{X}_2$ is constructed. Main effects are modelled as $\mathbf{X}_2^p = \tilde{\mathbf{Z}}^p \boldsymbol{\omega}^p$, $\forall\ p = 1, \ldots, N_d$. This replaces the variable values in $\mathbf{X}$ by the coefficients of their intervals. Hence if only main effects are involved, $\mathbf{X}_2 \in \mathbb{R}^{N \times N_d}$ has the same size as $\mathbf{X}$. Each considered interaction increases the dimensionality of $\mathbf{X}_2$ with one. These additional columns are given as

$\tilde{\mathbf{Z}}^{p_1,p_2}\boldsymbol{\omega}^{p_1,p_2}$. This approach can be considered as a data-driven discrete non-linear transformation. $\mathbf{X}_2$ encodes data trends for each variable and interaction.

3. The goal remains to eliminate certain variables. To obtain this, any feature selection method could be applied on the trend-informed data $\mathbf{X}_2$. Here, elastic net was chosen. Similarly to the iterative reweighting discussed in the previous Chapter's section 3.2.3, the selection is semi-automatic: the user can decide on the sparsity based on cross-validated performance estimation.

4. Finally, the elastic net indicates which columns of $\mathbf{X}_2$ should be kept. Since every column corresponds to one main effect or interaction, these columns correspond to the set of main effects and interactions that ICS should consider. This information can subsequently be used when setting up the actual ICS problem as described in Appendix A.

4.1.2 Uniqueness: a quadratic formulation

The discussion at the end of Chapter 3 made a distinction between two kinds of uniqueness. Model uniqueness will not be discussed further, since it cannot be solved. However, the mathematical uniqueness stems from the concrete optimization problem being solved. This can be enforced by a different objective function that only allows for a single solution. For example, when dealing with correlated informative variables, a possible solution could be to always include all of them, with lower coefficients for both.

A possible approach to achieve this kind of uniqueness is a transformation to a quadratic problem as suggested in literature by Sra [153]. His work focuses on a representation for the ℓ_1 SVM which is identical to the original SVM except that the ℓ_2 penalty is replaced by ℓ_1. This, in turn, is similar to `lpICS` which uses this penalty on $\mathbf{D}\tilde{\mathbf{w}}$ instead of on $\tilde{\mathbf{w}}$ alone, as shown in the LP formulation of the `lpICS` core in Equation 3.5. More generally, Sra describes an approach that can be applied to any LP problem. To facilitate the discussion, the LP formulation of ICS can be simplified with the introduction of two grouping variables $\mathbf{g} = [\tilde{\mathbf{w}}; b; \mathbf{f}; \boldsymbol{\varepsilon}] \in \mathbb{R}^{N_s+1+N_{ds}+N}$ and $\mathbf{c} = [\mathbf{0}; 0; \mathbf{1}; \gamma\mathbf{1}]$ with corresponding sizes for the subvectors. Moreover, all constraints (N data constraints, $2N_{ds}$ sparsity constraints and N slack positivity constraints) can be summarized using a coefficient matrix $\mathbf{A} \in \mathbb{R}^{(2N+2N_{ds})\times(N_s+1+N_{ds}+N)}$ as follows, using block matrices of zeros and ones of appropriate sizes:

$$\mathbf{A} = \begin{bmatrix} -\mathbf{D} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{D} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{Y}\tilde{\mathbf{Z}} & \mathbf{Y1} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{4.1}$$

The right-hand side corresponding to this coefficient matrix is $\mathbf{d} \in \mathbb{R}^{2N+2N_{ds}} = [\mathbf{0}; \mathbf{0}; \mathbf{1}; \mathbf{0}]$ with the same block division as in matrix $\mathbf{A}$. Given these notations, `lpICS` is cast in the general LP formulation in matrix notation as:

$$\begin{aligned} \min_{\mathbf{g}} \quad & \mathbf{c}^T\mathbf{g} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{g} \geq \mathbf{d} \end{aligned} \tag{4.2}$$

For this type of problems, the theorem introduced by Mangasarian [106] can be applied:

Theorem 1. *If Equation 4.2 has a solution, then there exists an $\epsilon_0 > 0$, such that for all $\epsilon \leq \epsilon_0$*

$$\operatorname*{argmin}_{\mathbf{g}\in G} \frac{1}{2}\|\mathbf{g}\|_2^2 + \frac{1}{\epsilon}\mathbf{c}^T\mathbf{g} = \operatorname*{argmin}_{\mathbf{g}\in G^*} \|\mathbf{g}\|_2^2 \tag{4.3}$$

where G is the feasible set for Equation 4.2 and G^* is the set of optimal solutions to Equation 4.2. The minimizer of Equation 4.3 is unique [153].

Adding an ℓ_2 penalty term to the `lpICS` problem makes it quadratic and strictly convex. A small value of ϵ gives a greater weight to the (original) linear part of the objective function. The theorem states that once this ϵ value is small enough, the solution $\mathbf{g}$ of the quadratic problem will be optimal for the linear problem as well. Moreover, among all non-unique possible solutions, it will be the one with the smallest ℓ_2 norm.

The variables and constraints for the linear and quadratic formulation given by $\mathbf{A}\mathbf{g} \geq \mathbf{d}$ remain the same. Using common solvers available in Matlab and CPLEX, the quadratic problem of the same size takes longer to solve. Therefore, the uniqueness comes at a cost.

Yet, the computational cost can potentially be reduced by taking into account the sparse structure of the quadratic problem. In particular, Hildreth's algorithm has been suggested as a way to solve this kind of systems efficiently [46]. It is a primal-dual approach alternating between enforcing individual constraints and updating $\mathbf{g}$. Concretely, one can define the dual problem to Equation 4.3 as:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \frac{1}{2}\|\mathbf{A}^T\boldsymbol{\beta} - \frac{1}{\epsilon}\mathbf{c}\|_2^2 - \mathbf{d}^T\boldsymbol{\beta} \\ \text{s.t.} \quad & \boldsymbol{\beta} \geq \mathbf{0} \end{aligned} \tag{4.4}$$

In this equation, $\boldsymbol{\beta} \in \mathbb{R}^{2N+2N_{ds}}$ contains the dual variables linked with the constraints. The primal and dual solutions $\mathbf{g}_{sol}$ and $\boldsymbol{\beta}_{sol}$ are linked as follows:

$$\begin{cases} \mathbf{g}_{sol} = \mathbf{A}^T\boldsymbol{\beta}_{sol} - \frac{1}{\epsilon}\mathbf{c} \\ \boldsymbol{\beta}_{sol}^T(\mathbf{A}\mathbf{g}_{sol} - \mathbf{d}) = 0 \end{cases} \tag{4.5}$$

These equations can subsequently be used in Hildreth's iterative procedure by cycling through all constraints:

1. Project $\mathbf{g}$ on the chosen constraint and update the corresponding element of $\boldsymbol{\beta}$ whilst ensuring its positivity.
2. Update $\mathbf{g}$ using the new value for $\boldsymbol{\beta}$ via Equation 4.5.
3. Select the next constraint and continue until convergence.

The iterations consist of simple operations, such as a dot product for projection in the first step and matrix-vector multiplication for the update of $\mathbf{g}$ in the second step. Although the vectors and matrices involved are large, their sparsity can be exploited for improved speed. Yet, the number of constraints (and hence iterations) is in the order of magnitude of N_{ds}, growing exponentially with an increasing number of variables if interactions are involved. Moreover, although current research claims good results with this method [153], the final stage of the convergence process has also been reported to be slow [46]. This, combined with implementation issues, have lead to abandoning this approach to ICS in favour of a new formulation based on elastic net, discussed next.

4.1.3 Uniqueness and speed: elastic net formulation

Although the quadratic approach solves the mathematical uniqueness problem, the execution speed's dependency on the dimensionality was not solved. This Section therefore addresses both by the introduction of a slightly different formulation based on elastic net, `enICS`.

Before continuing, repeating the previous Chapter's remark on the naming conventions might be useful. Both `lpICS` and `enICS` implement the general ICS flow outlined in Algorithm 3.2 (p. 30). In this algorithm, the overarching concepts are used. For example, $\mathbf{w}$ is the coefficient vector, $\varphi(\cdot)$ the data transformation and $\mathbf{Z}$ the generated binary matrix of transformed data. The specific implementations of these concepts are indicated by accented equivalent representations. Recall from the previous Chapter that tilde is used to refer to `lpICS` variables e.g. $\tilde{\mathbf{w}}$, $\tilde{\mathbf{Z}}$. Similarly, `enICS` equivalents will be indicated by a hat accent e.g. $\hat{\mathbf{w}}$, $\hat{\mathbf{Z}}$.

In `enICS`, as in `lpICS` the sparsity is not induced on $\mathbf{w}$, but on the difference $\mathbf{Dw}$. However, this is done implicitly. The `enICS` data transformation and coefficients can be derived from their counterparts used by `lpICS`. Using the `lpICS` notation, the `enICS` optimization problem can be written as:

$$\min_{\tilde{\mathbf{w}},b} \|\tilde{\mathbf{Z}}\tilde{\mathbf{w}} + b - \mathbf{y}\|_2^2 + \epsilon\|\mathbf{D}\tilde{\mathbf{w}}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{D}\tilde{\mathbf{w}}\|_1 \leq t \tag{4.6}$$

ϵ is a parameter fixed at 0.05. It provides the stability of elastic net to the LASSO formulation [164]. The corresponding ℓ_2 penalty enforces the mathematical uniqueness. The hyperparameter t defines the trade-off between sparsity (model simplicity) and performance. Note that the modelling is posed as regression rather than classification. Instead of minimizing the errors with respect to the classification border, it explicitly aims at the minimization of the deviation from the label values 1 and -1.

The advantage of using this approach becomes apparent when using a slightly different data representation converting it to a standard elastic net formulation, the actual `enICS`:

$$\min_{\hat{\mathbf{w}},b} \|\hat{\mathbf{Z}}\hat{\mathbf{w}} + b - \mathbf{y}\|_2^2 + \epsilon\|\hat{\mathbf{w}}\|_2^2 \quad \text{s.t.} \quad \|\hat{\mathbf{w}}\|_1 \leq t \tag{4.7}$$

Conceptually, $\hat{\mathbf{Z}} = \tilde{\mathbf{Z}}\mathbf{D}^{-\mathbf{1}}$ (if the inverse exists), although it is never calculated explicitly, but results from direct construction as discussed further. Therefore, Equation 4.7 directly optimizes the coefficient differences $\hat{\mathbf{w}}$ and later reconstructs the interval coefficients from them. As another consequence, the iterative reweighting formulation changes slightly as well, which is also discussed below.

The new data representation and the reduction to a standard elastic net are useful for two reasons. Firstly, the addition of the ℓ_2-norm penalty yields the uniqueness properties due to the strict convexity of the elastic net on top of the sparsity characteristics of LASSO. More than `lpICS`, `enICS` aims for a grouping effect: if several correlated variables are present, it does not pick one at random, but treats them all equally.

Secondly, it has been shown that the elastic net is equivalent to the dual formulation of a squared-hinge loss support vector machine [184]. The reasoning is as follows. As a first step, one can divide both the constraint and the objective by t without affecting the outcome. Furthermore, one expects the constraint to be tight, since sparsity is the goal. Hence, the equation becomes:

$$\min_{w} \|\mathbf{Y}\hat{\mathbf{Z}}\hat{\mathbf{w}}/t - \mathbf{1}/t\|_2^2 + \epsilon\|\hat{\mathbf{w}}/\mathbf{t}\|_2^2 \quad \text{s.t.} \quad \|\hat{\mathbf{w}}/t\|_1 = 1 \tag{4.8}$$

Note that the bias term b has been absorbed into an extended $\hat{\mathbf{w}}$ and the influence of the labels is indicated by premultiplication with the diagonal matrix $\mathbf{Y}$, taking into account that $\mathbf{Yy} = \mathbf{1}$. Next, the data dimensionality is doubled and the corresponding coefficients are required to be positive. With new data [$\mathbf{Y}\hat{\mathbf{Z}}$, -$\mathbf{Y}\hat{\mathbf{Z}}$], the coefficient vector $\hat{\mathbf{w}}$ should double in size as well. The new coefficient vector (incorporating the scaling by t) is called $\mathbf{q}$. $\mathbf{q}$ is strictly positive. This does not change the solution: the first part of $\mathbf{q}$, corresponding to $\mathbf{Y}\hat{\mathbf{Z}}$, yields positive contributions. The second part, corresponding to -$\mathbf{Y}\hat{\mathbf{Z}}$, has negative contributions. Due to this approach, the ℓ_1 norm becomes a simple

sum. This yields:

$$\min_{\mathbf{q}\geq 0} \|[\mathbf{Y}\hat{\mathbf{Z}},\ \text{-}\mathbf{Y}\hat{\mathbf{Z}}]\mathbf{q} - \mathbf{1}/t\|_2^2 + \epsilon\|\mathbf{q}\|_2^2 \quad \text{s.t.} \quad \sum \mathbf{q} = 1 \tag{4.9}$$

At this point, one can construct a new dataset $\{\mathbf{C}, \mathbf{l}\}$ such that $\mathbf{LCq} = [\mathbf{Y}\hat{\mathbf{Z}},\ \text{-}\mathbf{Y}\hat{\mathbf{Z}}]\mathbf{q} - \mathbf{1}/t$, in which $\mathbf{L}$ is the diagonal version of $\mathbf{l}$. It can be verified that $\mathbf{C} = [\mathbf{Y}\hat{\mathbf{Z}} - 1/t \quad \mathbf{Y}\hat{\mathbf{Z}} + 1/t]$ and $\mathbf{l} = [\mathbf{1}\ \text{-}\mathbf{1}]$ satisfy these conditions. This transformation finally yields the following formulation:

$$\min_{\mathbf{q}\geq 0} \|\mathbf{LCq}\|_2^2 + \epsilon\|\mathbf{q}\|_2^2 \quad \text{s.t.} \quad \sum_i q_i = 1 \tag{4.10}$$

Now, the dual formulation of a soft-margin SVM with squared hinge loss for $\mathbf{C}, \mathbf{l}$ can be expressed as:

$$\min_{\alpha\geq 0} \|\mathbf{LC}\boldsymbol{\alpha}\|_2^2 + \varepsilon \sum_i \alpha_i^2 - 2\sum \alpha \tag{4.11}$$

Implicitly, one can add the constraint $\sum \alpha_i = \sum \alpha_i^*$, where $\boldsymbol{\alpha}^*$ is the solution of the dual SVM. This is trivially satisfied at the solution. In that case, the last term in Equation 4.11 can be dropped, since it becomes a constant. Overall, this leads to the final dual SVM formulation as:

$$\min_{\alpha\geq 0} \|\mathbf{LC}\boldsymbol{\alpha}\|_2^2 + \varepsilon\|\boldsymbol{\alpha}\|_2^2 \quad \text{s.t.} \quad \sum_i \alpha_i = \sum \alpha_i^* \tag{4.12}$$

Comparing Equations 4.10 and 4.12, it is clear that it suffices to solve the SVM and divide the obtained dual variables α by $\sum \boldsymbol{\alpha}^*$ to find the solution to the original elastic net problem, except one still needs to transform $\mathbf{q}$ back to $\hat{\mathbf{w}}$.

The SVM can be efficiently solved in the primal [39] with a complexity related to the number of observations N rather than the size of the expanded feature space N_{ds} which is usually greater for ICS. It makes the problem complexity independent of the size of the feature space and allows for much faster execution. Furthermore, it automatically solves either the primal problem or the dual problem, depending on its characteristics e.g. in case of a large dataset. This approach is called SVEN (Support Vector Elastic Net) [184].

4.1.3.1 Data transformation

It was mentioned that $\hat{\mathbf{Z}}$ is the result of direct construction from the data and is not calculated via the inverse of the difference matrix $\mathbf{D}$. Indeed, an `enICS` data transformation $\hat{\varphi}$ can be defined as follows for a single observation $\mathbf{x}_i$:

$$\begin{aligned} &\hat{\mathbf{z}}_i = [\hat{\mathbf{z}}_i^1, \hat{\mathbf{z}}_i^2, \ldots, \hat{\mathbf{z}}_i^{N_d}] \\ &\text{where} \quad \hat{\mathbf{z}}_i^p = \hat{\varphi}^p(x_i^p) = [1, I(x_i^p \geq \tau_1^p), I(x_i^p \geq \tau_2^p), \ldots, I(x_i^p \geq \tau_{N_p-1}^p)] \end{aligned} \tag{4.13}$$

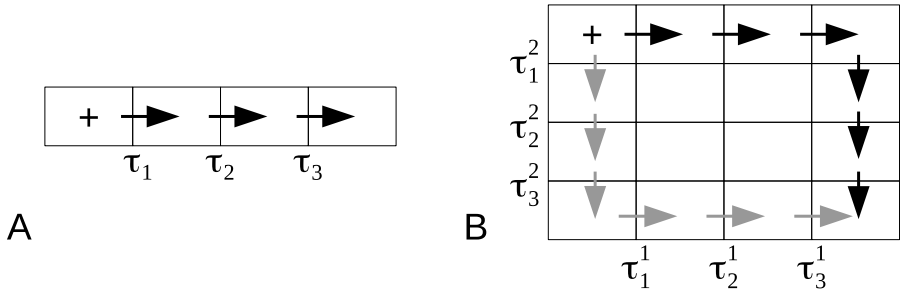


Figure 4.1: Illustration of the data transformation for `enICS`. Part A shows a computation path for a main effect, part B two possible paths for an interaction. The plus sign indicates the origin with fixed zero coefficient. The arrows indicate summing of the weight differences between intervals.

This transformation uses the same intervals as `lpICS` with the same thresholds τ. However, $\hat{\varphi}$ uses more than a single 1 to encode an observation. If a value corresponds to a certain interval, all previous intervals are filled with 1. Hence, the coefficients no longer reflect the impact of an interval, but a difference with respect to the previous interval. Of course, in the end, one is still interested in the importance of the intervals themselves, given by $\tilde{\mathbf{w}}$. This is easily calculated from the `enICS` coefficients by cumulative summing for each variable encoded in a reconstruction matrix $\mathbf{R}$, such that $\tilde{\mathbf{w}} = \mathbf{R}\hat{\mathbf{w}}$.

As a drawback, Equation 4.13 no longer contains interaction terms. This stems from the fact that for interactions the `enICS` data transformation is not uniquely defined. Figure 4.1 clarifies this with an example for a main effect and interaction. `enICS` aims to directly estimate the differences between intervals. This is stored in $\hat{\mathbf{w}}$. Eventually though, any ICS model is expressed with the scores for the intervals given by the `lpICS` coefficients $\tilde{\mathbf{w}}$. The latter can be computed from the former by cumulative summation over the intervals of every variable. On the left of the Figure, the $\tilde{\mathbf{w}}$ coefficient for an interval is calculated for a single variable using the differences stored in $\hat{\mathbf{w}}$ by summing according to the arrows starting from the origin indicated by a '+' symbol. This origin always has a fixed coefficient value of zero. The right part of the Figure shows the problem with interactions: there are several paths to compute an interval coefficient which should all yield the same result. If the transformed data $\hat{\mathbf{Z}}$ would be encoded via a single path e.g. always down first followed by right, as in the gray path, other differences would not be taken into account for the optimization so different paths would yield different values. As a result, the ℓ_1 penalty would no longer be a total variation minimization.

Hence, `enICS` does not support interactions. Two unsuccessful attempts to alleviate this by Dykstra decomposition or constraints are discussed next.

4.1.3.2 Interactions via Dykstra decomposition

Equation 4.6 expressed `enICS` in terms of the data transformation of `lpICS`, with an explicit difference matrix $\mathbf{D}$. As was discussed in the previous Chapter, this matrix encodes differences for interactions of an arbitrary number of variables as well, though binary interactions will continue to be used as examples. An elastic net with a penalty term on a product as in Equation 4.6 is called a generalized elastic net, similar to the generalized LASSO [165]. More specifically, when the product matrix encodes differences in two dimensions, it is called *2D fused* LASSO or elastic net, applicable to e.g. image denoising [81].

One way of dealing with it is a decomposition in two 1D fusion problems using the proximal Dykstra method [81]. The method is described for matrix identification e.g. image denoising, but can be adapted for the problem at hand. Mathematically, the goal is to solve the following decomposition:

$$\begin{cases} \min_{\tilde{\mathbf{w}},b} & \|\tilde{\mathbf{Z}}\tilde{\mathbf{w}} + b - \mathbf{y}\|_2^2 + \epsilon\|\mathbf{D_1}\tilde{\mathbf{w}}\|_2^2 + \lambda\|\mathbf{D_1}\tilde{\mathbf{w}}\|_1 \\ \min_{\tilde{\mathbf{w}},b} & \|\tilde{\mathbf{Z}}\tilde{\mathbf{w}} + b - \mathbf{y}\|_2^2 + \epsilon\|\mathbf{D_2}\tilde{\mathbf{w}}\|_2^2 + \lambda\|\mathbf{D_2}\tilde{\mathbf{w}}\|_1 \end{cases} \tag{4.14}$$

The difference matrix has been split between the horizontal differences in 2D interactions, including also main effects, in $\mathbf{D_1}$ and the vertical differences in $\mathbf{D_2}$. A slightly different notation for elastic net with the ℓ_1 constraint included as additional parameter weighted by hyperparameter λ has been used. Dykstra's method allows to solve this system iteratively by adapting the labels $\mathbf{y}$ as a way to inject information about the obtained previous result from one subproblem into the other subproblem [43].

This is potentially interesting, since each subproblem can be solved via SVEN. However, despite some successful experiments a problem arises due to the different elastic net formulations. λ expresses the weight of the penalty with respect to the solution, but t in the constraint notation in Equation 4.6 is an explicit limit on the value of the penalty. They are functionally equivalent since for each value of t a matching value of λ can be found. Still, this relation is not fixed and can only be found post-hoc. Both Dykstra and SVEN build on their notation to solve the problem and are incompatible with the other notation. For example, using the constraint notation with Dykstra would imply that it should be determined beforehand what part of the penalty should be on horizontal and what on vertical differences since it gives an absolute bound. Splitting the value of t in half for each works somewhat, but is ad hoc and converges slowly. Due to this incompatibility, the decomposition path was abandoned.

4.1.3.3 Interactions via constraints

Another attempt to solve 2D fusion has been investigated in literature for the LASSO, but the approach can be carried over to the elastic net. As discussed above with the data transformation the difference notation of the generalized elastic net can be circumvented for main effects. However, in the general case including interactions, it has been shown that additional constraints are needed to do this [177]. Indeed, in the case of the interaction discussed in Figure 4.1 any path could be used for interval weight encoding. In addition, the constraint that all paths should yield the same result should be added. To avoid listing all paths this can be formulated on the level of all (main) diagonally connected intervals. For any pair, the sum of differences down-right should equal the sum of differences right-down.

Let $\hat{\mathbf{Z}}^*$ be the binary data matrix identical to $\hat{\mathbf{Z}}$, but with support for interactions via the path encoding given in gray in Figure 4.1. Let $\mathbf{Q}$ be the matrix with the coefficients for path consistency constraints. Then, `enICS` supporting interactions could be formulated as:

$$\begin{aligned} \min_{\hat{\mathbf{w}},b} \quad & \|\hat{\mathbf{Z}}^*\hat{\mathbf{w}} + b - \mathbf{y}\|_2^2 + \epsilon\|\hat{\mathbf{w}}\|_2^2 \\ \text{s.t.} \quad & \begin{cases} \|\hat{\mathbf{w}}\|_1 \leq t \\ \mathbf{Q}\hat{\mathbf{w}} = \mathbf{0} \end{cases} \end{aligned} \tag{4.15}$$

As a result, methods for solving constrained elastic net could be applied [57, 75]. However, since these constraints cannot be incorporated in SVEN, its speed gain is lost. Due to the associated execution cost of straightforward implementations (e.g. quadratic) it is not beneficial to prefer this approach over `lpICS` building on highly efficient sparse LP solvers. It should be investigated further whether specific implementations for constrained elastic net could lead to future improvements.

4.1.3.4 Iterative reweighting for `enICS`

The reweighting procedure for `enICS` is similar to `lpICS` but should be slightly adapted to account for the new data formulation. Based on the current solution $\hat{\mathbf{w}}$ and the weighting parameter a supplied by the user, reweighting can be done as expressed in Algorithm 4.1. A key difference with the `lpICS` equivalent Algorithm 3.1 lies in the calculation and application of the reweighting values $\hat{\boldsymbol{\chi}}_{i,i}$ since they are applied on the data $\hat{\mathbf{Z}}$, rather than in the (`lpICS`) regularization on $\mathbf{D}\tilde{\mathbf{w}}$. Hence, they are the reciprocal of the `lpICS` reweighting values $\tilde{\boldsymbol{\chi}}_{i,i}$ (neglecting ϵ_1). The value of ϵ_2 remains at 1E-8. Due to different convergence

properties, the maximum number of iterations is higher for `enICS` compared to `lpICS`.

Algorithm 4.1 Iterative reweighting in `enICS`

1: Given: $\hat{\mathbf{w}}$, a
2: **repeat**
3: $\quad \hat{\mathbf{w}}^{\text{old}} = \hat{\mathbf{w}}$
4: $\quad \hat{\boldsymbol{\chi}}_{i,i} = a|\hat{w}_i|$
5: $\quad \hat{\mathbf{w}} \leftarrow$ solve Equation 4.7 with $\hat{\mathbf{Z}}$ replaced by $\hat{\mathbf{Z}}\hat{\boldsymbol{\chi}}$:

$$\min_{\hat{\mathbf{w}},b} \|\hat{\mathbf{Z}}\hat{\boldsymbol{\chi}}\hat{\mathbf{w}} + b - \mathbf{y}\|_2^2 + \epsilon\|\hat{\mathbf{w}}\|_2^2 \quad \text{s.t.} \quad \|\hat{\mathbf{w}}\|_1 \leq t \tag{4.16}$$

6: **until** AVG$(|\hat{\mathbf{w}} - \hat{\mathbf{w}}^{\text{old}}|) < \epsilon_2$ or #iter == 25

4.2 Experiments and applications

A first experiment indicates the gain in execution time by replacing `lpICS` by `enICS` using a synthetic dataset. Then, a detailed study is made on the impact of using both algorithms with and without preselection on an existing biomedical dataset. A final series of experiments applies all ICS approaches on the biomedical datasets introduced in the previous Chapter, comparing them statistically to standard machine learning algorithms.

4.2.1 Execution time

Similarly to the previous Chapter, a synthetic dataset is introduced since its outcome is known and it can easily be expanded to increase the problem size. A binary classification problem was created without interactions as `enICS` does not support them. The labels are generated according to the following formula:

$$y = (x^1 + (x^2)^2 + 2\text{sign}(x^3)) > 0 \tag{4.17}$$

It involves a linear effect, a quadratic effect and a Heaviside step effect. As before, superscripts are used to denote variables. The sign function yields -1 for negative and 1 for positive numbers. Additionally, $\text{sign}(0) = 0$. x^1, x^2 and x^3 are sampled from a standard normal distribution $\mathcal{N}(0,1)$. Additional non-informative variables, also drawn from $\mathcal{N}(0,1)$ are added during the experiment.

The execution time is compared via the execution of the `lpICS` and `enICS` core as formulated in Equation 3.4 and 4.7, respectively. To this end, 2000

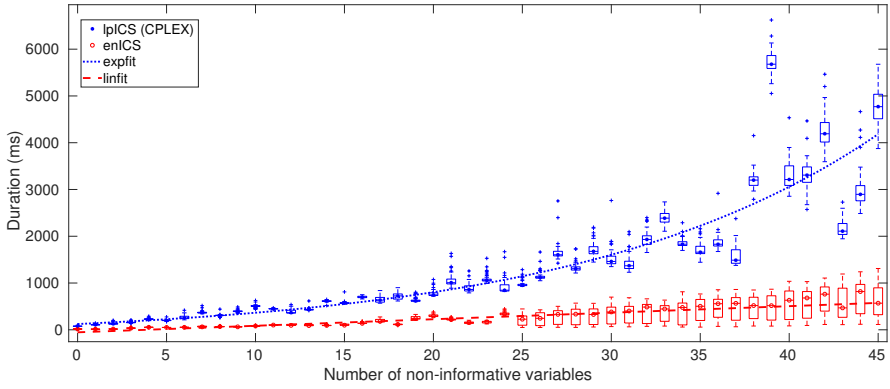


Figure 4.2: Execution time of the core algorithms of `lpICS` and `enICS` for an increasing number of non-informative variables. For every number of non-informative variables, 50 repetitions of both algorithms are reported as boxplots. The dotted line shows the exponential execution time increase for `lpICS`, whereas the dashed line shows a linear increase for `enICS`.

data points, 1000 from each class, are generated. The problem is solved for an increasing number of non-informative variables ranging from 0 to 45. For each added variable the distribution of 50 executions is reported as a boxplot.

Figure 4.2 shows the impact of using `enICS` instead of `lpICS`. The latter's execution time increases exponentially, as indicated by the dashed exponential fit. It starts at a fraction of a second if only the three informative variables are used and grows to several seconds if more than 25 non-informative variables are added. In contrast, `enICS` grows slowly, linearly, and remains under 500ms. This shows its ability to deal with high-dimensional data. In principle, `enICS` execution time should be constant. However, the growing data matrices used while solving the problem yield a linear increase instead because of the underlying matrix computations. It should also be mentioned that the CPLEX solver has been used to solve `lpICS`, which is significantly faster than the standard Matlab implementation.

4.2.2 Model comparison

An in-detail model comparison is performed based on the 12-lead ECG arrhythmia dataset of the UCI Machine Learning database [67]. It contains records of 452 patients classified as either normal or one of several classes of arrhythmia. Here, the problem is considered as binary: normal vs any of the

arrhythmia classes. 80% of the data is selected by stratified sampling for training. The remainder is used for testing. Each patient record contains 279 features, 206 continuous and 73 binary. 5 of them are removed due to missing values. Feature details can be found in [67]. This high-dimensional dataset is used to assess the impact of the proposed measures on the complete ICS workflow: both the choice of type (`enICS`/`lpICS`) and the presence of preselection will be evaluated. Automatic tuning will be used during iterative reweighting since the semi-automatic mode would disrupt time measurement. The number of bins per variable was set to 7.

The test results are shown in Table 4.1. Notice the long computation time for `lpICS`. It can be improved drastically by either switching to `enICS` or by using the preselection, which reduces the time by a factor of 53.7 and 19.1, respectively. The preselection reduced the number of variables from 274 to 28. It can be run for `enICS`, but this appears counterproductive. The execution time increases, because the preselection approach is slower than direct `enICS`. This might be due to the specific LSSVM package used for preselection. The AUC of `enICS` can be improved by preselection, but its accuracy reduces.

Overall, `lpICS` tends to be geared towards simpler methods than `enICS` when used in automatic mode as can be deduced from the number of intervals. Also, the number of scoring points attributed by `enICS` tends to span a wider range. This leads to a more complicated risk profile since all possible scores have to be listed. This can be an advantage if one looks for a more detailed risk assessment, as long as it remains accurate. Alternatively, a detailed risk profile could also be obtained by increasing the maximum number of intervals per variable, if enough data is available.

Another observation is that the models change when the problem changes either in formulation or in the variables involved. For example, after preselection, `lpICS` finds a different model than before although its complexity is identical and its performance similar. This can partly be explained by the fact that the problem solution is not unique to begin with. However, another contributing

Table 4.1: Test results of the approaches on the arrhythmia database.

	No preselection		**Preselection**	
	lpICS	**enICS**	**lpICS**	**enICS**
Duration (s)	852.14	15.88	44.71	24.73
#variables	4	4	4	3
#intervals	8	11	8	11
AUC	0.816	0.772	0.803	0.799
Acc (%)	82.2	77.8	81.1	74.4

Table 4.2: Selected effects of ICS approaches on the arrhythmia database. The ECG lead is mentioned between brackets.

No preselection		Preselection	
lpICS	**enICS**	**lpICS**	**enICS**
R' width (V1)	R' width (V1)	R' width (V1)	R' width (V1)
T ampl (V6)	T ampl (V6)	T ampl (V6)	T ampl (V6)
S ampl (V3)	Q ampl (V3)	S width (V3)	QRS duration
R' width (V2)	Q width (V2)	Age	

factor is the use of the automatic mode. In this case, a cutoff factor (default 0.95) defines the sacrifice with respect to the maximal AUC to obtain a simpler model. Alternatively, the user can intervene and steer the algorithm for both preselection and iterative reweighting to the desired trade-off (semi-automatic mode) using the interface described in Appendix A.

The change in the model can also be understood when looking more in detail at the selected variables for all investigated methods in Table 4.2. The features are defined in [67]. Apart from a patient's age, they all refer to elements of the ECG waveforms: QRS duration, the widths and amplitudes of Q, R', S and T waves. R' indicates a second R peak, sometimes occurring in case of right bundle branch blocks. The indication between brackets refers to the ECG lead. It can easily be seen that two of the effects have been selected in all cases. The performance is increased by adding some other variables. However, inspection shows that these are removed in the preselection phase. The same variance can probably be explained by other minor variables as well, hence their selection. A larger number of samples might alleviate this issue, although correlation could be involved as well.

4.2.3 Comparison with standard techniques

This experiment extends the validation on five biomedical datasets introduced in the previous Chapter, yielding six binary classification problems. ICS will be run with its standard settings (automatic tuning) in four setups: `lpICS` with and without preselection and `enICS` with and without preselection. The validation also includes a performance comparison with other classification techniques. Naive Bayes (NB) and linear and nonlinear (RBF kernel) Least-Squares Support Vector Machines [158] have been chosen as they are widely accepted techniques. CART Decision Trees (DT) [32] have been included as an alternative interpretable classifier. Stratified random sampling selects two thirds of the data for training and one third for testing. In case of binary

Table 4.3: Comparison of test performance (AUC) and complexity (I: number of intervals, V: number of variables) for all ICS setups (`lpICS` and `enICS`, each with and without preselection), Decision Tree (DT), Naive Bayes (NB), linear and nonlinear (RBF) SVM. Datasets include Acute inflammation ('Inflammation' and 'Nephritis' labels), Breast Cancer Diagnosis, Cardiotocography ('Cardio'), Chronic kidney disease ('Kidney'), and Indian Liver Patient ('Liver') data.

	Inflammation		**Nephritis**		**Breast Cancer**		**Cardio**		**Kidney**		**Liver**	
	AUC	**I (V)**	**AUC**	**I (V)**	**AUC**	**I (V)**	**AUC**	**I (V)**	**AUC**	**I (V)**	**AUC**	**I (V)**
lpICS	1	6 (3)	1	8 (4)	0.933	5 (2)	0.945	11 (5)	0.932	4 (2)	0.677	60 (9)
lpICS-pre	1	6 (3)	1	4 (2)	0.933	5 (2)	0.927	96 (8)	0.959	8 (3)	0.688	20 (5)
enICS	0.962	4 (2)	1	4 (2)	0.942	4 (2)	0.959	14 (6)	0.938	6 (3)	0.685	6 (2)
enICS-pre	0.955	4 (2)	1	4 (2)	0.942	4 (2)	0.873	6 (3)	0.941	8 (2)	0.685	6 (2)
DT	1	/ (4)	1	/ (2)	0.936	/ (3)	0.935	/ (13)	0.952	/ (4)	0.659	/ (3)
NB	1	NA	1	NA	0.976	NA	0.938	NA	0.964	NA	0.720	NA
SVM-lin	1	NA	1	NA	0.993	NA	0.957	NA	0.948	NA	0.696	NA
SVM-rbf	1	NA	1	NA	0.994	NA	0.957	NA	0.985	NA	0.706	NA

data, Naive Bayes builds the model using multivariate multinomial distributions instead of the default normal distributions. The SVMs are tuned via simulated annealing, whilst the Decision Trees are pruned to obtain the tree with the smallest cross-validation classification loss on the training data.

Table 4.3 provides an overview of the performance and model complexity of all ICS methods and established classifiers. Performance is judged by the AUC. The model complexity of the ICS methods is given by the number of variables (V) and intervals (I) in the final model. Since this measure is not applicable for non-ICS methods their value has been replaced by NA. However, one can argue they use all variables in the datasets.

One can observe that the classical machine learning methods seem to outperform ICS in most cases. A statistical analysis was carried out to judge the significance of this perception. Firstly, all algorithms were compared jointly over all datasets using a non-parametric Friedman test on the AUC values, which proved significant ($p = 0.022$). Subsequently, a multiple comparison between all algorithms was carried out to identify which algorithms performed significantly different. The Bergmann-Hommel procedure was applied to the overall significance level of 0.05 as posthoc correction for the influence of multiple comparison [166]. Here, no significant differences were found, indicating we cannot reject that all algorithms perform equally well. The average ranks of all algorithms as determined in the Friedman test are displayed in Table 4.4. Furthermore, Figure 4.3 shows a graphical representation of the multiple comparison test [49]. The dashed lines correspond to the average ranks of the corresponding algorithms on a scale of all possible ranks. Note that the ranks are inversely proportional to the AUC values: lower ranks indicate higher AUC values since the first-ranking algorithm is the best. The bold line groups all algorithms for which the null hypothesis of equal rank distribution means cannot be rejected on a pair-by-pair basis. In this case, all algorithms are in a single group since none of the pairwise tests was significant. Hence, one cannot claim that traditional Machine Learning algorithms outperform any of the ICS approaches in all cases.

Apart from the model complexity and its performance in terms of AUC, one could also be interested in its risk assessment. It can be judged by the slope and bias of the calibration curve, a (weighted) linear fit between the estimated and empirical risk. For ICS, estimated risks are available. In the NB and DT models, posterior probabilities fulfil that role. In case of SVMs, one can build the risk assessment from the latent variables using logistic regression. A comparison of the risk prediction capacity of all algorithms is shown in Table 4.5. The lesser results for the Liver dataset were already apparent in Table 4.3, but are even more pronounced here for all classifiers excepts the SVMs.

In order to perform a statistical evaluation of the classifiers' risk prediction

Table 4.4: Average ranks of the classifier AUC values over all datasets. Lower values are an indication of possible better performance.

lpICS	**lpICS-pre**	**enICS**	**enICS-pre**	**DT**	**NB**	**SVM-lin**	**SVM-rbf**
5.75	4.92	4.92	6.08	5.33	3.17	3.42	2.42

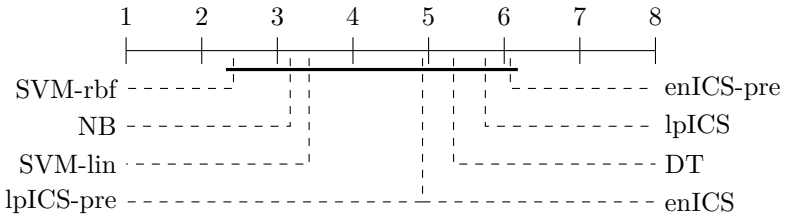


Figure 4.3: Representation of the multiple comparison of the AUC values for all algorithms using Bergmann-Hommel posthoc correction after a non-parametric Friedman test. Dashed lines show the average rank of the corresponding algorithms over all datasets. The bold line shows where the null hypothesis of equal distribution means could not be rejected: no significant differences were found.

capabilities, the slope and bias can be combined into a single measure of goodness of fit with respect to the ideal curve. Consider the calibration curve $t = ap + b$ with slope a and bias b which models the relation between predicted risks p and empirical ('true') risks t. The ideal model would be $t = 1p + 0$ with slope 1 and bias 0. Hence, one can define a calibration error measure e as the ℓ_p norm of the difference between the actual and the ideal curve [36]. For the simplest case (ℓ_1), this yields:

$$e = \int_0^1 |(a-1)p + b| dp$$

Using the error values calculated for all (slope, bias) pairs in Table 4.5 an identical statistical analysis as before can be carried out. Table 4.6 and Figure 4.4 show the results. The Friedman test was significant ($p = 0.021$) and the multiple comparison test indicated a single significant difference. We can reject the hypothesis that Naive Bayes' calibration is equally good as SVM-lin's. As a result, Figure 4.4 has two groups within which equal calibration performance cannot be rejected.

Table 4.5: Comparison of the quality of risk estimation on the test data via the slope and bias of the calibration curve for all ICS setups (`lpICS` and `enICS`, each with preselection), Decision Tree (DT), Naive Bayes (NB), linear and nonlinear (RBF) SVM. Datasets include Acute inflammation ('Inflammation' and 'Nephritis' labels), Breast Cancer Diagnosis, Cardiotocography ('Cardio'), Chronic kidney disease ('Kidney') and Indian Liver Patient data ('Liver').

	Inflammation		**Nephritis**		**Breast Cancer**		**Cardio**		**Kidney**		**Liver**	
	slope	**bias**	**slope**	**bias**	**slope**	**bias**	**slope**	**bias**	**slope**	**bias**	**slope**	**bias**
lpICS	1	0	1	0	0.89	0.03	0.95	-0.01	0.95	0.05	0.44	0.40
lpICS-pre	1	0	1.02	-0.02	0.89	0.03	0.92	0.01	0.98	0.02	0.55	0.32
enICS	1.01	-0.01	1.02	-0.02	0.96	-0.01	1.04	-0.01	0.95	0.05	0.72	0.21
enICS-pre	1	0.01	1.02	-0.02	0.96	-0.01	1.04	0	0.92	0.05	0.72	0.21
DT	1	0	1	0	0.90	0.01	0.94	0.03	0.91	0.09	0.63	0.27
NB	1.15	-0.05	1.03	0.05	0.84	0.05	0.74	0.06	0.84	0.18	0.34	0.61
SVM-lin	1	0	1	0	0.96	0.01	1	0	0.99	0.03	0.72	0.22
SVM-rbf	1	0	1	0	0.94	0	0.88	0.03	0.97	0.03	0.68	0.24

Table 4.6: Average ranks of the classifier calibration errors over all datasets. Lower values are an indication of possible better performance.

lpICS	lpICS-pre	enICS	enICS-pre	DT	NB	SVM-lin	SVM-rbf
4.83	4.83	4.00	4.00	4.17	8.00	2.83	3.83

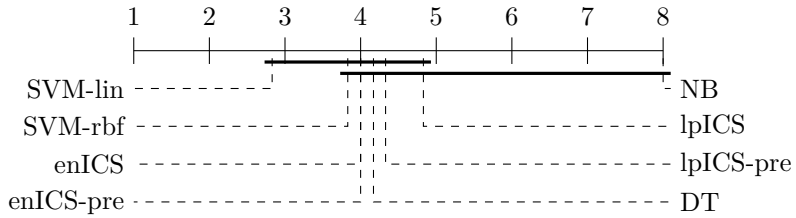


Figure 4.4: Representation of the multiple comparison of the calibration error for all algorithms using Bergmann-Hommel posthoc correction after a non-parametric Friedman test. Dashed lines show the average rank of the corresponding algorithms over all datasets. The bold line shows where the null hypothesis of equal distribution means could not be rejected. Two groups are identified: one excluding SVM-lin and one excluding Naive Bayes since these two had significantly different ranks.

4.3 Discussion

This Chapter addressed two problems. On the one hand, the mathematical uniqueness of the `lpICS` model could not be guaranteed, although in practice, it has proven quite robust. On the other hand, the binary expansion of the feature space lead to very large LP problems which could take a long time to solve.

The quadratic approach proved unable to improve both uniqueness and execution speed. Yet, the alternative `enICS` brings improvements for both aspects. Without interactions, `enICS` was shown to be a huge improvement over `lpICS` in terms of computational cost. Its solution as an SVM leads to the interesting option to solve either the primal or dual problem depending on the dimensionality of the feature space and the number of observations. Recall that the elastic net corresponds to the dual of the SVM. For smaller data sets, the corresponding SVM primal problem is more easily solved with fast iterative methods [39] because it will always have a number of variables equal to the number of original observations. For very large datasets, the dual SVM itself can be solved by construction of a kernel matrix. Both the primal and the dual could make use of the fact that at any given time only a part of the data is needed. This can

be exploited as a solution for the reduced sparsity of $\hat{\mathbf{Z}}$ compared to $\tilde{\mathbf{Z}}$. In fact, precomputing all kernel elements for the dual problem is often wasteful [31]. Finally, for large datasets, not all data is necessary to have a good estimation of its impact on the feature space. A well-selected subset might yield the same results as the full set with reduced complexity. Fixed-size methods have been developed to find such subsets [158].

In this Chapter, elastic net has been used for its sparsity characteristics, both in the preselection and in `enICS`. It is related to LASSO and often used for feature selection. Their convexity differences lead to differences in which variables are going to be selected. In particular, elastic net is characterized by a grouping effect [186]. This means that if variables are strongly correlated, elastic net will give them equal coefficients, effectively grouping them. In contrast, the LASSO would attribute the combined coefficients randomly to one of the variables. Multicollinearity, that is, correlated variables, can be an issue because they carry related information and have an impact on the stability of the model. However, in some cases, the grouping effect is in fact desirable. A typical example is the detection of correlated gene expressions in bioinformatics. In such cases, the elastic net detects all important effects instead of picking a single one. Still, a more sparse solution can be obtained by increasing the ℓ_1 penalty hyperparameter, thus obtaining decorrelation. More generally, the elastic net balances ridge regression with LASSO. By adapting its hyperparameters it can be more tuned towards one or the other to yield the desired behaviour.

The experiments clearly show the merit of `enICS`. Its execution speed is superior to `lpICS` and it leads to similar models. Yet, they are not exactly the same, since the underlying problem is slightly altered. The same applies for preselection. In general, the same most important effects can be discovered, but additional model details can differ.

When looking at a comparison on several biomedical datasets, ICS could not be proven significantly inferior to standard techniques, neither on the level of AUC performance nor for calibration. In contrast, one can observe the simplicity of most ICS models which leads to its interpretability. The total number of intervals for all ICS models is usually below 10, resulting from two to four variables. One can also conclude that ICS outperforms the interpretable decision tree in terms of number of selected variables in almost all cases.

However, care has to be taken for the generalization of these results. The datasets are still relatively small which leads to a large confidence interval on the AUC values and calibration performance. Moreover, it should also be taken into account that only six datasets were used for the comparison. As a result, one cannot generally deduce that ICS has equivalent power compared to the other approaches. Nevertheless, this conclusion could not be discarded either. More and larger datasets would allow to make more precise and therefore more

general claims about ICS's performance difference with other algorithms.

The validation datasets are representative for biomedical datasets in the fact that they do not (seem to) contain important interactions between variables. ICS only detected interactions with `lpICS-pre` for the cardiotocography set and this was far from the optimal solution, as can be derived from its comparison with e.g. `lpICS`. In all other datasets all interactions have been discarded in favour of simpler main effects. Judging from its high complexity of 60 intervals one could also suspect interactions in the Liver `lpICS` model. However, this is an example of overfitting for which nine out of ten main effects have been selected. The estimated training AUC is 0.86, which is much higher than the test AUC of 0.677 mentioned in Table 4.3.

The acute inflammation dataset ('Inflammation' and 'Nephritis') was modelled mostly perfectly. This was expected, since they have been shown to be perfectly classified by the rule-based expert system they were recorded for. Hence, nearly all methods manage to find a perfect model, except for the `enICS` approaches with Inflammation due to oversimplification. This case also illustrates that `lpICS` and `enICS` do not necessarily yield the same model due to the different problem formulation based on resp. classification and regression.

No significant performance differences between ICS and standard machine learning could be proven, yet the observed insignificant differences can be explained by a number of factors. Most importantly, the model is different, including its data transformation. ICS uses a piecewise linear data model but does not pose any restrictions on the weight distribution. Hence, it can model e.g. linear, quadratic or even sinusoidal effects, within the limits of its granularity. Furthermore, it has the advantage of taking the variable type into account. In a way, it transforms all variables to a categorical representation. Decision Trees follow a similar approach and are therefore also interpretable. Their drawback is the larger number of variables selected here. Moreover, the flat structure of ICS is easier to interpret than the nested structure of the trees: how can one easily determine or quantify the impact of a single variable?
In contrast to ICS and DT, Naive Bayes attempts to model the underlying variable distributions to make statistical inferences. In practice, its default assumption of gaussianity will probably not always be satisfied. Table 4.3 shows this does not necessarily result in lower performance. Moreover, it can deal with binary and categorical data via an adjustment of the distribution assumption. The linear SVM works surprisingly well given its limitations. It does not take variable types into account at all and considers only a simple linear combination of the variables. Yet, this model appears to be largely sufficient for most, if not all, of the problems considered here. Finally, the nonlinear RBF SVM is known to be a universal approximator, that is, it can describe any model. However, whether it detects the correct one is a matter of tuning of the available data. Due to its flexible model, it is more prone to overfitting than its linear

counterpart.

The performance results also demonstrate a shortcoming of ICS. In its automatic mode, it does not always select the trade-off one would aim for in manual training. Indeed, the automatic trade-off is performed in a rather naive way. This phenomenon presents itself at two different levels. Firstly, preselection boosts execution speed in large problems, but it does not always lead to the best model. This is particularly true for the cardiotocography problem, where `lpICS` yields a simpler model than `lpICS-pre` and yet obtains higher performance because some informative variables have nevertheless been discarded in the preselection. This is possible because preselection, although informed, still happens at the level of the complete effects (main effects or interactions). It is not as optimized as the subsequent workflow to take the interval structure into account. This drawback also exists in semi-automatic mode, but a user could base a decision on stability of the curve as well, which automatic mode does not take into account. Still, a good model is obtained, but not the simplest one. Secondly, one can observe a simpler, yet accurate model can be available. In case of the breast cancer problem, `enICS` yields a slightly simpler, yet better model. Also, in the liver dataset, one would rather be interested in the `enICS` model. However, keep in mind that the starting conditions are different for `lpICS` and `enICS`, both due to the problem formulation and due to the fact that the former considers interactions, whereas the latter does not. If the impact of the tuning parameters would be presented to the user, he or she would be able to guide the model towards the simpler, more stable and more powerful models.

Furthermore, despite the lack of significant differences for the calibration, one can make some observations on part of the datasets. Although Naive Bayes explicitly works with probability distributions it tends to be similar or weaker than the ICS models. This might be caused by a large deviation of the actual data distribution compared to the assumed gaussianity. The SVMs provide a good prediction of the risks, particularly if the linear model is used.

Summarizing the comparison experiment, one can conclude that performance loss with respect to established techniques is not significant, which is particularly worthwhile considering the gain in interpretability.

The ICS algorithms have been implemented in a Matlab toolbox, which is freely available for research purposes. It is discussed in detail in Appendix A. It offers easy access to the formulation of an ICS problem and allows the user to intervene via a graphical interface. Visualization and assessment are also incorporated. Finally, an existing model can easily be applied to new data.

4.4 Conclusion

In this Chapter, Interval Coded Scoring has been extended for larger problems. Two different problems have been addressed. On the one hand, the solution offered by `lpICS` is not unique. On the other hand, large problems could lead to a heavy computational burden due to the data transformation. A first solution for the computational cost is the introduction of preselection. The size of the problem is reduced by means of an interval-informed wrapper method based on a new data transformation and elastic net selection. At this point, a user can intervene or an automatic threshold can be used to select the remaining features.

Secondly, the quadratic ICS formulation addressed the uniqueness problem by exploiting theoretical findings applied on the Linear Programming formulation. The resulting quadratic problem had the same size and sparsity as the original LP problem. The optimized LP solver outperformed the solver for the quadratic problem (QP). Yet, more efficient solution methods could yield a benefit for the QP approach.

Although a solution for uniqueness, the QP approach does not scale well for larger problems. To address this restriction as well, `enICS`, based on elastic net was introduced. A new data transformation encodes the data in terms of interval differences instead of interval values, which also required small adaptations to the iterative reweighting. The `enICS` core can be solved efficiently and independent of the size of the feature space due to its equivalence with a Support Vector Machine. For very large datasets future expansions could include support for fixed-size approximations. As a drawback, the data transformation does not allow variable interaction effects. A decomposition and constraints approach were investigated but did not yield a definite solution. Hence, the ICS toolbox described in Appendix A is limited to `enICS` with main effects.

Nevertheless, the experiment Section shows the advantage of the improvements described in this Chapter. A repeated timing of the core algorithms demonstrates that `enICS` outperforms the optimized CPLEX solver for `lpICS`. A direct comparison with and without preselection further indicates the advantage that can be gained over pure `lpICS`. However, the comparison also points to differences in outcome due to different formulations or the use of preselection. The main important effects are detected, but additional model details are sometimes provided via different variables. As the discussion indicates, this might also be due to correlation, which is not dealt with explicitly in ICS.

A full comparison of ICS with standard approaches such as Decision Trees, Naive Bayes, linear and RBF Support Vector Machines has been performed. Although performance differences appear they are not significant for the datasets under study. This hold true both for the classification outcome and for the calibration

error. This shows that in similar settings where interpretability is a key issue, ICS would be the method of choice.

Part II

Activity recognition and assessment from wearable sensors

Sometimes the most important thing in a whole day is the rest we take between two deep breaths, or the turning inwards for five short minutes.

E. Hillesum, *An Interrupted Life*

Chapter 5

Activity monitoring

This Chapter is an overview of the current state-of-the-art and serves as introduction to the next three Chapters in Part II. It discusses activity monitoring, linked to the objectives of the SPARKLE project. Therefore, it first introduces axial spondyloarthritis as a representative and known example of a chronic musculoskeletal condition (cMSC). For this disease, it stresses the importance of functional capacity and indicates that it should be addressed objectively, requiring automatic activity recognition and assessment. Hence, the Chapter elaborates on current approaches for activity recognition with a primary focus on wearable sensors because of their applicability in a home environment. Then, activity assessment is addressed, followed by a description of discriminant analysis for classification and relevant performance metrics. Finally, the last Section describes the datasets which have been recorded as part of the studies.

5.1 Chronic musculoskeletal conditions

The decision support systems discussed in Part I can be applied in many fields for various applications. In this work, the specific focus is on patients suffering from chronic musculoskeletal conditions (cMSCs) in general and (axial) spondyloarthritis (axSpA) in particular.

5.1.1 Cause and symptoms of axial spondyloarthritis

Spondyloarthritis (SpA) is a chronic musculoskeletal disease that is typically characterized by inflammatory back pain (axial variant, axSpA), though a

peripheral variant exists as well. Other symptoms include inflammations of the sites where tendons and ligaments attach (enthesitis), fingers (dactylitis) or the eye (uveitis), psoriasis, inflammatory bowel disease and alternating buttock pain. Some other recognized criteria pointing towards axSpA are a limited motion range of the lumbar spine and restricted chest expansion [162]. These symptoms can be grouped in fluctuating and degenerative developments. For example, inflammations can be treated and suppressed, but they can reappear in a later stage. On the other hand, ossification of the spine, restricting the patient's motion range, is degenerative.

The exact cause of axSpA is unknown. Research has shown a strong connection to the presence of the HLA-B27 antigen, suggesting a genetic link. Yet, the exact process of the pathogenesis is unclear since many people carrying the antigen are never even affected [161]. Because a single criterion does not suffice for diagnosis the disease is often detected quite late based on the diverse body of symptoms that together rule out other possibilities.

An early approach to diagnosis was based on radiography by checking for advanced sacroiliitis. Yet, it is only conclusive for part of the patients suffering from ankylosing spondylitis (AS) where the vertebrae of the spine sometimes partly fuse together, greatly limiting the mobility of the patient. Due to the introduction of Magnetic Resonance Imaging (MRI) inflammations could be detected more accurately and the definition was broadened. Ankylosing spondylitis was brought under the umbrella of spondyloarthritis and all other variants were termed non-radiographic. Still, instead of immediately resorting to MRI, other methods can be applied for diagnosis, based on the number of observed symptoms and the clinician's judgement on the radiograph. Moreover, a blood sample yields additional information such as the presence of HLA-B27 mentioned earlier, but also assesses the level of C-reactive protein (CRP) and the erythrocyte sedimentation rate (ESR), two indicators for inflammation [162]. An overview of an approach based on current practice can be found in Figure 5.1.

5.1.2 Treatment

The main goal of the treatment is reducing disease activity and avoiding joint damage. Therefore, the focus is mainly on anti-inflammatory drugs. The first treatment consists of Non-Steroidal Anti-Inflammatory drugs. In many cases, they allow to control joint symptoms and axial structural damage. For peripheral manifestation, intra-articular steroid injections can be used as well, though their effect is temporary. Systemic steroids are not usually applied due to adverse effects.

The last two decades a biological approach with Tumor Necrosis Factor α (TNF-α) inhibitors has been introduced with good results. They are mostly applied

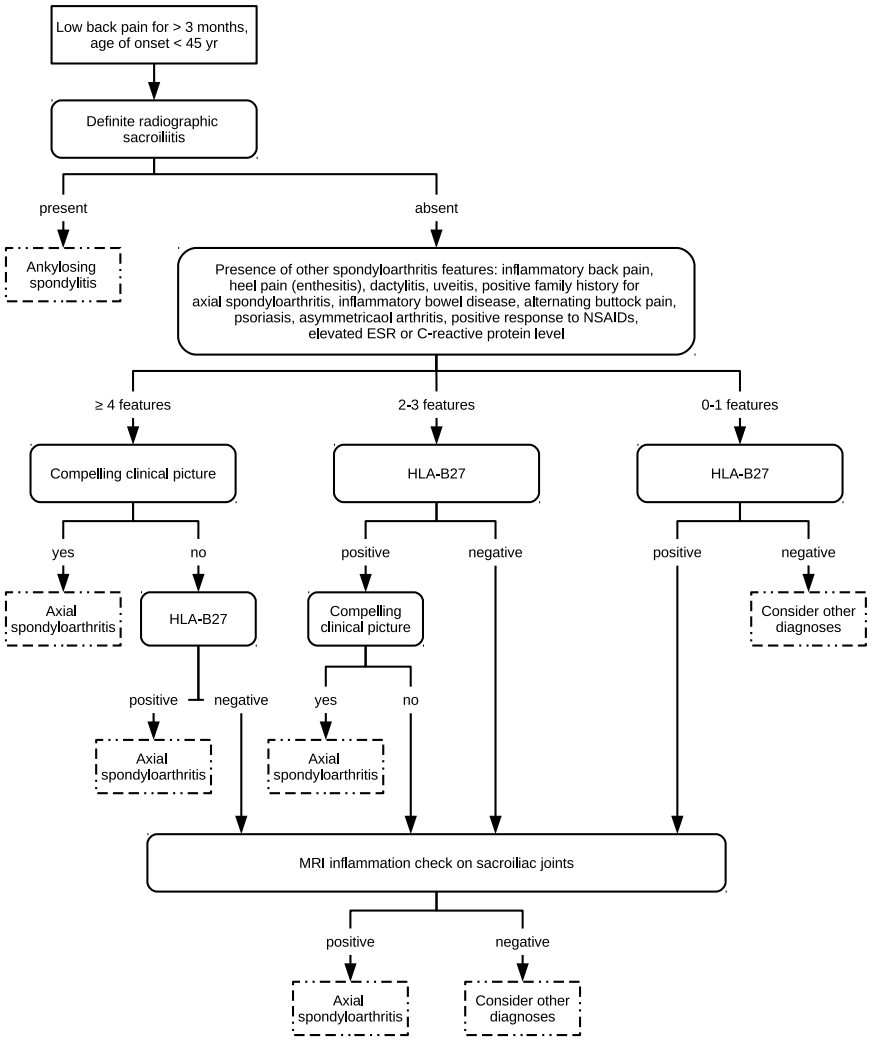


Figure 5.1: Flowchart for the diagnosis of spondyloarthritis, adapted from [162]. Rounded boxes represent tests, dashed boxes the outcome. After a radiographic check-up, other factors are considered such as various inflammations, the response to Non-Steroidal Anti-Inflammatory Drugs (NSAIDs), the blood level of C-reactive protein and the Erythrocyte Sedimentation Rate (ESR). Also, the radiologist's judgement on the clinical picture and the presence of the HLA-B27 antigen are taken into account. The final diagnosis stage involves Magnetic Resonance Imaging (MRI).

as a second line of defence since, depending on the specific type, they may lead to increased psoriasis or activation of latent tuberculosis or viral infections [38].

5.1.3 A functional approach

So far, SpA has been approached from the point of view of disease activity. However, another important aspect is patients' functional capacity, that is, their ability to execute a task e.g. whether they are still able to work. Functional capacity and disease activity are expected to correlate to some extent, but additional factors such as pain thresholds and regular exercise influence the results. This is recognised in current practice by prescribing physical therapy in addition to drug therapy. The combination has proven to yield added value [170], though some claim that this effect is only established for the ankylosing spondylitis variant [124]. Functional assessment will be discussed in Section 5.3.

Automatic objective functional assessment can be introduced as an addition or alternative to current, often subjective, practice. In order to implement this, activities should be detected and recognized automatically, after which an adequate assessment model can be applied. The current state of the art is introduced in the next two Sections on Activity recognition and assessment, respectively.

It should be noted that such an objective functional assessment is likely to exhibit fluctuations over time corresponding to the previously mentioned groups of symptoms. A fluctuating trend might be observed, linked to the inflammations, whereas ossification leads to a degenerative assessment, converging to the current stabilized state of the patient.

5.2 Activity recognition

Early approaches to physical activity recognition were often vision-based and are rooted in general object recognition and tracking for industrial and military purpose [60]. The goal is to extract images from video data and label them as (part of) a specific activity. Historically, global representations were used, later followed by descriptions based on local regions of interest combined with models [131, 181]. Over the last decades, these techniques have been refined and continue to be used, mainly because of the increased availability of computational power and large training databases. For example, depth maps were introduced as an additional source of information for increased robustness [5] and are now readily available, e.g. in popular entertainment industry products. Due to all these improvements, vision approaches have been used for many applications.

Vision systems have obtained good results in e.g. visual surveillance, tracking, face recognition, human computer interaction and also contribute to the growing field of autonomous vehicles [1, 29, 116, 131, 182]. Another application, closely tied to patient monitoring, is a *motion laboratory*. A patient is equipped with reflective markers which are tracked in real-time in three dimensions, allowing to extract detailed motion patterns. Although general activity recognition is theoretically possible, the setup is mainly used to perform kinematic studies e.g. on gait patterns [63].

Despite these results, vision approaches still suffer from several drawbacks. Apart from the inherent inter- and intrasubject variations, they are challenging because of the influence of background clutter, lighting conditions, camera positioning and possibly stereoscopic camera matching [181]. This makes them suitable to be used only under restricted conditions. Moreover, the necessary specialized equipment is often expensive and requires careful calibration to operate well. These characteristics indicate it is not an optimal choice for patient monitoring, particularly not in daily life or ambulatory contexts.

Wearable sensors are alternatives that have become increasingly popular and powerful. They are discussed in the next Section. After introducing them, sensor placement and recognition methods will be discussed.

5.2.1 Wearable sensors

Technological improvements combined with an increasing miniaturization have lead to the ubiquity of relatively cheap sensors. This is of special importance in patient or elderly monitoring. Data acquisition can hence more easily be transferred into the home environment, particularly since many people nowadays carry several sensor units incorporated into a smartphone. Moreover, since its computing power and connectivity also improves, the phone can serve as a local computing hub for other sensors in a body sensor network. These trends have been exploited in the last decade to design various monitoring applications [123]. Yet, in what follows, this work focuses on generic 'sensors' as a concept which encompasses smartphones but is not limited to it.

Monitoring in daily life can be implemented by equipping the living space with such sensors. Examples include monitoring the use of electrical appliances combined with force sensors on stairs and bed [157], state-change sensors on doors, windows, drawers [160] and even the use of audio recordings [6]. These approaches offer new possibilities, especially in the field of elderly monitoring. The drawback is their restriction to a single environment. This is circumvented with the use of the wearable instead of fixed location sensors. As an example, Figure 5.2 portrays a setup of a hypothetical single-sensor monitoring system attached to the waist. It contains sensor units capturing multiple modalities.

Accelerometers, measuring acceleration, often along three axes, are the most common. When calibrated correctly, they don't exhibit major low frequency drift, but are still susceptible to high frequency noise. Their useful information can be decomposed into on the one hand, the gravity component and on the other hand the inertial (rotational and translational) accelerations induced by motion [76]. In slow movements, the gravitation can be used to assess the (partial) orientation of the sensor since the inertial acceleration is negligible. However, in cases of fast acceleration/deceleration, e.g. fall detection or sports monitoring, it is not.

Another common sensor unit is a tri-axial gyroscope measuring angular velocity. The current orientation of the sensor can theoretically be obtained by dead reckoning, integrating the velocity over time. Yet, due to low-frequency non-stationary noise, the result tends to drift away from the real orientation.

Still another available unit is the magnetometer. It delivers an estimate of the orientation of the earth magnetic field and can therefore be used for absolute orientation estimation, akin to gravity. However, this measure is greatly influenced by the presence of ferromagnetic materials which is a major hurdle with contemporary buildings and objects.

Therefore, the three modalities listed so far are often combined to obtain a robust estimate of sensor orientation. The basic principle remains dead reckoning of

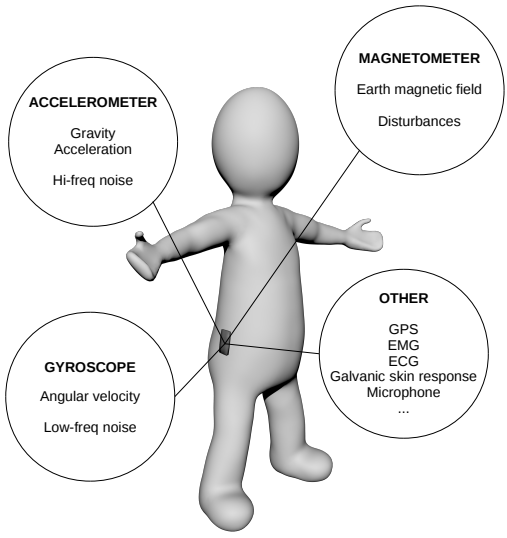


Figure 5.2: Example of a wearable single-sensor setup combining multiple modalities.

the gyroscope, but this is corrected by means of the gravitation and/or magnetic field vector. In case of magnetic disturbances or sudden accelerations, the latter's importance can be tuned down. Several approaches such as complementary filters [69] and various Kalman filters [85] have been designed to solve this *sensor fusion* problem.

Apart from these modalities, other candidates include a GPS receiver, electromyography (EMG), electrocardiography (ECG), galvanic skin response and even microphones for ambient noise [62, 102]. Although adding sensor units yields more potentially valuable information, this has to be weighted against power consumption, a major concern in wearables, particularly when measuring over longer periods of time. High-energy batteries still tend to be bulky and cumbersome [173]. This contrasts with miniaturization as a key concept in wearable technology to achieve unobtrusiveness. As a result, many studies are restricted to the use of one or more accelerometers, still obtaining good results. Our own research mostly follows this approach.

5.2.2 Sensor setup

In addition to the number of sensors, the sensor setup is one of the most important decisions when designing a monitoring system. It is closely tied to the objectives of the study and the activities that should be captured. For example, if only whole-body movement is of importance, a single sensor on the chest or on the back might suffice. However, if reaching or other arm movements have to be detected, the sensor placement should cover the arm.

Various approaches have been used in research and commercially available products. One extreme case is the Xsens MVN system [137]. It consists of 17 sensors covering all body segments. Accelerometry, gyroscope and magnetometer are fused to obtain all segments' orientations which can subsequently be combined into a model for full motion capture. Other commercial devices consist only of a single sensor or a pair of sensors for a dedicated task. For example, McRoberts' MoveMonitor (DynaPort+) has a single triaxial accelerometer and gyroscope on the lower back, used, among others, for detection of walking [127]. Another example, Hocoma's ValedoMotion consists of a pair of IMUs combining accelerometers, gyroscopes and magnetometers for interactive low-back exercises [71].

Several review or comparison studies cover different setups [13, 18, 92, 109]. The sensor positions that were selected most often were the wrists, upper arms, chest, lower back or waist, hip or thigh. The number of sensors was often limited to one or two, although studies with four or more, and even up to 36 sensors have been reported [18]. The preference for a low number of sensors can be explained by several reasons. Firstly, more sensors create additional

technical difficulties. They have to be synchronized in a body sensor network, the data has to be transmitted and combined in a model. Moreover, placing them consistently on the same locations with the same orientation in repeated experiments is a challenge. Secondly, perhaps even more important is the need to have a simple system for long term monitoring. In that way, it can be applied correctly by the subjects themselves and is unobtrusive in their daily activities or e.g. at night during sleep monitoring. This becomes progressively harder when more sensors are used.

It is difficult to directly compare sensor setups since different studies have different objectives and focus on different sets of activities. However, one study compares the impact of accelerometers in different positions on recognition of 15 activities classified according to their intensity and duration (transitory or not) [11]. For low-level activity, e.g. getting dressed, the waist seems preferred, medium-level activities such as walking or wiping the table are associated with chest and wrist. High-level activities like running and cycling are recognized best with a head, arm or knee sensor. Finally, the waist, chest and knee sensors yielded the best results for transitory activities such as sit-to-stand. Often, the best sensor seems connected to the type of movement. For example, high level activities feature a distinct repetitive motion of the legs and head whereas activities such as wiping the table are connected to the wrist. Yet, the study also advises against using the wrist for various activities such as running since its variance is very large and not directly related to the activity. This reinforces the hypothesis that the optimal sensor placement is connected to the nature of the activities under consideration.

5.2.3 Activity recognition methods

Activity recognition and segmentation can mostly be divided in two categories: window-based and pattern-based. Window-based approaches are general: they characterize fixed segments of the signals with statistical features. In contrast, pattern-based approach try to match a predefined activity-specific pattern against the signal to detect the occurrence of and activity and its position in the signal. Both approaches are applicable to any kind of signals e.g. from accelerometers, magnetometers or gyroscopes. This Section first discusses window-based recognition, followed by pattern-based approaches. It focuses on acceleration signals for simplicity and power reasons, as mentioned in Section 5.2.1. It finishes with a short overview of activity segmentation, which can be used as a preprocessing step towards recognition.

5.2.3.1 Window-based recognition

Sliding windows are a common technique in literature. It divides a measured signal in short segments based on a window which advances over it. Such advancement can lead to overlapping or non-overlapping segments. First, a window length and a step size are defined. This also defines the overlap. A window length of 1-2s has proven to give a good trade-off between accuracy and detection speed [17]. Then, from each window, features can be extracted. A classifier can be trained based on these features to attribute an activity label to every window. Many studies follow this approach, but there is a lot of variability in its implementation. Studies vary in, apart from the sensor placement discussed above, the window length and overlap, the features extracted from each window and the choice of classifier [12, 13, 132, 133].

The activities to be classified vary [89]. They often include ambulation (walking, running, stairs) and static poses (sitting, standing, lying). Also, typical fitness exercises have been incorporated [16] as well as more general daily life activities including vacuum cleaning, sweeping or house cleaning, doing the dishes, ironing and watching TV [12, 92, 130].

Many different kinds of features can be extracted. 'Feature' is a general term used for any extracted characteristic value. In this work, the terminology 'statistical feature' is used for general features extracted from windows, independent from the signal. As will be explained later, pattern matching can lead to 'pattern features'. Statistical features can be divided in three groups [12, 13, 89, 132]. *Time features* typically include the mean, median, percentiles, variance, root mean square, kurtosis and zero crossings of a signal, among others. The *frequency features* are based on the Fourier or Cosine transform. Coefficient values are used, but more often the information is condensed into spectral power of a certain band, the principal frequency or spectral entropy. Finally, the (discrete) *wavelet transform features* can also be used. Here, a tradeoff is made between the time and frequency resolution. The transformation allows to simultaneously assess a signal at different scales [104]. The coefficients yielded by this kind of transform can also be used as features, similarly to Fourier features. Of course, when extracting a large set of features, potentially combining all three kinds, not all of them are important for the classification problem at hand. To extract the relevant subset, dimensionality reducing techniques can be applied, as shortly discussed in Section 4.1.1.

Any kind of features can be used to train a classifier. Again, a large number have been used in literature [12, 133]. Decision trees are among the less complex, together with k nearest neighbour (kNN) and threshold-based classifiers. More complex classifiers are Naive Bayes, Random Forests, Support Vector Machines, artificial Neural Networks or Markov models. The best choice depends on the problem, e.g. the feature distribution. However, a simple model is often

favoured because of limited resources, unless the data is processed afterwards and real-time performance is not an issue.

A typical study with a wrist sensor focuses on recognizing six typical activities in six subjects: sitting, standing, walking, running and going up and down the stairs. The extracted time-domain features include the mean, root mean square value, variance, zero crossing rate etc. After comparison of several classifiers a Decision Tree was selected to run locally on the sensor, obtaining a classification accuracy of 80-87%, depending on the sensor position [113]. Another study also recognizes six activities from a single belt-worn accelerometer using a Naive Bayes and kNN classifier with an estimated accuracy of 98% [66].

The studies mentioned above yield good results and are representative for the obtainable performance using the current state of the art. However, they were performed on small groups of healthy subjects. Research has shown that patient populations exhibit a greater variability than healthy subjects [4, 72], further complicating the problem. One specific study monitored 18 healthy subjects and 8 Parkinson patients via a mobile phone. Subject-wise cross-validation showed a difference of over 10% between healthy and patients populations' activity recognition performance [3].

Moreover, the windowing technique is typically applied for repetitive activities such as walking. In the listing of activities mentioned above, the more complex tasks are harder to recognize and therefore less often part of a study. Sometimes, transitory activities are included as well. For example, a hierarchical classifier based on a single accelerometer classifies static, dynamic and transitory activities using an artificial neural network based on statistical features with an accuracy of 97.9% [86]. However, the task was simplified by requiring a static pose of at least 40s before a transition. To overcome this drawback, the methods discussed in the next section could be used.

5.2.3.2 Pattern-based recognition

In this work, transitory activities are defined in contrast to long-term activities. They have a very short duration and are often executed only once in a longer time window. This includes transitions such as sit-to-stand, but also reaching or picking something up. These short activity stretches can be assumed to be characterized by a movement pattern which can be captured by wearable sensors. Therefore, pattern-based recognition aims to explicitly match a movement against a predefined pattern e.g. obtained by averaging training data. From a matched pattern, 'pattern features' can subsequently be extracted. They can be as simple as the correlation between the original signal and the pattern, but more complex features are often desired to account for variability.

This variability can occur due to within- and between-subject differences. A single pattern is an overly simplistic model, unable to capture them. For this reason, pattern-based recognition has often been restricted to a limited number of activities in controlled conditions e.g. to detect sit-to-stand transitions [58]. To solve this limitation, an activity pattern could be broken up in motion primitives, each with their own subpattern. Subsequently, these detections could be combined in a bag of features for recognition of the full activity [180]. In contrast, the most widely used technique, Dynamic Time Warping (DTW), tries to capture the whole movement pattern but allows for deformations [120].

DTW matches signals by a nonlinear transformation (warping). It can be computed using a Dynamic Programming (recursive) approach, as used e.g. for global sequence alignment in bioinformatics (e.g. Needleman-Wunsch algorithm [122]). The principle will be explained for the matching of two patterns $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ (length N) and $\mathbf{y} = [y_1, y_2, \ldots, y_M]$ (length M) as in [99]. The goal is to compute the optimal 'warping path' $W = \{(i_1, j_1), (i_2, j_2), \ldots, (i_K, j_K)\}$ (length K), a sequence of pairs of indices $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, M\}$ in $\mathbf{x}$ and $\mathbf{y}$, respectively. Warping paths can be visualized in a warping grid, as shown in the left part of Figure 5.3. Paths should be continuous, monotonic ($i_k \leq i_{k+1}, j_k \leq j_{k+1}, i_k + j_k < i_{k+1} + j_{k+1}$ $\forall$ k) and the beginning and end of the sequences should match ($(i_1, j_1) = (1, 1), (i_K, j_K) = (N, M)$).

Each path W has an associated cost which can be defined as:

$$C_W = \sum_{k=1}^{K} (x_{i_k} - y_{j_k})^2$$

In DTW, one is interested in finding the lowest cost warping path W_{min} since it corresponds to the best matching of the two sequences. One can use the approach as outlined in the right part of Figure 5.3 using a small subgrid of four points. Imagine the bottom right point p is the end point of the full grid. The cheapest path must necessarily lead through the bottom left, the top left or the top right point. If one knows the cost up to these points, the final step comes down to choosing between the three options indicated by the black arrows. Generalizing, this applies for any point $p_{i,j}$ in the large matching grid in the left part of Figure 5.3. To calculate the cheapest path to it, one simply adds the local matching cost to the minimal cost so far:

$$C(p_{i,j}) = (x_i - y_j)^2 + min \begin{cases} C(p_{i-1,j-1}) & \text{(diagonal)} \\ C(p_{i-1,j}) & \text{(horizontal)} \\ C(p_{i,j-1}) & \text{(vertical)} \end{cases} \tag{5.1}$$

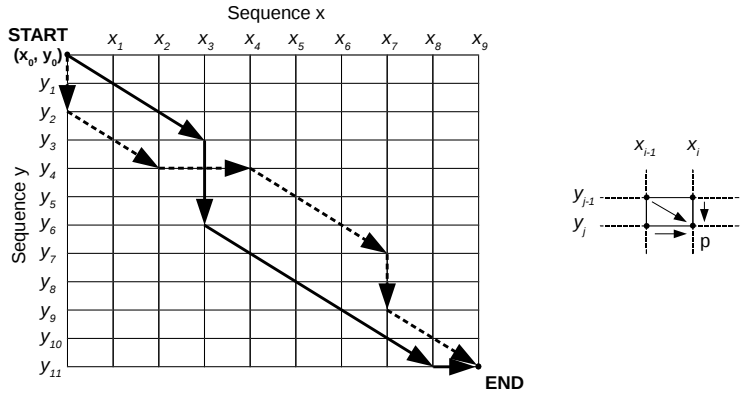


Figure 5.3: Dynamic Time Warping: alignment of two sequences. Two possible warping paths (left) and a detail of the computation of a path (right).

One could calculate this recursively, but it can also be computed in a forward fashion to reduce the computational complexity. This can be summarized as follows:

1. Calculate the costs of the paths to the top row or leftmost column points in the warping grid in Figure 5.3.
2. Calculate the rest of the grid point costs using Equation 5.1, row-by-row or column-by-column.
3. Once the end is reached, the final optimal cost is known. Backtrack through the grid using the costs to find the optimal path.

Note this is a very simple example of DTW that has a time complexity of $\mathcal{O}(MN)$. One can define heuristic boundaries on $|i - j|$ to reduce this. Furthermore, one could use more complex cost functions e.g. normalizing for path length etc.

In short, DTW allows a certain flexibility in signal matching by calculating matching points. Figure 5.4 shows an example result for two damped sine waves of slightly different frequencies. As can be seen, DTW scales shapes along the length of the signal, but because of this it can be prone to errors when large amplitude differences occur. Extension techniques such as derivative dynamic time warping have been introduced to alleviate this issue [84, 121].

In addition to capturing more variance in the data, DTW has also been shown to improve pattern matching robustness with regard to sensor misplacement [83]. Moreover, a study comparing statistical learning classifiers to several template

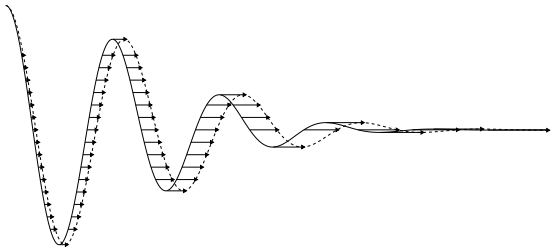


Figure 5.4: An example of Dynamic Time Warping.

matching techniques decides only slightly in favour of the former. Pattern matching accuracy is around 75-80% for 8 main classes, whereas the optimal neural network could attain on average 85% accuracy [110].

5.2.3.3 Segmentation

Both window-based and pattern-based approaches can be applied on the full recorded signals. However, this leads to unnecessary overhead, since the signal is likely to contain activities that are not of interest for the classification problem at hand. The non-informative activities are sometimes called the NULL or rejection class [34]. Moreover, activities are not always well separated from each other e.g. a subject might stand up and start walking immediately. One solution, particularly suitable to sliding window approaches, is merging the stages of detection and recognition. This technique adds a specific rejection class to the data and processes it as any other class [17]. Alternatively, detection can be used as a preprocessing step to classification. For example, it can be dealt with implicitly using Hidden Markov Models (HMM) [90]. It can also be modelled explicitly based on signal energy [65] or using a multi-resolution segmentation based on prior knowledge of the expected length of all activities [179].

5.3 Activity assessment

Assessment can be understood as a broad term. In this work, it encompasses all measures to quantify physical activity. It can be derived as an overall measure based on long-term monitoring or be built based on specific detected activities.

A type of assessment most frequently encountered in wearables' research literature is energy expenditure. Based on a measure of e.g. acceleration from an inertial sensor an estimate is made of the energy used over longer periods [7]. The expenditure can be estimated via regression, independently

of what activity has been performed. Alternatively, it can be calculated via activity recognition. Each activity is regarded as belonging to a group. Then, these groups are linked to a certain expenditure e.g. using metabolic equivalents (METs) [80]. The activity-specific approach is considered superior compared to the regression [30], but it adds to the complexity of activity recognition.

A similar assessment focuses on sedentarity. This is important in elderly populations and patients, but also e.g. at the workplace. A cohort study on 1659 osteoarthritis patients showed a relation between the sedentarity, assessed from an accelerometer, and function loss in terms of gait and sit-to-stand speed [144]. Another study on 193 employees highlighted the impact of work on sedentary behaviour [163].

A more specific activity assessment measure is functional capacity, that is, the 'ability to perform a task' [175]. It can be assessed in several ways. A care provider can visually observe a patient's performance during a specific task and complete a behavioural scale or measure the duration of the movement using a hand-held chronometer. The latter is known as a performance-based test [171]. Typical activities include the timed-up-and-go (TUG) or the six minute walking test. Alternatively and most popular, patient-reported outcome measures, such as the Bath Ankylosing Spondylitis Functional Index (BASFI) questionnaire, can be collected [35]. In a series of ten questions, patients are asked to rate how well they are able to perform regular tasks. The first eight questions target general functional anatomy, whereas the last two refer to typical daily activities. An example of the BASFI questionnaire can be found in Appendix B.

Despite their common use, these approaches have some disadvantages. Firstly, direct observation and performance-based tests are operator dependent, time consuming and limited to the clinical environment [79]. In addition, these techniques require regular consultations at the hospital causing an organizational cost and burden for both the patient and the healthcare system. Furthermore, since a patient-reported outcome reflects the patient's self-judgement or the therapist's expert opinion, the assessment may suffer from over- or under-estimation of their functional capacity. This, in turn, might obscure the patient's true status or progress during treatment [33].

These disadvantages can be alleviated using a more automated approach as an alternative or complement to subjective techniques reflecting a patient's perception. For example, it has been shown that accelerometers can be used to reliably assess the duration of an activity instead of relying on manual use of a chronometer [159]. Going beyond timing, gait parameters such as speed, stride length and symmetry have also been extracted from accelerometer signals for neurological patients as an indication of functional capacity [154]. This evokes the question whether a similar approach would be possible in a home environment for other medical conditions.

5.4 Linear discriminant analysis and performance metrics

The Chapters in Part II develop approaches for activity recognition and assessment. This section therefore introduces Linear Discriminant Analysis (LDA), a simple classifier and performance metrics that will be used throughout this Part of the thesis.

5.4.1 Linear Discriminant Analysis

Discriminant analysis can be described starting from Bayesian classification [54]. The main goal of a classifier is the selection of the class c_i with the highest posterior probability $p(c_i|\mathbf{x})$, given an observation $\mathbf{x}$. Using Bayes' rule, the objective can also be written as the maximization of $p(\mathbf{x}|c_i)P(c_i)$, or, equivalently, $\ln\, p(\mathbf{x}|c_i) + \ln\, P(c_i)$. In these formulas, $p(\mathbf{x}|c_i)$ refers to the probability of an observation $\mathbf{x}$ given the class c_i and $P(c_i)$ to the (prior) class probabilities.

In case of a binary classification problem, a discriminant function makes a decision between the classes c_1 and c_2. Therefore, it can be written as the difference between the two posteriors, rewritten as:

$$g(\mathbf{x}) = \ln\frac{p(\mathbf{x}|c_1)}{p(\mathbf{x}|c_2)} + \ln\frac{P(c_1)}{P(c_2)} \tag{5.2}$$

The optimization becomes simple if one assumes a multivariate normal distribution of the data, that is, $p(\mathbf{x}|c_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ with $\mathcal{N}$ the normal distribution, $\boldsymbol{\mu}_i$ the mean of class c_i and $\boldsymbol{\Sigma}_i$ its covariance matrix. The specific shape of the discriminant surface $g(\mathbf{x}) = 0$ depends on further assumptions on the class distributions. Two cases can be considered, based on the class covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$

Case 1: $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$
If the covariance matrices are equal, it can be shown that the discriminant surface is a hyperplane given by $g(\mathbf{x}) = \mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0$ with $\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ and $\mathbf{x}_0 = 0.5(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$ if the prior class probabilities are equal. In that case, the surface passes exactly through the middle between the class means. If not, it is shifted to increase the probability of the class with the highest prior. Because of the linearity, the procedure is called Linear Discriminant Analysis (LDA).

Case 2: $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$
In case of arbitrary multivariate normal distributions the discriminant surface

$g(\mathbf{x}) = 0$ is quadratic. The procedure is called Quadratic Discriminant Analysis (QDA). The mathematical details will not be derived here.

Although LDA and QDA have been described for binary classification, they can easily be adapted to a multiclass setting.

5.4.2 Performance metrics

One can make a distinction between measures used for binary or multiclass classification in a standard approach, measures that specifically assess segmentation as a preprocessing step and measures that assess effects of both segmentation and classification.

Classification For classification, standard measures can be used.

- **Accuracy** is defined as $\text{Acc} = \frac{\text{Number of correctly classified samples}}{\text{Total number of samples}}$.
  It is the fraction of correctly classified samples. Its drawback is its bias when classes are unbalanced.
- **False Positives/Negatives** are samples which have been falsely predicted as belonging to the positive/negative class. This can be generalized for multiclass problems by considering a single class as reference (positive) class.
- **True Positives/Negatives** are samples correctly predicted as belonging to the positive, resp. negative class. Once again, this can be generalized for multiclass problems.

Segmentation In case of segmentation, additional metrics can be introduced. First of all, one can quantify how many of the segments of interest have been detected, that is, a detection at least partially overlaps with the ground truth. This is expressed by the Detection True Positive Rate (**DTPR**):

$$\text{DTPR} = \frac{\text{number of correctly detected segments}}{\text{number of ground truth segments}} \tag{5.3}$$

It is the ratio of the number of segments that have (correctly) been accepted and the number of actual informative segments. It is an alternative for the number of false negatives mentioned above if one considers detection/rejection as a binary classification. Whether segments are classified correctly is irrelevant for the DTPR, only the difference between detection and rejection is assessed.

Its counterpart is the number of false detections (**FD**). This quantifies the number of segments detected in the continuous signal that do not overlap with segments defined by the ground truth.

At the segment level, the correctness of the segmentation should also be assessed. This is expressed via the Sørensen-Dice coefficient [51, 151] (**SDC**). Given segments X and Y, it can be defined as:

$$\mathrm{SDC}(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \tag{5.4}$$

Here, set notation has been used: $|X|$ refers to the length of segment X. $X \cap Y$ is the intersection between segments X and Y. The SDC can be considered as a percentwise measure of overlap between X and Y.

Classification and segmentation DTPR, FD and SDC focus solely on segmentation, but the interplay between segmentation and classification can also be quantified by adapted definitions of accuracy, called 'pure' and 'actual':

- The pure accuracy $\mathbf{Acc}_p$ is the classification accuracy when only looking at the detected segments corresponding to ground truth segments. Given S_1 as the set of all detected segments and S_2 as the set of ground truth segments, it can be defined as $\mathrm{Acc}(S)$, the classification accuracy of the set of segments S, with $S = S_1 \cap S_2$. S should be understood as the set of segments in S_1 that at least partially overlap with segments in S_2. The pure accuracy neglects the impact of false (FD) and missed detections (FN).
- The actual accuracy $\mathbf{Acc}_a$ is the most complete measure of performance. Using the same sets S_1 and S_2 as before, it is defined as $\mathrm{Acc}(S^*)$, where $S^* = S_1 \cup S_2$. Hence, S contains all segments of S_1, including FDs, and adds the non-detected segments from S_2 as FNs. The actual accuracy is therefore the classification accuracy taking into account both FDs and FNs as misclassifications.

5.5 Datasets

Several datasets have been gathered as part of the SPARKLE project. The next Chapters build on this data for development and evaluation of algorithms. The first two datasets consist of an activity protocol, performed in the hospital. However, it has been set up in such a way that it can easily be moved to the home environment. The last two datasets consist of specific activities as study cases for automatic extraction of assessment parameters.

5.5.1 SenseWear protocol data

A patient study was performed at the Division of Rheumatology of the University Hospitals, Leuven [159]. The study protocol was approved by the Medical Ethics Committee (ML 5236). The study includes 28 patients diagnosed with axSpA according to the ASAS classification criteria [150], as verified by an ASAS expert. They all gave informed consent. Their functional capacity was estimated using the patient-reported BASFI score prior to the experiments. An overview of the patient population is presented in Table 5.1, taken from [159].

The patients were equipped with a two-axial accelerometer (SenseWear Pro 3 Armband, Bodymedia Inc., Pittsburgh, USA) sampling at 32Hz. It was mounted on the biceps of the dominant arm, its orientation aligned to the longitudinal and transverse axes. The armband is a convenient and non-obtrusive device, easily mounted by the patients themselves. This position was chosen as a compromise between arm and body movement. On the one hand, it is on the upper arm which implies it can capture activities such as reaching. On the other hand, it remains close to the center of mass of the patient and captures mostly the general movement of the body. In contrast, positioning on e.g. the wrist would allow more details on arm movements, but its higher degrees of freedom due to the longer kinematic chain obscure whole-body movements. Figure 5.5 shows a mounted sensor and a patient performing some activities.

Table 5.1: Demographic, anthropometric and disease-specific measures for the SenseWear patient population (n=28) [159]. Non-binary measures are reported with their average (Avg), standard deviation (Std), minimum (Min) and maximum (Max) values. NSAIDS: Non-Steroidal anti-inflammatory drugs, DMARDs: Disease-modifying anti-rheumatic drugs.

Variables	**Avg**	**Std**	**Min**	**Max**
Age (yrs)	43.69	10.45	23.9	60.4
Disease duration (yrs)	12.45	10.52	0	38.7
Height (cm)	169.41	11.53	148.90	193.1
Weight (kg)	75.45	18.52	46.7	120
BMI (kg/m^2)	26.19	5.71	17.48	39.36
BASFI (0-10)	3.14	2.19	0	8.1
	Frequencies (%)			
Gender (male/female)	16/12 (57/43)			
NSAIDs (yes/no)	17/11 (61/39)			
Biologicals (yes/no)	15/13 (54/46)			
DMARDs (yes/no)	9/19 (32/68)			

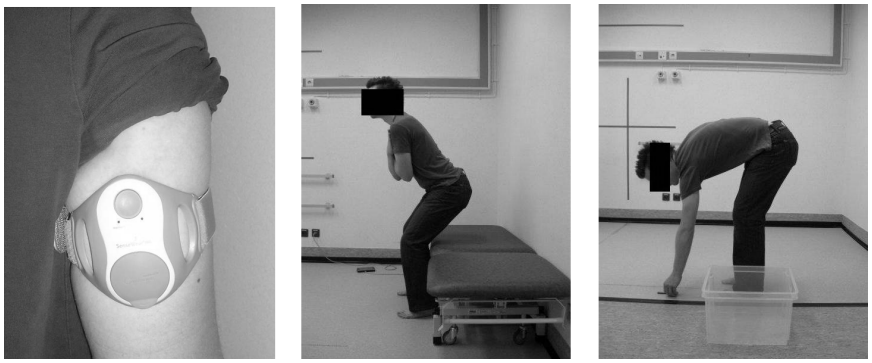


Figure 5.5: Example of a mounted sensor and a patient resp. sitting down and picking up a pen

Table 5.2: List of Sensewear setup activities and their abbreviations

Abbreviation	**Description**
getup	getting up starting from lying down
liedown	lying down starting from stance
maxreach	reaching up as far as possible
pen	picking up a pen from the ground
pen5	repeating pen 5 times, as fast as possible
reach5	touching a mark 5 times, as fast as possible
sts	performing a sit-to-stand movement
sts5	repeating STS 5 times, as fast as possible
socks	sitting down and putting on socks
stairs	climbing stairs (13 steps)

Next, they were instructed to perform a series of ten activities based on the BASFI questionnaire. Table 5.2 lists and describes them. All activities are transitory, focusing on short movements, hence yielding information about the functional capacity. Furthermore, they mostly correspond to situations encountered in daily life: getting up and lying down, reaching up, picking something up etc. However, in several cases, repeated activities were selected to minimize the impact of single-trial variability and to be able to judge possible changes within one sequence. Although the repetition does not correspond to free-living behaviour, it has been shown to yield more clinically valid measures than a single execution [159]. Additionally, the patients were required to perform the activities as fast as possible, to avoid differences due to the self-selected execution speed. Furthermore, to decrease the peripheral influence in whole-body activities, the patients were asked to fold their arms across the chest.

From this description, it should be clear that the activities are part of a semi-controlled setup drawing on current clinical knowledge. It can be performed in the home environment, eliminating the need to go to the hospital for assessment. Hence, the goal is not to derive the parameters in daily life activities, but move to a protocol at home as a first step towards that aim. Every patient performed each activity twice for test-retest assessment in a randomized activity sequence. Hence, in total, 560 trials were measured, 56 for each activity class.

Activities were manually segmented and labelled by master students of the Faculty of Rehabilitation Sciences, KU Leuven, supervised by a physical therapist. Directions were given beforehand and the results were checked by the therapist to minimize the impact of individual raters. With this process in mind a custom segmentation toolbox was implemented in Matlab as described in Appendix C.

5.5.2 Shimmer protocol data

This dataset results from a continuation of the previous data study, also performed at the Division of Rheumatology, University Hospitals, Leuven. The same inclusion criteria, ethical approval and informed consent apply. Details on the 11 patients are shown in Table 5.3

Table 5.3: Demographic, anthropometric and disease-specific measures for the Shimmer patient population (n=11). Non-binary measures are reported with their average (Avg), standard deviation (Std), minimum (Min) and maximum (Max) values. NSAIDS: Non-Steroidal anti-inflammatory drugs, DMARDs: Disease-modifying anti-rheumatic drugs.

Variables	**Avg**	**Std**	**Min**	**Max**
Age (yrs)	42.69	9.3	23.53	52.21
Disease duration (yrs)	17.41	10.31	2.08	38.81
Height (cm)	175.28	14.90	148.7	190.7
Weight (kg)	80.55	20.74	49	114.5
BMI (kg/m^2)	25.86	4.23	20.26	32.99
BASFI (0-10)	2.49	4.23	0	7.1
	Frequencies (%)			
Gender (male/female)	7/4 (63/37)			
NSAIDs (yes/no)	3/8 (27/73)			
Biologicals (yes/no)	8/3 (73/27)			
DMARDs (yes/no)	2/9 (18/82)			

Table 5.4: List of Shimmer setup activities and their abbreviations

Abbreviation	Description
cervrot	cervical rotation, left and right
getup	getting up starting from lying down
pen5	repeating pen 5 times, as fast as possible
reach5	touching a mark 5 times, as fast as possible
sts5	repeating STS 5 times, as fast as possible
socks	sitting down and putting on socks
stand	stand upright for 2 minutes
stairs	walking a staircase (up and down)
walk	walk a fixed distance

The patients were equipped with six tri-axial inertial measurement units (Shimmer3, Shimmer Sensing, Ireland) with a sampling frequency of 32Hz placed at the middle of the forehead, the lateral side of both femurs, the spinal thoracic (T3) and lumbar (L3) processes and the triceps of the dominant arm. The arm sensor was mounted such that two axes corresponded with the setup of the SenseWear setup described earlier. Other sensors were used because, on the one hand, the SenseWear was no longer supported and, on the other hand, the need to record and synchronize multiple body positions.

The patients were subsequently subjected to an activity protocol with nine activities in randomized order. The set does not fully coincide with the SenseWear protocol but covers among others also cervical rotation. An overview is presented in Table 5.4.

The activities were segmented and labelled with the custom-built Matlab toolbox under the supervision of a physical therapist, similarly to the previous dataset.

5.5.3 Motor control parameter data

This dataset consists of long-term monitoring of physical activities using two ViMove sensor units (ViMove, Dorsavi, Victoria, AUS) sampling at 20Hz. The units each contain a tri-axial accelerometer, magnetometer and gyroscope. Proprietary algorithms convert the sensor readings to orientation angles along the sagittal and the transverse axis, hence, in the frontal, resp. sagittal plane.

The data setup consists of the monitoring of 27 subjects, 24 healthy controls and 3 axSpA patients (2 male, 1 female; age: 32 $\pm$5.6 years). Due to technological problems, two healthy subjects were excluded from the study. All subjects

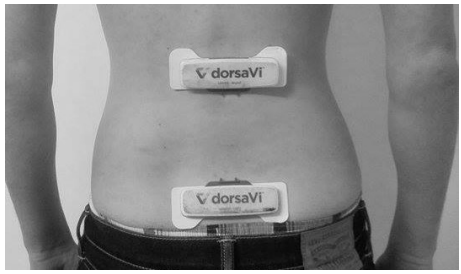


Figure 5.6: Positioning of the ViMove units.

signed an informed consent document prior to participating. More patient details can be found in [47].

All subjects were required to wear the ViMove for the duration of a normal working day for a minimal duration of ten hours. The lower unit was placed on the second sacral vertebra (S2), the upper unit on the third lumbar vertebra (L3) as shown in Figure 5.6. Apart from the normal activities, all subjects were required to perform an activity protocol in three sessions. Two sessions, at the beginning and the end of the monitoring period, were supervised at the Division of Rheumatology, University Hospital, Leuven. In between, an unsupervised session was recorded as well. A session consists of 14 posture and movement tasks listed in Table 5.5. The description lists both the activity and the duration for which a static pose needed to be maintained. For example, the patients were asked to bend forward, maintain that pose for three seconds and bend back. The study was approved by the local ethics committee (ML 5236). The recordings were exported from the proprietary software and the sessions were segmented using the custom-built Matlab toolbox described in Appendix C.

Table 5.5: Activities performed in the motor control parameter study.

Activity	**Description**
extension	3× bending backwards for three seconds
flexion	3× bending forwards for three seconds
functional bow	folding an A4 paper on a low chair
lateral flexion, left	3× bending left for three seconds
lateral flexion, right	3× bending right for three seconds
rotation, left	3× left axial rotation for three seconds
rotation, right	3× right axial rotation for three seconds
sitting	five types of sitting, each 5 seconds
stance	Standing straight for five seconds
squat	maintain a squat position for 10 seconds

5.5.4 Timed Up-and-Go data

The Timed Up-and-Go test is considered a reliable indicator for increased fall risk, or, more generally, functional disability in older adults [149]. The test requires a patient to sit on a chair, get up, walk 3 metres, turn, walk back and sit down again. Walking aids can be used if necessary.

The test was performed by healthy controls and patients equipped with two triaxial Shimmer Sensors sampling at 100Hz, one attached to the leg and one to the torso. The duration was measured using a chronometer and the number of steps was assessed based on video data by three independent assessors. The average of their estimations is used as ground truth. All participants signed an informed consent. The test protocol was approved by the local ethical committee (ML 5236).

Firstly, 10 healthy controls performed the test. Each person performed the test twice. Also, 54 hip fracture patients were measured after surgical intervention. However, some patients were unable to finish the experiment. Moreover, several recordings proved invalid due to technical problems e.g. loss of the wireless sensor signal. Excluding these leads to a final set of 19 patients. Every included patient performed the test at least once. Sometimes, two or three annotated repetitions are available. All patients used a walking aid for the test, either a walker (2), a rollator (13), two crutches (2) or something else (2). Details on the participant population are given in Table 5.6.

Table 5.6: Demographic and anthropometric measures for the TUG subjects population (n = 10 healthy, 19 hip fracture patients). Non-binary measures are reported with their average (Avg) and standard deviation (Std).

	Controls (n=10)		**Patients** (n=19)	
Variables	**Avg**	**Std**	**Avg**	**Std**
Age (yrs)	22.8	1.32	76.98	11.68
Height (cm)	174.05	9.95	162.2	9.56
Weight (kg)	74.34	11.1	64.21	12.26
BMI (kg/m^2)	24.53	3.14	23.32	3.9
Gender (m/f)	4/6		7/12	
	Patient-specific measures (n=19)			
Surgery side (left/right)	11/8			
Walking Aid (yes/no)	19/0			
Mental problems (yes/no)	4/15			
Intravenous drugs (yes/no)	5/14			

One forms provisional theories and waits for time or fuller knowledge to explode them. A bad habit, Mr. Ferguson, but human nature is weak.

A. C. Doyle, *The Adventure of the Sussex Vampire*

Chapter 6

Activity recognition from statistical features complemented with patterns

Statistical features have often been used for activity recognition. They have proven to offer high discriminative power for cyclic activities such as walking. Yet, for transitory activities they might not be sufficient. These are hypothesized to be better captured using patterns, which, in turn, are known to be less robust. This Chapter starts from the most common approach. It aims to investigate whether performance can be improved by complementing the statistical features with features extracted from patterns into one classifier, a novel approach. To that end, Section 6.1 first introduces the statistical and pattern features as well as the approach to segmentation. It finishes with a description of the algorithms built from these features and applied on axSpA patient data in the experiments. The next two Sections list the results and discuss them in more detail, respectively. The studies discussed in this Chapter have previously been published in [22, 23].

6.1 Features and methods

Combining statistical features with pattern matches hinges on the assumption that the activities can be isolated through segmentation. In contrast to a general sliding window approach the beginning and ending of the activities need to be known. Subsequently, statistical features can be extracted as in a

classical setting, whereas patterns can also be applied on the same segmentation, yielding pattern similarity features. This Section therefore first discusses both statistical and pattern approaches separately, after which segmentation is discussed. Finally, the validation experiments are described.

6.1.1 Statistical features

When segments are available, the standard approach used in window-based algorithms can be applied. Only, instead of defining features for a sliding window they can be extracted directly from the activity segments, since these are the only regions of interest. The following features, mostly calculated per data axis, are common in literature. A segment S of N samples and sampling frequency f_s is assumed. In general, a data segment can consist of multiple acceleration channels. S^k is used to refer to the kth channel of segment S, whereas S_i^k refers to the ith sample of this channel.

- **duration**: besides being an important performance-based marker for the health of a patient, this feature might also help to discriminate between activities. It is defined as N/f_s.
- **mean**: mostly important to distinguish poses or pose changes due to the orientation of gravity.
- **standard deviation**: captures the spread of the data.
- **range**: another measure for the spread of the data, but this is more influenced by extreme signal values, e.g. sudden peaks.
- **3 subpart means**: the segment can be divided in three uniform intervals (indices $1, \ldots, N/3$, $N/3{+}1, \ldots, 2N/3$ and $2N/3{+}1, \ldots, N$, rounded to integer values). The interval means are a coarse representation of signal shape. Due to the coarseness, these features are somewhat robust against the trial-to-trial and subject-to-subject variability.
- **power**: a measure for the intensity of the activity.
- **power spectral entropy (PSE)**: yields information on the spectral structure or complexity of the signal. For channel k, it is defined as $SE^k = -\sum_i PS_i^k \log(PS_i^k)$, with PS the normalized power spectral density and log Brigg's logarithm.
- **line length**: another measure for signal complexity. For channel k, it is defined as $L^k = \sum_{i=2}^{N} |S_i^k - S_{i-1}^k|$.

- **autocorrelation**: allows to separate repetitive and non-repetitive tasks as it is an indicator of self-similarity. One way of quantifying this is taking the average autocorrelation over a limited number of lags.

6.1.2 Pattern-derived features

Static patterns have the disadvantage of limited applicability since they cannot take variability into account. Therefore, it is common practice to apply Dynamic Time Warping discussed in the previous Chapter. Yet, two extensions are needed for most activity data. Firstly, a trial often consists of multiple channels instead of one. For example, the sensors used in this thesis often yield a biaxial output. These two channels have to be matched jointly between channels (multi-channel matching). Secondly, more than two segments should be matched to obtain common patterns over all training data (multi-segment matching). This has been implemented in a Matlab toolbox by Zhou and De la Torre [183].

Pattern matching then consists of two steps. Firstly, the multi-channel examples in the training data are all matched simultaneously for each class. This yields aligned versions of the examples. The average of the aligned signals is used as a template to represent the class. In many settings, one is interested in shape rather than magnitude. Therefore, signals are standardized first. This procedure yields a two-channel template for each class. Subsequently, these templates can be used to extract features. Any segment can be matched against all class templates. Therefore, the segment can be characterized by its similarity to these templates. Two types of features are extracted:

- DTW yields a matching cost, the distance between the matched signal and the original pattern. This gives one feature per class.
- One can also look at correlation rather than at distance. For each class and each channel, Pearson's correlation between the pattern and the matched segment is computed. This procedure yields a number of features equal to the number of classes multiplied by the number of channels. These features allow to make certain channels more important than other when training a classifier.

The simplest way of using the patterns in a classification setting would be to just vote for the closest match, either using correlation or distance as similarity metric. However, using all distances and correlations yields more information to base a classifier's decision on.

6.1.3 Segmentation

Each subject is required to go through a sequence of activities, yielding a continuous acceleration signal. A first step in activity recognition is therefore to segment this signal into potentially interesting activities. As the previous Chapter mentioned, this can for example be achieved based on the energy or variance of the signal e.g. assuming periods of rest between activities. Dynamic regions containing activities can then be derived via the following three steps:

1. **Rough segmentation** A first phase provides a rough per-channel segmentation. The signals are divided in windows of one second, with 50% overlap. The standard deviation and range are then compared to empirical thresholds. The regions for which both thresholds are exceeded are marked as 'dynamic'.

2. **Refining** Then, the initial segmentation is refined based on the variance of the static regions (regions between two dynamic segments). Shrinking and extending with a quarter of a second are considered. The decision is based on the difference in variance between half-second regions. The initial and final half second of a static region serve as baselines. For the start of the static segment, extending is accepted if the half second starting at a quarter second before the current start has a variance which is maximally 10% higher than the baseline. This tries to grow the static regions avoiding incorporating too much movement. Shrinking is accepted if the variance of the half second starting at a quarter second later than the current start is at least 10% lower. This tries to eliminate movement at the start of the region. For the end of the region, the procedure is identical. The value of 10% was chosen empirically as an acceptable deviation.

3. **Merging** After refinement of the segmentation, the channels are joined. A region is considered dynamic if one of its channel regions is dynamic. Furthermore, if regions are less than half a second apart and their mean is similar, they are joined. Finally, dynamic segments of less than one second are discarded.

However, one cannot always assume that only known activities occur, that is, the segmentation will also detect movements that should be discarded. This can be handled via a rejection layer in the classifier.

6.1.4 Algorithms

The merging of patterns and features is compared to the individual types via two experiments on the SenseWear dataset described in Section 5.5. 28 axSpA patients performed ten activities based on the BASFI questionnaire, recorded via an arm-worn two-axial strap-on accelerometer sampling at 32Hz.

6.1.4.1 Patterns and features in a closed world

The first experiment makes the assumption of a closed world. The segmentation is assumed to be given and all possible activities are known. All ten activities are considered for recognition. The study is conducted in a leave-one-subject-out (LoSo) approach: all patients' data, except for one, is used to train the model. This is then evaluated on the remaining subject. This is repeated until every subject has been used for testing once.

The experiment compares four algorithms, summarized in Figure 6.1. From the training data, activity patterns are obtained via DTW as described above. Then, both training and testing segments follow the flow outlined in the Figure for all four algorithms.

- **algo 1 (linSt)** At training time, statistical features (St) are extracted from training segments and used to train an Linear Discriminant Analysis (LDA) classifier. More specifically, it extracts the mean, standard deviation, three subpart means, power, power spectral entropy and autocorrelation for both data channels, leading to 16 features. The classifier is subsequently used for prediction.
- **algo 2 (purePat)** DTW aligning of the standardized training segments yields activity patterns. New (standardized) segments can be matched against these patterns. The class pattern closest to the test segments indicates the class. This similarity is assessed via the Euclidean distance between the warped patterns and test segment.
- **algo 3 (linPat)** Algo 3 uses the same activity patterns as algo 2. Next, each training segment is matched against all patterns. This creates a new training set: the resulting Euclidian distances to each pattern are the training features. An LDA classifier is trained with this new training set. Test segments are converted to the feature space by DTW to the activity patterns and fed into the LDA classifier for prediction.
- **algo 4 (linPatSt)** Algo 4 merges algo 1 and algo 3 via early fusion. The feature data (St features and pattern distances) are concatenated. An LDA classifier is trained on this extended training data set.

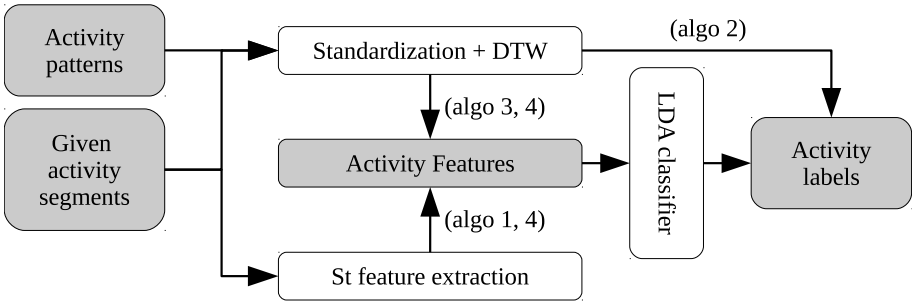


Figure 6.1: Flowchart for the closed world algorithms. All algorithms start from activity patterns and manually segmented activities. These segments are given as input to DTW, together with the patterns, and to the statistical feature (St) extraction. Not all algorithms make use of all features, as indicated by the labelled arrows. Algo 1 uses St features and the LDA classifier. In contrast, algo 2 directly predicts based on the DTW. Algo 3 uses the DTW features with the LDA classifier. Finally, algo 4 fuses St and DTW features for use with the classifier.

In algo 1, 3 and 4 the final classification is performed by means of the LDA classifier introduced in Section 5.4.1. This is also visualized in Figure 6.1. Since the amount of data is rather small for 10 classes, this linear classifier was selected. It has the advantage of model simplicity, which limits the risk of overfitting, particularly important with a limited data set.

The dataset is perfectly balanced, consisting of 56 observations of each class, two trials of each activity for all 28 patients. Hence, the performance can be reliably assessed using LoSo activity recognition accuracy. The nature of LoSo then allows to statistically compare pairs of algorithms using a paired t-test [114] with a null hypothesis of equal distribution means and a two-tailed alternative hypothesis. To further study the differences between classes, a confusion matrix over all patients will be provided.

6.1.4.2 Patterns and features with segmentation

The second experiment consists of a full setup including automatic segmentation. It also uses the SenseWear data, but a statistical study on the dataset has shown that not all activities are equally valid in terms of test-retest reliability [159]. Therefore, the set is reduced to six activities discussed in Chapter 5, Section 5.5: getup, liedown, maxreach, pen5, reach5 and sts5. This leads to a total of 336 trials, again 56 per activity.

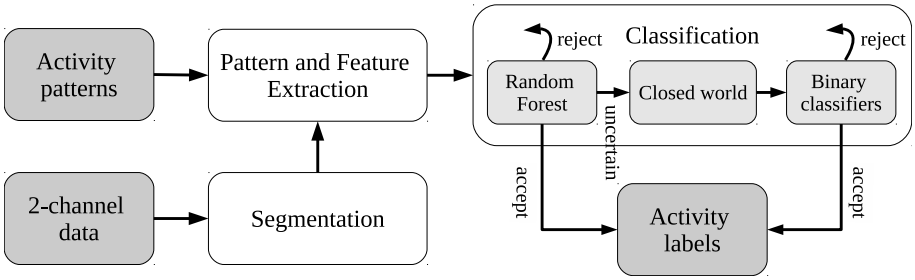


Figure 6.2: Flowchart of the full activity recognition approach including segmentation. At the left, the input data and training patterns are shown. The data is automatically segmented and statistical and pattern-based features are extracted. These are then used for prediction in a multistage framework to allow for rejection of non-informative segments.

The experimental approach is similar to the previous experiment, but it includes automatic segmentation instead of using the ideal manual segments. As a result, some segments will need to be discarded: the automatic segmentation detects not only activities of interest but includes e.g. the four activities that have been left out for this study and random movements of the patients. The full algorithm is shown in Figure 6.2. The details are discussed in what follows.

The features differ from the previous experiment since they should not only distinguish between classes, but are also used for rejection. Here, all features discussed above are used. It leads to 18 pattern features, the matching distances for all six activities and correlations for both axes for all activities. Furthermore, the definitions above lead to a total of 21 statistical features: the duration (1), means (2), subpart means (6), standard deviations(2), power(2), range(2), line length(2), spectral entropy(2) and average autocorrelation(2). All in all, the full feature vector consists of 39 features.

Class patterns and training features are extracted from the manually annotated training dataset. The segmentation is also applied on this training data. By comparison of the detected and manually segmented regions, one also finds representations of the so-called NULL or rejection class, an amalgam of all movements that should be rejected.

The classification including rejection consists of three stages as can be seen in the right part of Figure 6.2.

1. **Random forest with rejection**
   In the top layer, a Random Forest (RF) classifies data according to the seven classes. The name stems from the observation that it is an ensemble consisting of Decision Tree classifiers. Although pruning can be applied to

improve generalization, the variance in the data can be approximated in a better way by growing multiple trees with only a part of the data, often limiting their growth to a fixed level. This kind of Trees are nicknamed *stumps*. This study uses Matlab's *TreeBagger* with 250 stumps, each created by sampling 85% of the training data to improve generalization. An important characteristic is its prior: it favours the rejection class with a 2:1 ratio compared to all other classes. In this case, if the RF does not predict the rejection class despite its higher weight, this decision is most probably correct. Default values were used for all other parameters.
Three outcome categories can be considered: activities that are recognized are considered correct, the ones that are rejected with high posterior probability (> 0.7) are immediately rejected (the leftmost 'reject' in Figure 6.2), the remaining rejections should be reconsidered ('uncertain').

2. **A closed world assumption**
   For the uncertain observations, a multiclass linear discriminant (LDA) model is trained under the closed world hypothesis, that is, the rejection class is not considered. This is in fact identical to the previous experiment. The predicted labels indicate the class for a particular data point, under the assumption it is not rejected.

3. **Activity-specific binary classifiers**
   Finally, the observations indicated as uncertain by the Random Forest can be reconsidered. To this end, 6 binary LDA models are trained, each with one activity class and the rejection class as possible outcomes. In the previous step, observations to be reconsidered have been attributed a label under the closed world assumption. This determines which model to use for its final acceptance or rejection. Eventually, this yields the final label for the remaining observations.

Assessment in this experiment does not compare to a model only consisting of patterns or statistical features. Instead, it uses different measures to evaluate several aspects of the single implemented algorithm. Each patient has 12 activity segments to be detected, since all six activities are performed twice. The evaluation should answer a number of questions: are all segments detected? Do we have many false detections? How precise is the detection? Are the detected segments recognized correctly? Measures to assess all of this have been introduced in Section 5.4.2.

The first and second question are answered by considering a detected segment as correct if it overlaps with the ground truth indicated by the physical therapists. This is quantified via the Detection True Positive Rate (DTPR). Furthermore, the number of false detections FD indicates the sensitivity of the segmentation.

To answer the third question one should focus on the correctly detected segments to assess the precision of the segmentation. Typically, this is evaluated using the Sørensen-Dice coefficient (SDC) as explained in Section 5.4.2.

The last question does not only focus on segmentation as in the previous parts, but instead aims to quantify the discrimination between classes. This is evaluated in two ways. On the one hand, the pure accuracy Acc_p is calculated using the correctly detected segments. On the other hand, the actual accuracy Acc_a considers the seven-class problem, including the rejections as misclassifications, as explained in Section 5.4.2.

6.2 Results

6.2.1 Patterns and features in a closed world

The LoSo accuracy of all algorithms for all patients is displayed in Table 6.1 (chance level is 0.1). The last rows show the average accuracy and the standard deviation. The best algorithm for each subject is highlighted in bold. Based on these results, conclusions can be drawn, backed up by statistical analysis. The pure pattern matching clearly performs worst, though not in all cases. It has a moderate average accuracy with a high standard deviation. Its accuracy is lower than linSt ($p = 0.0194$) and significantly lower than linPat ($p = 6\text{E-}4$). The latter indicates similarity to multiple patterns is more conclusive than only looking for the closest match. Comparison of linSt and linPat is not conclusive ($p = 0.725$). Indeed, both their average accuracies and standard deviations are similar as well. Finally, linPatSt significantly outperforms all others (p values of 0.0021, 1.4E-6 and 5.2E-5, respectively). Although its performance is comparable to linSt for many patients, its lower standard deviation shows it is able to be more robust to variability by fusion of complementary information. This is particularly clear for subject 26: though all methods have low performance, the fusion allows to obtain an acceptable accuracy of 70%.

Table 6.2 shows the pooled confusion matrix for linPatSt. The class-averaged sensitivity is 93.6% (std 6.4%). The specificity is 99.3% (std 0.8%). As can be observed, liedown and reach5 are always identified correctly (perfect sensitivity). *sts5* has the lowest sensitivity, 80.4%, because it is often wrongly identified as climbing the stairs. Recalling the sensor is mounted on the arm, the movements can seem similar. The templates for the movements are similar as well, but different enough to be distinguished by a human observer. However, they generate similar statistics. Experiments show that indeed, linSt performs worse than linPat or linPatSt for the *sts* activity. The confusion matrix further shows the decision boundary favours stairs over *sts5*, since *stairs* is not that often

Table 6.1: LoSo accuracies of all algorithms for all patients. The last row shows the significance of the difference between the algorithms, indicated by their column index. For example, (2,4) for linSt indicates it performs significantly different from purePat and linPatSt.

	linSt	**purePat**	**linPat**	**linPatSt**
Patient 1	0.65	0.85	0.85	**1**
Patient 2	0.9	0.7	0.8	**0.95**
Patient 3	0.7	**0.95**	**0.95**	0.9
Patient 4	**0.8**	0.6	0.7	**0.8**
Patient 5	0.8	0.95	**1**	**1**
Patient 6	**0.9**	0.4	0.7	0.85
Patient 7	**0.85**	0.55	0.8	0.8
Patient 8	0.9	0.75	0.85	**1**
Patient 9	0.8	0.9	**0.95**	0.9
Patient 10	**0.9**	0.8	0.8	**0.9**
Patient 11	**0.95**	0.3	0.5	0.85
Patient 12	0.95	0.85	**1**	**1**
Patient 13	**1**	0.9	0.8	**1**
Patient 14	**1**	0.9	0.95	**1**
Patient 15	**1**	0.95	**1**	**1**
Patient 16	0.85	**1**	0.95	**1**
Patient 17	0.75	0.55	0.75	**0.8**
Patient 18	**0.95**	0.8	0.85	**0.95**
Patient 19	0.9	0.95	**1**	**1**
Patient 20	0.9	0.9	0.95	**1**
Patient 21	0.9	0.45	0.8	**0.95**
Patient 22	0.9	0.85	0.95	**1**
Patient 23	0.9	0.8	0.9	**1**
Patient 24	**1**	0.45	0.75	0.9
Patient 25	0.95	0.85	0.95	**1**
Patient 26	0.4	0.55	0.55	**0.7**
Patient 27	**1**	0.85	0.9	**1**
Patient 28	0.75	0.9	**1**	0.95
avg	0.866	0.759	0.856	**0.936**
std	0.131	0.196	0.133	**0.084**
$p \leq 0.05$	(2,4)	(1,3,4)	(2,4)	(1,2,3)

Table 6.2: Pooled confusion matrix over all patients for linPatSt.

		Predicted labels									
		getup	**liedown**	**maxreach**	**pen**	**pen5**	**reach5**	**sts**	**sts5**	**socks**	**stairs**
Actual labels	**getup**	54	0	0	0	0	0	0	0	2	0
	liedown	0	56	0	0	0	0	0	0	0	0
	maxreach	0	1	55	0	0	0	0	0	0	0
	pen	0	0	0	48	0	0	6	0	2	0
	pen5	0	0	0	0	51	2	0	0	1	2
	reach5	0	0	0	0	0	56	0	0	0	0
	sts	0	0	0	4	0	0	52	0	0	0
	sts5	0	0	0	0	0	0	0	45	0	11
	socks	0	0	0	1	0	0	1	0	54	0
	stairs	0	0	0	0	0	0	0	2	1	53

misclassified as *sts5*. The same argument can be applied for the confusion between *pen* and *sts*: the shapes are more dissimilar than the St features. Yet, for both cases, combining statistical features and patterns (linPatSt) still yields an advantage (up to 5%) over either of the single classifiers as can be derived from a comparison of the confusion matrices of all algorithms (not shown).

6.2.2 Patterns and features with segmentation

The second experiment includes segmentation as a factor, but only assesses a single approach. Table 6.3 presents the LoSo results, both segmentation and classification, as a summary per patient. Table 6.4 shows the segmentation performance for all activities.

To assess if all necessary segments are detected and how many false detections occur, Table 6.3 shows the DTPR and the number of false detections (FD). The Table shows that the method obtains an average DTPR of 98.5%. Missed detections only occur for 3 out of 28 patients. The Table also highlights the average FD of 0.6. In half of the cases no false detections occur. For the remaining 14 cases, 12 have a single false detection. The remaining two patients have two or three false detections. The per-activity overview in Table 6.4 shows that *getup* and *maxreach* have the highest number of false detections (6) over all patients, whereas *liedown*, *reach5* and *sts5* are never falsely detected.

The segmentation quality, judged by the SDC, is very good. Over all patients, a mean SDC of 0.92 is obtained. The segmentation quality per activity indicates that repeated activities are generally segmented with a higher SDC.

Finally, the pure accuracy Acc_p of the system is very high. The detected activities are recognized with an average accuracy of 99.4%. For only two patients, recognition is imperfect. Taking into account the false detections leads to a drop of about 6% to 93.5% actual accuracy Acc_a. Still, for 11 patients, recognition remains perfect. This means all activities were detected, no segments were falsely detected (or they were correctly rejected) and all segments were classified correctly. Only four patients have an actual accuracy lower than 90%.

Some insights can be gained by comparing the closed world and the segmentation experiment, particularly for the patients with low accuracies. For example, it is clear that the slightly lower recognition accuracy for patient 4 is shared between the experiments. Another example is patient 26. The results are the worst of all patients in both experiments. However, in the second experiment this is not directly due to misclassification. As can be seen from the DTPR and the pure accuracy, the harder trials are not misclassified as other activities, but simply rejected.

Table 6.3: LoSo recognition performance in terms of number of False Detections (FD), Detection True Positive Rate (DTPR), average Sørensen-Dice Coefficient (SDC), SDC standard deviation (stdSDC), pure accuracy (Acc_p) and actual accuracy (Acc_a.

	FD	**DTPR**	**avgSDC**	**stdSDC**	**Acc_p**	**Acc_a**
Patient 1	1	100%	0.94	0.05	100%	92.3%
Patient 2	0	91.7%	0.93	0.09	100%	91.7%
Patient 3	1	100%	0.94	0.04	100%	92.3%
Patient 4	1	100%	0.95	0.03	91.7%	84.6%
Patient 5	0	100%	0.93	0.09	100%	100%
Patient 6	1	100%	0.95	0.03	100%	92.3%
Patient 7	1	100%	0.92	0.05	100%	92.3%
Patient 8	0	100%	0.84	0.11	100%	100%
Patient 9	0	100%	0.93	0.05	100%	100%
Patient 10	1	100%	0.91	0.08	100%	92.3%
Patient 11	3	100%	0.92	0.07	100%	80.0%
Patient 12	1	100%	0.92	0.06	100%	92.3%
Patient 13	0	100%	0.92	0.08	100%	100%
Patient 14	0	100%	0.95	0.03	100%	100%
Patient 15	1	100%	0.91	0.08	100%	92.3%
Patient 16	0	100%	0.87	0.08	100%	100%
Patient 17	0	91.7%	0.93	0.04	100%	91.7%
Patient 18	0	100%	0.91	0.06	100%	100%
Patient 19	2	100%	0.91	0.08	100%	85.7%
Patient 20	0	100%	0.96	0.04	100%	100%
Patient 21	0	100%	0.93	0.06	100%	100%
Patient 22	1	100%	0.90	0.10	100%	92.3%
Patient 23	0	100%	0.93	0.04	100%	100%
Patient 24	0	100%	0.92	0.08	100%	100%
Patient 25	1	100%	0.90	0.12	100%	92.3%
Patient 26	1	75%	0.90	0.09	100%	69.2%
Patient 27	1	100%	0.93	0.05	100%	92.3%
Patient 28	0	100%	0.88	0.08	91.7%	91.7%
Average	0.6	98.5%	0.92	–	99.4%	93.5%

Table 6.4: Per-activity segmentation performance in terms of number of False Detections, Detection True Positive Rate, average Sørensen-Dice Coefficient, SDC standard deviation.

Activity	**FD**	**DTPR**	**avgSDC**	**stdSDC**
getup	6	98.2%	0.92	0.07
liedown	0	96.4%	0.89	0.10
maxreach	6	100%	0.87	0.06
pen5	5	96.4%	0.95	0.06
reach5	0	100%	0.94	0.04
sts5	0	100%	0.95	0.05

Another insight stems from the activity selection, based on an earlier statistical study [159]. The pure accuracy in the second experiment improves the first experiment's accuracy. On the one hand, it is due to a lower number of classes, but it also indicates the activities that have been removed can indeed be considered confounders. As a general example, 14 patients had perfect recognition in the first experiment for linPatSt (Table 6.1). In contrast, Table 6.3 has 23 patients with perfect DTPR and perfect pure accuracy. The actual accuracy is only diminished later due to false detections. As a more specific example, in the first experiment, *sts5* was often wrongly identified as *stairs* and *pen* had a lesser recognition rate as well (Table 6.2). Their removal has therefore a positive impact on the system. Yet, they are still present in the rejection class in the second experiment. One can therefore assume that the more powerful Random Forest with rejection improves the performance of the system.

6.3 Discussion

As indicated in the previous chapter, explicit pattern matching has some disadvantages. Moreover, the abundant positive results in literature seem to suggest window-based approaches are well suited to solve the activity recognition problem. Then, why still use patterns? It is mostly a principled choice, rooted in the observed clinical reality. Physical therapists reason on how a movement is being performed. When presented with accelerometer data, this is the way they look at it. Ideally, orientation, rather than acceleration patterns, would be the primary way of presenting the information. But even with acceleration patterns, points of interest can be defined. E.g. how long does it take a patient to get up, what is the average acceleration when bending over etc. Clinicians can reason on patterns, particularly when these are complemented

with video recordings. Therefore, the fusion approach developed in this work tries to combine the best of both worlds: patterns are being matched to retain the interpretability and identify the specific interesting parts of the data, but statistical features leverage their classification performance. The results show that this fusion clearly influences the results. The first experiment indicates statistically significant differences, whereas the second one asserts that the performance is not lost when segmentation is involved.

This pilot study was deliberately kept simple to be easily applicable in practice. This is reflected in the use of a single sensor, worn around the upper arm. It is also reflected in the choice of a limited set of six to ten activities. Although the setup remains somewhat restricted due to the protocol, it can already be moved to a home environment. As an alternative to the sensor, it has been shown that sensors included in some current smartphones are equally accurate as consumer wearables [119]. In that sense one could replace the SenseWear with e.g. a smartphone mounted in a typical sports armband used by runners. This has the additional advantage of gaining access to a third acceleration axis, since most phones have triaxial accelerometers. This could further improve the results. One might also look into smartwatches, located on the wrist. In that case, one would need to investigate the effect of sensor placement in the current setting. On first sight, the upper arm appears most interesting since it remains close to the center of gravity while also capturing arm motion. Wrist sensors, in contrast, include several additional degrees of freedom, yielding more information, but also complicating the recognition task. Yet, many fitness trackers are wristbands, although they mostly seem to focus on walking and energy expenditure. Hence, it is an interesting issue for further study. For now, a first next step would be the extraction of parameters or, more generally, a system to assess the activity capacity or disease progression of patients. For example, patients could take some time every day to perform the protocol. In such a setting, the activities can be recognized and important informative parameters can be extracted. When the data is transferred to the therapist, he or she can follow the evolution of the parameters in between scheduled therapy sessions. This kind of systems will be discussed in more detail in Chapter 8.

Particularly with respect to this future assessment step, the experimental results are promising. This reflects in the good recognition results in the closed world experiment, but it extends to the approach with segmentation as well. The high performance of the recognition algorithm ensures correct feature values for the assessment phase. For example, the high Sørensen-Dice Coefficients for segmentation show that activity duration is measured accurately. False detections and misclassifications are of special importance, since they will yield wrong inputs to a future assessment system. The experiments show that the number of false positives is limited, though some patients are affected more than others (see Table 6.3). Moreover, the closed world accuracy of 93.6% and

the high pure accuracy of 99.4% for six activities indicates that classification is nearly always correct if the segmentation algorithm yields the right outcome. Missed activities are less important compared to spurious detections: a clinician would rather discard information than base decisions on wrong detections . Yet, the algorithm has shown to perform well here as well. Both missed and spurious detections are taken into account for the actual accuracy of the recognition. Consequently, the latter can only be lower. As an example, Patient 26 in Table 6.3 has several missed detections and one false detection. The correctly segmented activities are perfectly recognised, but the actual accuracy drops to 69.2%. Therefore, the actual accuracy is the correct measure to evaluate the recognition's usefulness.

All approaches in this work are subject-independent. This is a pragmatic choice. Since only two observations of each activity are available for any subject, even including half of it would only mean one data point would be the subject-dependent part. Yet, additional experiments show this already leads to a minor improvement, especially for subject 26 in the first experiment. Another way to estimate the influence of between-subject variability is to check whether misclassifications are tied to patients, that is, if both instances of the patients are misclassified, rather than only one. In the latter case, the within-subject variability would be more important. As it turns out, this test is inconclusive: for example, for the first experiment, in 11 cases, both trials are misclassified; in 14 cases, only one of the trials of a subject is misclassified. On the other hand, one can also observe that in particular cases, bad performance of a subject is indeed tied to a misclassification of both trials. Generally speaking though, this claim cannot be proven. This is a positive observation: recognition of the activities works well for all patients without the need for patient-dependent evaluation. No patient-specific training is required.

It is difficult to compare the work in this chapter to other results in literature due to its novel and specific approach, its focus on transitory activities and the fact that it is intended as a stepping stone for assessment of activities rather than only focusing on activities, which might influence performance metrics and priorities. Still, one can assume that given the same data, more powerful algorithms would yield better results, but at the cost of interpretation.

6.4 Conclusion

The main aim of this Chapter was the investigation of the hypothesis that classification with commonly used statistical features could be improved with DTW-derived features, more suitable for transitory activities. It confirms this hypothesis with a study on 28 axSpA patients.

Comparison of four simple algorithms in a closed world setting shows that merging the features improves accuracy with respect to a naive DTW comparison, a more complex DTW classifier and a classifier based on statistical features alone. The combined recognition system for 10 activities leads to an average accuracy of 93.6%.

Secondly, the Chapter further indicates the viability of activity recognition in a more realistic framework applied on continuous acceleration signals. This required the implementation of a heuristic segmentation algorithm and a novel multistage classification approach combining a random forest with linear classifiers to allow for rejection of false detections. Recognition of six activities leads to an average accuracy of 93.5%, taking into account false detections and misclassifications.

Overall, this Chapter shows the viability of activity recognition in the proposed protocol setting. It can therefore be used as a framework underlying activity assessment, discussed in Chapter 8.

There is in all things a pattern that is part of our universe. It has symmetry, elegance, and grace. We try to copy these patterns in our lives and our society, seeking the rhythms, the dances, the forms that comfort.

F. Herbert, *Dune*

Chapter 7

Activity recognition using patterns

Chapter 5 hypothesized that transitory activities might best be captured using patterns instead of sliding window techniques. Hence, this Chapter investigates approaches solely based on patterns. This contrasts with the previous Chapter which incorporated pattern features into the more standard statistical feature approach.

The Chapter introduces the problem of using only patterns with a simple example to highlight the challenges related to variability and its relation to false positives. As part of this illustration, an adaptive version of the Continuous Wavelet Transform has been developed. Going beyond basic techniques, the Methods Section discusses ways to exploit structure in the data based on unsupervised or supervised data decomposition. The primary contribution of this Chapter is the application of dynamic time warping of activity patterns in the multilinear higher order discriminant analysis (HODA) framework for recognition of activities in a continuous acceleration signal. Experiments on axSpA patients show that this way of taking into account structural information significantly outperforms a more traditional pattern matching approach. Part of the work presented in this Chapter has been published in [26].

7.1 Introduction: a simple step detector

Some basic approaches of pattern matching will be introduced using a simple example: the detection of steps. This is different from merely detecting when a

subject is walking because of the level of required detail. For the example in this study, a public dataset of 17 healthy subjects performing 33 gymnastics activities will be used [16]. They were equipped with nine inertial measurement units (XSens, The Netherlands) sampling at 50Hz. For this study, only the sensor on the left calf has been used. The triaxial acceleration has been transformed to a single amplitude signal. This introduction will build a step detector trained on a single subject and evaluate it on another subject as a simple proof of concept.

7.1.1 Matching via correlation

The most straightforward approach consists of first defining a pattern and then matching it for every possible position using correlation. The pattern has to be learned from the training subject. A step is defined as the basic unit of the walking activity, which is considered distinct from jogging and running. Steps are manually segmented from the walking part and averaged to obtain the template shown in Figure 7.1. Its bimodal character is clearly noticeable.

This template can be slid over and correlated to the continuous signal obtained from the left calf of another subject. Then, the peaks in this correlation signal can be detected via empirical thresholding since they correspond to the steps. Figure 7.2 shows the walking part of the recognition consisting of consecutive steps. The dashed black horizontal line indicates the ground truth, that is, the part labelled as walking in the dataset. As can be observed, most of the steps are detected, although some are missed due to small changes in morphology. After walking the subject proceeds to jogging as can be derived from the larger amplitudes. Since the activity unit shape is different, it is no longer detected.

Improving the detector The top part of Figure 7.3 shows a more extensive result over the first 500 seconds of the recording. Several false positive detections

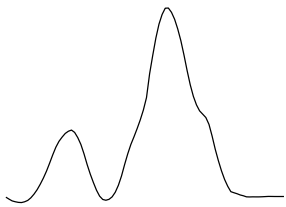


Figure 7.1: The step template, obtained from manual segmentation and averaging of steps on the amplitude of the acceleration of the left calf sensor.

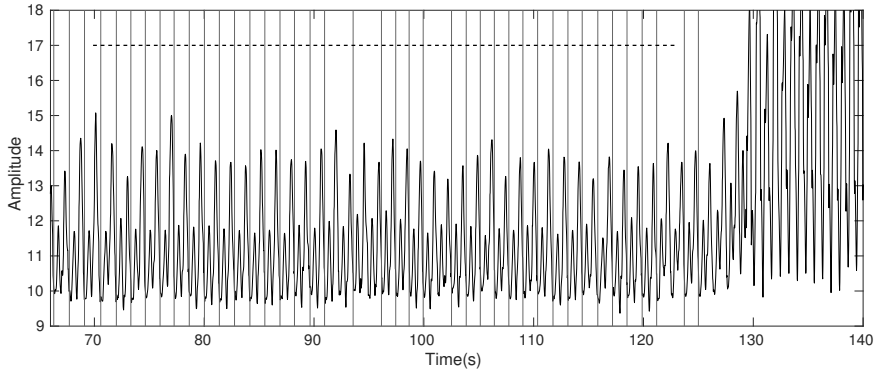


Figure 7.2: Detail of the step detection results showing the acceleration amplitude in black. The detections are indicated by vertical lines. The dashed black horizontal line designates the dataset's walking label indicating the region in which steps should be detected.

can be observed. Some are legitimate since the subject supposedly also walks at other moments, even if this has not been indicated in the labelling. For example, the step detections clearly correctly start before the region labelled as walking. Yet, others differ much more, particularly in amplitude. To solve this, additional information can be incorporated into the detector. For example, the training steps used to build the pattern all have a certain average and variance. These values approximate a normal distribution which can be characterized by its mean and standard deviation. This allows for an additional criterion of detection: given a newly detected step, the probability of its mean and variance belonging to the normal distributions learned from the training data can be obtained. If this probability is too low, the detection fails. The application of this procedure is illustrated in the lower part of Figure 7.3, significantly reducing the number of false positives.

7.1.2 Scalability

The correlation approach works well due to the nature of the recorded activities. They are executed as part of a gym exercise session, in a fixed rhythm and with limited variability. In a daily life setting, the duration of each step might differ. This can be taken into account by noticing the similarity between pattern matching and the continuous wavelet transform (CWT) [103]. Basic pattern matching as used above simply uses a static pattern Ψ. In CWT, Ψ can be scaled with a factor a. This scalable pattern is also called a *mother wavelet*. CWT's procedure of sliding over the signal is implemented via an offset b which

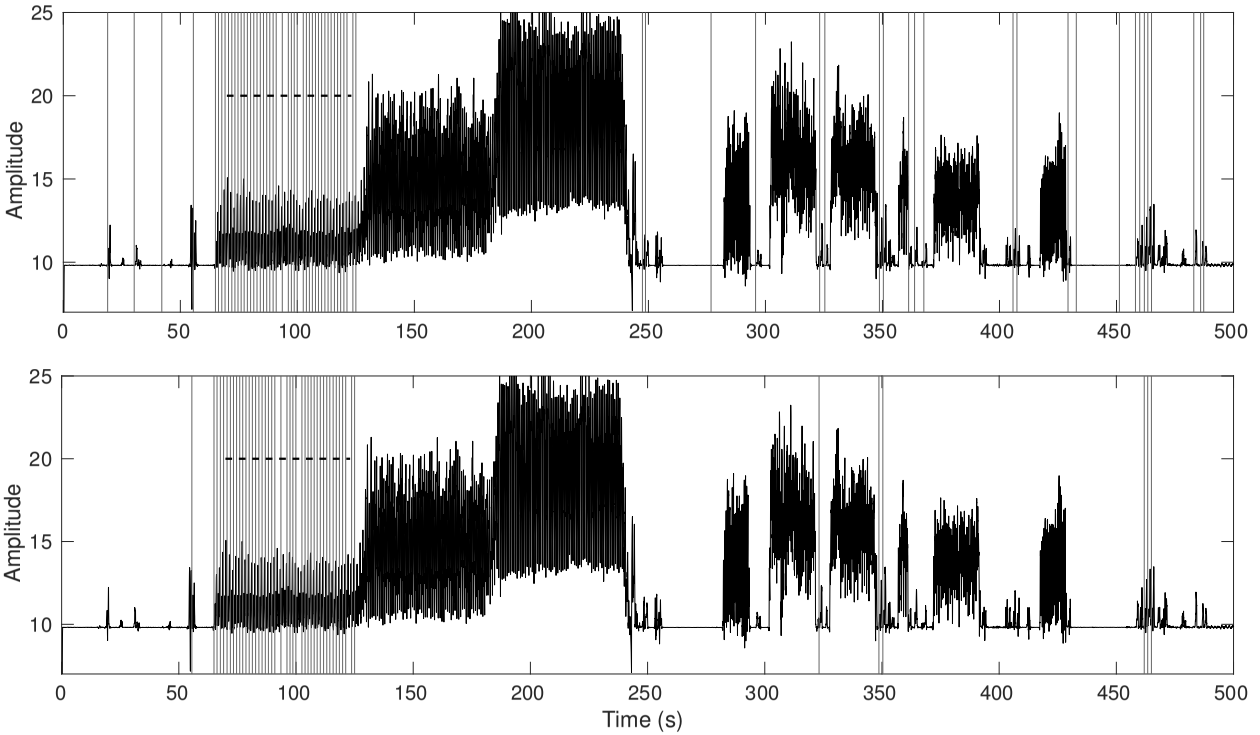


Figure 7.3: Correlation step matching of the first 500 seconds. The top shows the result of pure correlation matching. At the bottom, the mean and variance of the detected pattern have also been taken into account. The acceleration magnitude is shown in black, with the dashed black horizontal line indicating the walking part. Vertical lines indicate step detections.

can take all possible position values. The matching of Ψ to a signal $x(t)$ in CWT therefore becomes a function W with parameters a and b, expressed as:

$$W(b,a) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{a}} \Psi \left(\frac{t-b}{a} \right) dt \tag{7.1}$$

Standard pattern matching could be implemented identically using a fixed value of a, whereas CWT typically features a predefined set of possible values. Although Equation 7.1 features continuous signals, it can be adapted for discrete settings as well. The scaling with a results in a multiresolution matching. In general, the mother wavelet has to fulfil certain requirements e.g. it should have zero average. In practice, a pattern can be approximated to meet these requirements. This approximation is implemented in the `pat2cwav` Matlab function. In case of the step pattern from Figure 7.1, the application of this function yielded a suitable mother wavelet that did not differ in shape from the original.

The impact of scaling is not noticeable in the current example, but the additional flexibility comes at a cost. The middle part of Figure 7.4 shows the result of applying the CWT with the calculated pattern with unity scaling. Although the typical step structure is discernible with an alternative stronger and weaker match, the coefficients for the walking part are lower than for jogging or running. Peak detection on the signal shown in the middle would lead to the inclusion of walking and jogging and, depending on the threshold, exclusion of walking. This can easily be understood via Equation 7.1: the amplitude of the signal $x(t)$ over the matched interval directly affects the wavelet coefficient $W(b,a)$. This can be alleviated as before, by defining an amplitude criterion instead of a single threshold. Alternatively it can be approached by modifying the CWT to use correlation instead of summation (integration): correlation matches only shape rather than the shape and amplitude influence of standard CWT. The result of this adjustment is visualized in the bottom part of Figure 7.4. The impact of amplitude of the signal has been negated. However, due to slight adjustment and scaling of the pattern for normalization, the match is no longer as discriminative as it was when only correlation was used (Figure 7.3). The peak correlation values are hardly higher for walking than for running. The added capacity of dealing with scalability thus comes at the cost of discriminability. This problem only increases when more scale values a are considered.

7.1.3 Limitations

Relatively good results could be obtained in this simple setting although adding flexibility influenced the results. However, the study has some strong limitations.

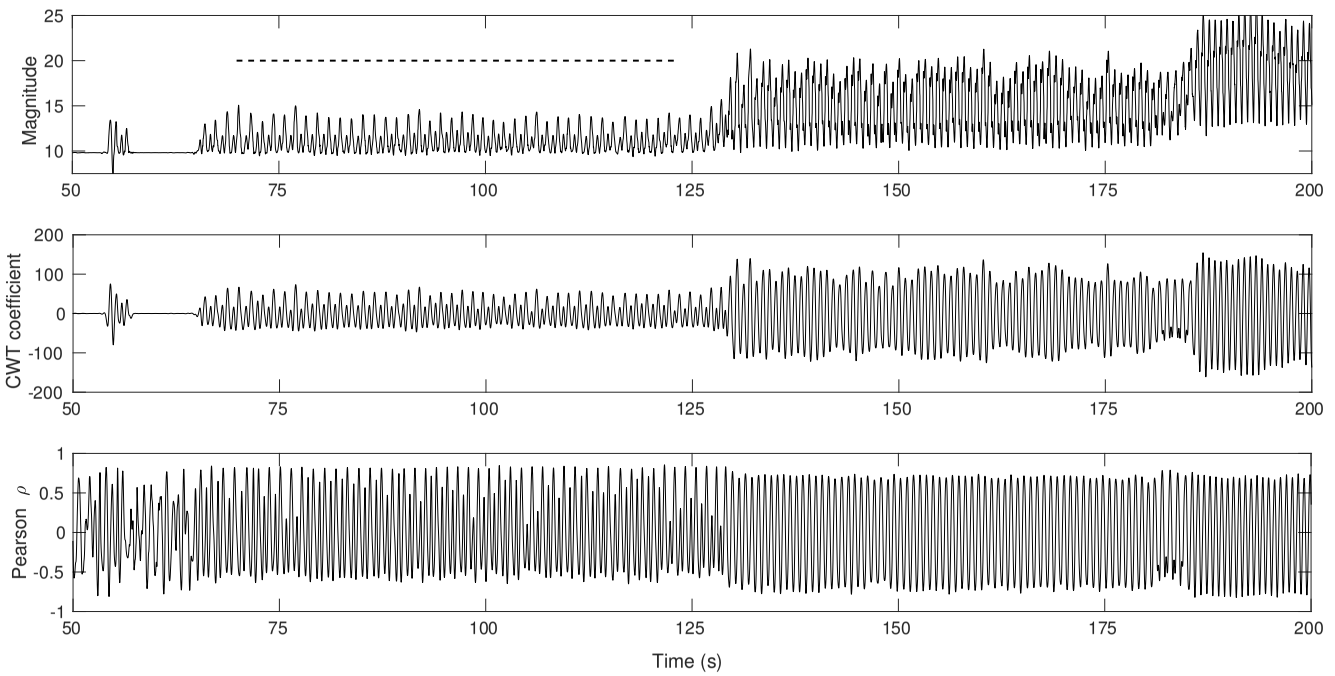


Figure 7.4: Step detection using the Continuous Wavelet Transform. The top part shows the first 200 seconds of the acceleration magnitude with an indication of the walking period via the dashed horizontal line. The second row shows the continuous wavelet coefficients for unity scale. Because it is amplitude based, peakfinding with a threshold would exclude walking and only select running or jogging. The bottom row shows the result of an adapted wavelet transform using Pearson correlation at unity scale. It is only shape based, resulting in slightly larger correlations in the walking part. Due to the scalability, the discriminability has been reduced with respect to Figure 7.3.

1. The approaches rely on threshold values. In case of correlation, the threshold decides which peak to accept and which to reject as detection. The same goes for the CWT approach. This threshold is not absolute, but changes depending on the individual and factors that influence walking e.g. carrying a load. A good value for these thresholds is fundamental for the success of the detection.

2. Even the CWT only allows for uniform scaling of the pattern. This is a strong limitation that is likely to be violated in practice since subjects can be expected to have different walking patterns which vary even within subjects based on e.g. the terrain.

3. All information of the sensor was captured in a single signal whereas three were available. The movement mostly reflects the vertical change in acceleration, but sideways sway could potentially be taken into account as well by using all channels separately. This multichannel matching is also needed when considering multisensor setups.

4. The approaches require computation of every possible match of the pattern. This is akin to a sliding window approach where the window shifts a single sample in each step. This can be avoided by taking larger steps which unfortunately reduces the precision of the match. A hierarchical approach discarding large non-informative regions should be applied instead.

5. The additional flexibility of CWT is intended to match activities executed at multiple scales corresponding to execution speed. However, it will also lead to an increased number of false positives, as can be deduced from the small amplitude difference between different activities in the bottom part of Figure 7.4.

The limitations show that purely correlation-based approaches are not a viable option in realistic dynamic settings. Therefore, this chapter investigates how to couple the benefits of local deformation using Dynamic Time Warping (DTW), discussed in Chapter 5, Section 5.2.3.2, with the information contained in the structure of the data to produce more powerful predictions suitable for use with patients in a real-life setting.

7.2 Advanced methods

7.2.1 Exploiting data structure

The methods discussed so far operate on single data channels. However, if the data has an inherent multiway structure, it is better represented and processed

as a tensor to capture its structural information. Tensors are data constructs in multilinear algebra, an extension of matrices [42]. A tensor has multiple *modes*. For example, a single-mode (or one-way) tensor with N elements is a vector denoted with a bold character $\mathbf{a} \in \mathbb{R}^N$. A dual-mode (two-way) tensor is a matrix, e.g. with N rows and M columns, denoted with a bold capital $\mathbf{A} \in \mathbb{R}^{N \times M}$. Stacking L such matrices yields a tensor $\mathcal{A} \in \mathbb{R}^{N \times M \times L}$, called a three-way tensor or, alternatively, a tensor of *order* three or a tensor with three modes. Such a general tensor is represented by a calligraphic font.

Matrices can be left- or right-multiplied by other matrices or vectors. Right multiplication acts on the column space of a matrix, left multiplication on its row space. For tensors, this can be generalized to mode-n multiplication. Imagine a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and three matrices $\mathbf{A} \in \mathbb{R}^{R_1 \times I_1}$, $\mathbf{B} \in \mathbb{R}^{R_2 \times I_2}$ and $\mathbf{C} \in \mathbb{R}^{R_3 \times I_3}$. In general, mode-n multiplication is indicated by $\times_n$. So, multiplying $\mathcal{X}$ along the first, second and third mode is expressed as $\mathcal{X} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$. This multiplication yields a new tensor of size $R_1 \times R_2 \times R_3$.

Typical examples of tensors include video data with two spatial and one time mode, MRI data with three spatial modes and several constructed tensors such as EEG with time, frequency and channel modes or multilead ECG [87, 138, 68, 156].

If data has a natural multiway structure or can be transformed to a tensorial representation, multilinear algebra allows to decompose it to extract subspaces capturing the most important characteristics of the data. Decompositions can be used for several purposes. For example, they can approximate a tensor by another one with a certain data structure, leading to a simplification of the data. A decomposition furthermore allows to extract several terms, each of which can model a specific aspect of the data e.g. ideally, a term could correspond to an activity class. In general, one could also see it as a projection on a multilinear subspace, where it could potentially be easier to discern between classes.

A tensor decomposition is similar to Singular Value Decomposition (SVD) in matrix algebra. In fact, Multilinear Singular Value Decomposition (MLSVD) exists, but some SVD properties such as best low rank approximation cannot be carried over to the multilinear case. To that end, other decompositions have been developed. They can be carried out in an unsupervised way, without taking into account class information. These methods are of limited use to solve a classification problem. They make the data fit a certain model, but this does not guarantee class separability. Yet, if class information is available, the unsupervised methods can be extended to derive discriminative subspaces via a supervised approach.

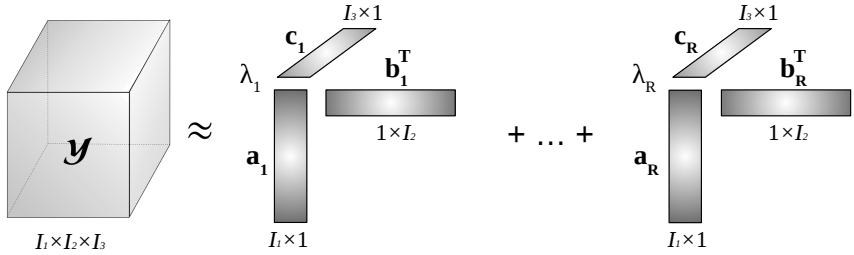


Figure 7.5: An example of the Canonical Polyadic Decomposition of a tensor $\mathcal{Y}$ into R rank one terms built from vectors $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$, $k = 1, \ldots, R$.

7.2.1.1 Unsupervised decomposition

The best low rank approximation of a tensor can be found via the Canonical Polyadic Decomposition (CPD) [42]. The rank of a tensor is defined as the minimum number of rank one tensors used to build it. A N-way rank one tensor is defined as the outer product of N vectors along mode 1 to N. CPD looks for such vectors. Therefore, the approximation of a tensor as rank R can be depicted graphically as in Figure 7.5. A three-way tensor $\mathcal{Y}$ is decomposed in R terms of rank one, in which $\otimes$ denotes the outer (Kronecker) product:

$$\mathcal{Y} \approx \sum_{k=1}^{R} \lambda_k \mathbf{a}_k \otimes \mathbf{b}_k \otimes \mathbf{c}_k \tag{7.2}$$

Note that the vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ need not be orthogonal, but same-mode vectors e.g. $\mathbf{a}_k, \forall\ k$ are linearly independent. Apart from scaling and permutation, the CPD is unique under some mild assumptions.

The CPD vectors contain an approximation of the data available in the full data tensors. In principle, one could concatenate these vectors to a feature vector that could be discriminative between classes. This relies on the assumption that each class has a distinct multilinear structure that emerges in the decomposition. The discriminability of such a feature vector is not guaranteed because it has been constructed in an unsupervised fashion.

A more general unsupervised method is called the Tucker decomposition [42]. It decomposes the tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ into a smaller tensor, called the *core tensor*, and *factor matrices*. The principle is illustrated in Figure 7.6. The core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is multiplied by matrices $\mathbf{A} \in \mathbb{R}^{I_1 \times R_1}$, $\mathbf{B} \in \mathbb{R}^{I_2 \times R_2}$ and $\mathbf{C} \in \mathbb{R}^{I_3 \times R_3}$ along the first, second and third mode, respectively. Summarizing,

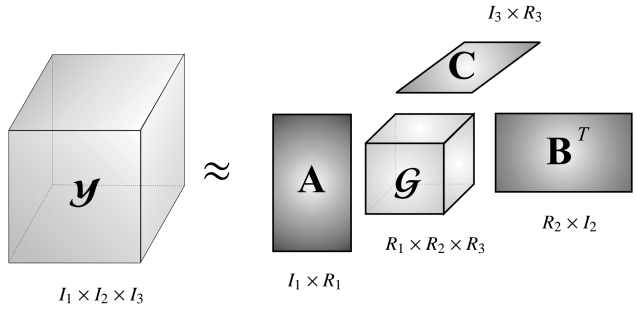


Figure 7.6: An example of a Tucker decomposition [128]. A tensor $\mathcal{Y}$ can be decomposed as a core tensor $\mathcal{G}$ and factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, one for each mode.

the Tucker decomposition in Figure 7.6 can be written as:

$$\mathcal{Y} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \tag{7.3}$$

The CPD for rank R is a special case of the Tucker decomposition. Its factor matrices are constructed from the vectors $\mathbf{a}_k$, $\mathbf{b}_k$, $\mathbf{c}_k$, $k = 1, \ldots, R$ in Equation 7.2. Its core tensor $\mathcal{G} \in \mathbb{R}^{R \times R \times R}$ is all-diagonal, the only non-zero entries being the λ_k values on its main diagonal.

Equation 7.3 is a decomposition because it expresses the content of a tensor as a combination of interactions of vectors, namely, the columns of the factor matrices. One could use the factor matrices of each decomposed data tensor for features, as with the CPD. However, a classification problem can also be solved as follows. If common factor matrices can be found for the observation tensors, their core tensors all express the full content of the original tensors in the multilinear space spanned by the factor matrices. The elements of the core tensors are then used as discriminative features. In the unsupervised setting data is merely compressed and decomposed along the modes. This yields insight into its trends in different modes, but it does not guarantee good discriminative properties. Discovering common factor matrices in a class-informed way, hence supervised, is the goal of the algorithm called HODA, discussed next.

7.2.1.2 Supervised decomposition: Higher Order Discriminant Analysis

HODA aims to construct common factor matrices by jointly decomposing a set of N-way tensors. This can be achieved with an explicit joint formulation, but the more common and elegant way is the formulation in terms of a decomposition of a single tensor of order $N + 1$. That is, it has been shown that stacking

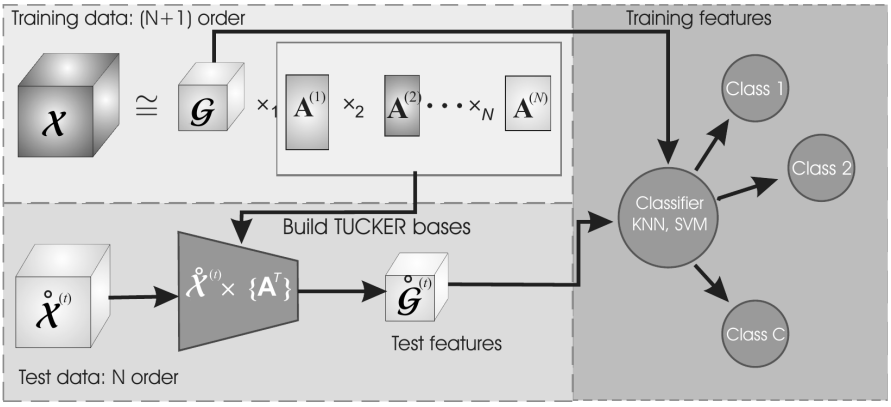


Figure 7.7: Graphical summary of the HODA procedure. An $(N+1)$-way tensor is obtained by stacking the N-way training tensors. The orthogonal Tucker decomposition maximizing Fisher's ratio yields the factor matrices $\mathbf{A}$ and core N-way core tensors $\mathcal{G}^{(k)}$ vectorized as training features. Applying the factor matrices on test data similarly yields the test features [128].

all tensors along an additional mode and performing a Tucker decomposition disregarding the factor matrix of this last mode is equivalent to the joint formulation [129]. Such a decomposition is in general not unique. However, uniqueness can be enforced by constraints such as orthogonality or nonnegativity. HODA finds common orthogonal factor matrices via an iterative procedure attempting to maximize Fisher's ratio [129] based on the class labels. Several implementations are available for this optimization. In our work, the approach based on generalized eigenvalue decomposition is used. More details can be found in [129].

Once the common factor matrices have been found, the individual core tensors resulting from the original tensor observations can be vectorized and serve as training features. Moreover, the factor matrices can be used on new observations, e.g. test data, to find the core tensor and hence test features. These features can subsequently be used with any classification approach. The full HODA procedure is presented graphically in Figure 7.7. The graphical depiction is limited in its possibility to express N-way tensors for $N > 3$. Therefore, all higher-order tensors are represented as cubes. Note that, as any decomposition, HODA yields an approximation of the original tensors, visible through the 'nearly equal' sign in the training part of the Figure. The quality of the approximation is based on the number of columns of the factor matrices, which is often limited via a predefined quality threshold indicating how good the approximation has to be.

7.2.2 Algorithms and experiments

This Section introduces this Chapter's main contribution as an application for the exploitation of structural information on top of dynamic time warping. It refers to the datasets and explains the novel data representation and the feature construction. Since features are constructed from segments, it also discusses the segmentation approach. Finally, it gives an overview of the experiments and how they will be evaluated.

7.2.2.1 Datasets

For these experiments, the SenseWear and Shimmer protocol datasets discussed in Chapter 5, Section 5.5 were pooled to increase the population size and to study the impact of heterogeneous conditions. This yields a total of 39 patients. For the Shimmer patients, where multiple sensors are available, the sensor on the upper arm was used for correspondence with the SenseWear setup. Moreover, the third axis was discarded to further minimize the differences between the setups.

Recall that the protocol consists of the patients performing a series of physical tasks based on the BASFI questionnaire. An earlier study on the test-retest (between-trial) validity of the activities on the SenseWear dataset lead to a subset of only six of the tests [159]. Table 7.1 presents a summary of the activities. As can be seen, half of the selected activities are repeated five times since this reduces the test-retest variability. In the Shimmer recording, only four out of six selected activities were performed, liedown and maxreach were not present. Nevertheless, all six activities are included in the experiment to allow for a wider variability. Table 7.1 indicates the number of patients that performed the activity in its last column.

7.2.2.2 Data representation

The data can be represented in several ways. Firstly, DTW can match each observation to class patterns. This allows to extract simple features as in Chapter 6. The procedure is repeated in detail below, listing the specific features used for the experiment. Alternatively, the data segments, matched to class patterns, can be incorporated in a tensor. Figure 7.8 summarizes this approach. Afterwards, these tensors can be used to extract HODA features. Throughout, data are assumed to be available as annotated segments, rather than as a continuous acceleration data stream. How to derive these segments will be discussed after the features have been introduced since the segmentation with rejection is closely tied to the recognition.

Table 7.1: Overview of the six activities performed in the experimental protocol. The last column shows how many patients performed the activity. liedown and maxreach do not occur in the Shimmer dataset, leading to a lower number of patients for these activities.

Activity	**# patients**
getup	39
liedown	28
maxreach	28
pen5	39
reach5	39
sts5	39

Pattern definition with Dynamic Time Warping

A standard DTW implementation matches two signals. In order to use it to construct a pattern for each activity in the current setup, the same two DTW extensions as discussed in the previous Chapter are needed to allow for multichannel and multi-trial matching. Hence, if training data has been selected, six patterns can be derived by applying DTW once for all segments belonging to the same activity class. Each pattern consists of two pattern channels. This is visualized in the left part of Figure 7.8.

Simple pattern features

Simpler pattern matching features are proposed for comparison to assess the added value of the tensors. They are extracted at the layer of the deformed segments in the Figure. At that layer, training or test segments are matched against the patterns. For each match, DTW provides an Euclidean distance measure to assess the similarity between the warped segment and the template. Overall, this yields a distance for every activity class pattern per segment. Hence, these distances can be used directly as feature vector. Starting from the derived activity patterns, the approach can be summarized as follows, both for training and test segments:

1. Match the segment to an activity pattern using DTW. This is a simple match between the two-channel segment and a two-channel pattern. Its result is a two-channel deformed segment and a distance score.

2. Repeat the first step for all activity patterns.

It is possible to directly derive the class label from the feature vector by a simple approach: the minimal distance to a pattern most probably points to the associated class as label. Yet, the previous Chapter showed that all distances

together yield more information than only this minimum. Therefore, the full feature vector will be used as benchmark for further analysis.

Tensor construction from activity patterns

Starting from the activity patterns defined earlier, a tensorial representation for any activity segment is constructed in four steps. This is visualized in the right part of Figure 7.8.

1. Match the segment to an activity pattern using DTW. This is a simple match between the two-channel segment and a two-channel pattern. Its result is a two-channel deformed segment.
2. Repeat the first step for all activity patterns.
3. Resample all deformed segments to a common length. A length of 150 samples was empirically selected for this study.
4. Stack all resampled deformed segments into a time $\times$ channel $\times$ activity tensor. Here, this yields a $150 \times 2 \times 6$ tensor.

The reasoning behind the approach is as follows. If segments represent a certain activity, the match between DTW and the pattern will be good and the deformed signal will still resemble the pattern. On the other hand, deformations to match other activity patterns will be more random. Hence, the constructed tensor contains information about how well a segment resembles each of the activities. This is also captured in the simple DTW features mentioned before, but they compress information into a single variable per activity, whereas the tensor approach preserves more localized similarity information.

HODA Features

The tensors that have been constructed are used as input for classification. To this end, features are extracted from them using Higher Order Discriminant Analysis (HODA) features as explained in Section 7.2.1.2. The implementation used here is provided in the Novel tensor toolbox for Feature Extraction and Applications (NFEA) [128] which requires the use of the tensor toolbox by Kolda and Bader [15].

7.2.2.3 Detection and Recognition

So far, the focus has been on the building blocks for activity recognition starting from accelerometer segments. Yet, this first requires detecting these segments in continuous data streams. The process consists of two steps. Firstly, potential segments of interest are detected. This process is independent of any features used for further stages. Secondly, non-informative segments should be discarded.

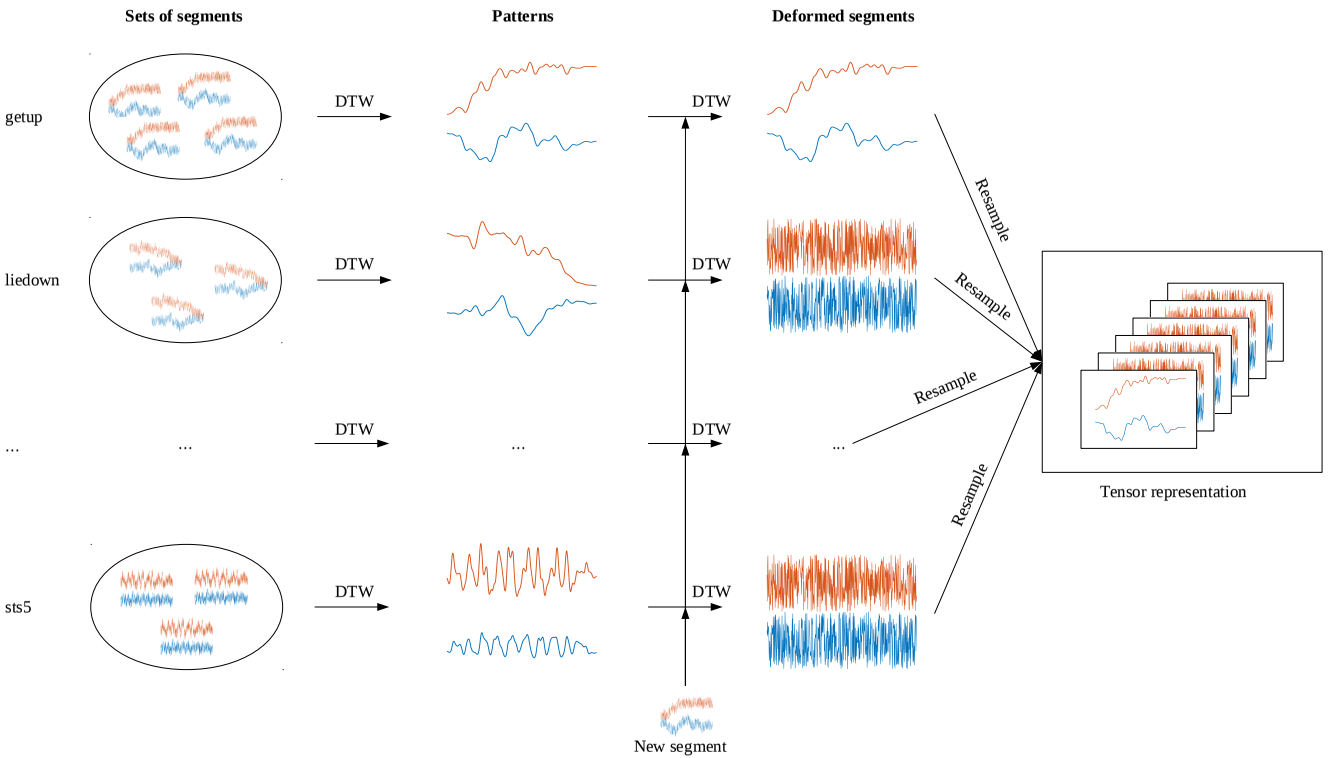


Figure 7.8: Overview of the data representation. First, DTW is used on sets of activity training segments to create activity patterns for all activities mentioned in Table 7.1. Next, a new data segment is matched on these patterns with DTW. Finally, the deformed representations are resampled and grouped in a tensor as a new multiway representation of the data segment. In the ideal case depicted here, the match is perfect for the correct class and random noise for the other classes.

Segmentation

Segments of interest are identified by looking at the signal variability. The dynamic regions of the signal are subsequently separated from the more static ones. This separation was implemented as a step-wise adaptive approach. Essentially, the approach is the same as described in Chapter 6, Section 6.1.3 but some aspects have been changed to fit this study.

1. The continuous data is filtered with a low-pass butterworth filter of the fourth order, with a cut-off frequency of 1.6Hz. This corresponds to 10% of the signal bandwidth. To judge the general movement pattern, low frequencies are the most important ones.

2. A rough segmentation splits the signal into windows of two seconds, with 50% overlap. Segments are marked as dynamic based on their standard deviation and range, compared to empirical thresholds obtained from preliminary analysis of the training data.

3. Refinement of the dynamic regions is achieved by shrinking or extending the static regions in between dynamic segments based with half a second. The decision is based on the difference in variance between half-second regions and is identical for the start and end of a region, so the discussion will be limited to the start. The initial second of a static region serves as baseline. Extension is accepted if the second starting at half a second before the current start has a variance which is maximally 10% higher than the baseline. This tries to grow the static region without incorporating too much movement. Shrinking is accepted if the variance of the second starting at half a second later than the current start is at least 10% lower. This tries to eliminate movement at the start of the region.

4. The above procedure is carried out independently for different channels. The detected dynamic regions are subsequently joined over all channels. Static gaps of less than a second in between dynamic regions are discarded if the means of the regions are similar. As a last step, dynamic regions of less than two seconds are discarded.

Rejection

Rejection involves discarding segments representing activities that are of no interest in this study. It is challenging because the rejection class is heterogeneous and cannot always be fully known beforehand since patients can perform any movement, whereas for the activity classes, their general characteristics are known. A possible approach is the use of a one-class classifier for each class of interest but this has not been applied for this study since in this case, an estimate of the segments to be discarded is available from training data.
For every patient, the informative segments are known in the continuous dual-channel acceleration signals. Hence, if the segmentation is run, all segments not

overlapping with the known activities are examples of the rejection class. In other words, this allows to increase the set of known segments and labels for each patient by introducing a rejection class label. Therefore, the problem is simply solved as a classical multi-class approach. A random forest with 250 trees was selected for its ability to model complex boundaries [70]. Because the rejection class is on average six times larger than all other classes combined, a uniform prior was imposed to avoid too strong a bias towards it. Individual trees were constructed by random resampling of the data with an in-bag-fraction of 0.65. Other than that, standard settings in Matlab's `TreeBagger` have been used.

7.2.2.4 Overview of the experiments

All experiments were performed in Matlab R2017b (The Mathworks). Three aspects of the approach were tested. A first experiment solely focuses on recognition, that is, it starts from already segmented data. The rejection class is not considered. A second experiment assesses the segmentation approach, neglecting any class labels. Finally, the third experiment brings together segmentation and classification. The first and the third experiment include a comparison between the methods applied on simple features or HODA features. The statistical significance of the results of this comparison is assessed with a sign test, a non-parametric test for the difference of medians of two populations, implemented by Matlab's `signtest`.

Recognition

To evaluate recognition only, a 'closed world' scenario is used. All activities are available as segments and all available segments can only be one of the six classes listed in Table 7.1. Data is processed as leave-one-subject-out (LoSo). The manual class identifications are used as ground truth labels.
Simple and HODA features are calculated as described in Section 7.2.2.2. Summarizing, in every fold, the following steps are carried out:

1. Class-specific patterns are derived from the training data using DTW.
2. Data segments, both training and test, are warped to all training class patterns. This yields deformed segments with the associated distances as *simple features*.
3. The deformed segments are converted to a tensorial representation.
4. HODA derives the discriminative subspaces and core tensors based on the training data. The toolbox parameters for the quality of the approximation is set to 98.5.

5. The subspaces defined by the factor matrices are used to extract the core tensors for the test data.

6. Training and test *HODA features* are obtained by vectorization of the core tensors.

7. Test data is evaluated via a multiclass-trained linear discriminant analysis classifier (LDA) for both types of features separately [74].

In the closed world approach the segments are expected to be relatively dissimilar since they correspond to different classes. Hence, a more complex classifier such as the random forest mentioned earlier to deal with the heterogeneous rejection class is not necessary here. In contrast, it might lead to overfitting. LDA is a simple linear classifier with less hyperparameters and was introduced in Section 5.4.2. Due to its linear nature, it leads to more robust models. Results will be assessed using accuracy per patient as well as a confusion matrix over all patients.

Segmentation

Segmentation is performed for every patient as outlined in Section 7.2.2.3. It is expected that many segments will be found, among which also the informative ones given as training data. This experiment matches the detected segments with the given ground truth. Detected segments that do not overlap with the ground truth are discarded. Aspects to be assessed include whether all informative segments have indeed been detected and how accurate this detection is. In Chapter 5, Section 5.4.2 the SDC was introduced for this purpose. Note that segmentation is independent of the type of features used later for recognition. Therefore, no comparison is provided.

Recognition and segmentation

The last experiment is similar to the first as it also runs with the LoSo setup and compares simple features to HODA features. However, as described earlier, a rejection class is introduced. Another difference is that the test data segments are no longer predefined through manual delineation, but are obtained through automatic segmentation, as in the previous experiment. The aim is to reject non-informative segments and correctly classify the others. Several measures will be used to assess the performance: the false detections (FD), detection true postive rate (DTPR), pure accuracy Acc_p and actual accuracy Acc_a. The measures were discussed in detail in Chapter 5, Section 5.4.2.

7.3 Results

This section subsequently discusses the results of the three experiments: Recognition, Segmentation and their combined application.

7.3.1 Recognition

Figure 7.9 shows a bar plot of the classification accuracy for all patients using HODA features. The average accuracy is 94.34% (std 9.24%). Moreover, the figure shows that perfect recognition was achieved for 25 patients. A further four patients had accuracies higher than 90%, six others between 80% and 90% and three between 70% and 80%. Only a single patient had an accuracy lower than 70%.

The boxplot in Figure 7.10 offers a comparison between the simple features and HODA. The latter outperforms the former with a highly significant difference ($p < 0.01$). Simple features lead to an average classification accuracy of 78.21% (std 17.74), a difference of 16%. Only 8 patients achieve perfect recognition using simple features.

Assessment can also be done on the level of confusion between activities. The conclusions will be illustrated for HODA, but are similar for the simple features. Table 7.2 displays the confusion matrix for all activities. It shows that all occurrences of *liedown* and *maxreach* are perfectly recognized. Moreover, they have no confusion with other activities at all. *reach5* and *getup* are only missed twice. Another observation is some confusion between the repeated activities. Most importantly, *pen5* is sometimes estimated to be *sts5* (five occurrences) or *reach5* (five occurrences). This can be explained by the fact that the movement of the arm is similar in these cases.

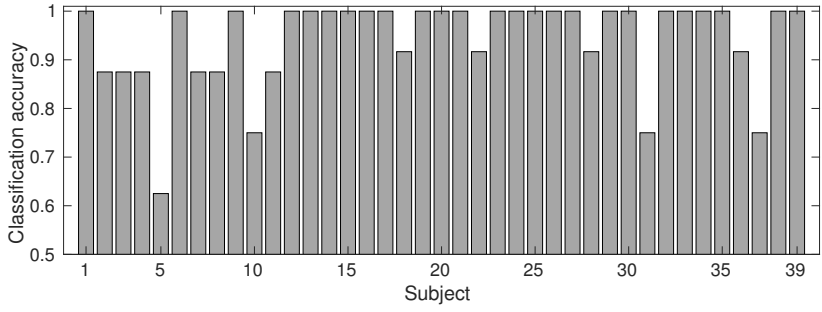


Figure 7.9: Classification accuracy for each patient using HODA features.

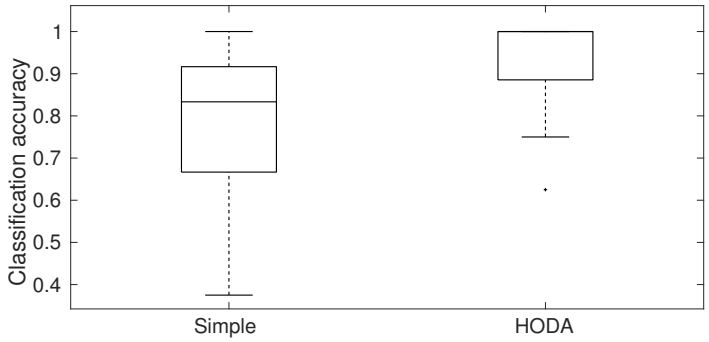


Figure 7.10: A comparison of the classification accuracy using LDA on simple and HODA features.

7.3.2 Segmentation

The results of the segmentation of all patients can be seen in Figure 7.11. All informative segments were detected. Although several detections cover only a small part, the majority of the detections is accurate as indicated by an overall average SDC of 0.8932 (std 0.1202). However, the distribution is highly skewed since 94.81% of the segments obtained an SDC higher than 0.7.

The SDC's calculation only takes into account correctly identified segments, that is, overlapping with the given segments for informative activities. However, non-informative segments have been detected as well, segments where activity occurs that should be rejected. On average 55.61 additional segments were found, with a maximum of 87. The next section discusses how well all of these can be discarded whilst still retaining the informative segments.

Table 7.2: Confusion matrix of the classified activities using HODA features pooled over all 39 patients.

		Predicted labels					
		getup	liedown	maxreach	pen5	reach5	sts5
Actual labels	getup	76	0	0	0	1	1
	liedown	0	56	0	0	0	0
	maxreach	0	0	56	0	0	0
	pen5	0	0	0	68	5	5
	reach5	0	0	0	1	76	1
	sts5	2	0	0	3	2	71

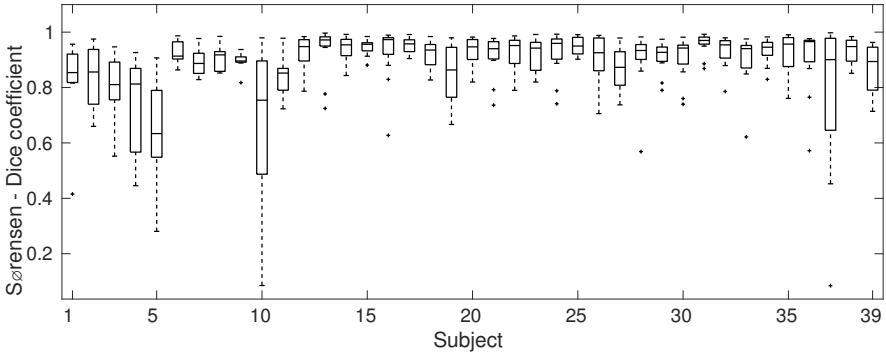


Figure 7.11: Segmentation performance for all patients expressed as boxplots of individual Sørensen-Dice Coefficients (SDC) of the segments. All informative segments were detected.

7.3.3 Recognition and segmentation

The number of false detections (FD), the detection true positive rate (DTPR) and the pure and actual accuracies (Acc_p and Acc_a) are given in Table 7.3 for both HODA and simple features. They were defined in Chapter 5, Section 5.4.2.

Compared to the Segmentation results in Section 7.3.2, the number of false detections is decreased by applying the rejection included in the random forest. From the average FD of 55.61, for HODA, only 2.90 remain, with a maximum of 16. Ten patients correctly rejected all spurious activities. In contrast, the simple feature approach still yields an average of 7.90 false detections, with a maximum of 21 and only for a single patient, all false detections were rejected. Hence, HODA features are more powerful than simple features for rejection.

Rejection is a trade-off, balanced with the amount of correct detections. For HODA, on average, 85.36% of the informative segments are detected, but large differences between individuals occur. In 14 patients, all segments are detected, but for four patients less than 60% of the segments are found. Using the simple features, 82.16% of the informative segments are detected. The sign test indicates a significant difference in the DTPR ($p = 0.0436$).

The last two columns of Table 7.3 bring everything together. As a reminder, Acc_p is equivalent to the performance for the closed world classification experiment performed in this study, but the detected segment boundaries are used, rather than the ideal ones. Also, rejected segments do not contribute to this accuracy. As a consequence, for HODA, classification accuracy drops on average from 94.34% (closed world) to 86.72% Acc_p. Patient one only recognizes half of

Table 7.3: Results for the combination of segmentation and classification expressed with the number of false detections (FD), the Detection True Positive Rate (DTPR), the pure accuracy (Acc_p) and the actual accuracy (Acc_a) using HODA and simple features. The first 11 subjects belong to the Shimmer dataset, the last 28 to the SenseWear set.

	HODA				**Simple**			
Patient	**FD**	**DTPR**	**Acc$_p$** (%)	**Acc$_a$** (%)	**FD**	**DTPR**	**Acc$_p$** (%)	**Acc$_a$** (%)
1	3	0.5	50	18.18	9	0.25	50	5.88
2	16	0.88	71.43	20.83	15	0.63	60	13.04
3	8	0.88	85.71	37.5	12	0.88	100	35
4	5	0.63	80	30.77	7	0.63	80	26.67
5	6	0.75	33.33	14.29	12	0.5	0	0
6	10	0.88	85.71	33.33	19	0.86	71.43	18.52
7	3	1	75	54.55	8	0.75	100	37.5
8	5	0.75	66.67	30.77	6	1	87.5	50
9	5	0.75	100	46.15	15	0.75	83.33	21.74
10	4	0.5	100	33.33	13	0.63	80	19.05
11	7	0.63	80	26.67	10	1	37.5	16.67
12	3	0.92	81.82	60	10	0.75	100	40.91
13	0	0.5	100	50	7	0.92	81.82	47.37
14	2	0.83	90	64.29	7	0.83	100	52.63
15	2	1	75	64.29	6	0.83	100	55.56
16	3	0.83	80	53.33	5	0.83	100	58.82
17	2	0.92	90.91	71.43	15	0.92	81.82	33.33
18	0	1	91.67	91.67	2	1	83.33	71.43
19	0	1	91.67	91.67	6	0.83	100	55.56
20	1	1	100	92.31	4	0.92	81.82	56.25

Continues on next page.

Table 7.3: Continued from previous page.

	HODA				Simple			
Patient	**FD**	**DTPR**	**Acc$_p$** (%)	**Acc$_a$** (%)	**FD**	**DTPR**	**Acc$_p$** (%)	**Acc$_a$** (%)
21	0	1	83.33	83.33	1	0.83	100	76.92
22	6	0.83	90	50	21	0.67	87.5	21.21
23	2	1	83.33	71.43	6	0.75	100	50
24	0	0.83	100	83.33	0	0.75	100	75
25	0	0.83	100	83.33	4	1	83.33	62.5
26	2	1	91.67	78.57	4	1	75	56.25
27	0	1	100	100	1	1	91.67	84.62
28	1	0.92	81.82	69.23	10	0.75	55.56	22.73
29	0	0.92	100	91.67	6	0.83	100	55.56
30	0	1	100	100	3	1	91.67	73.33
31	2	0.67	100	57.14	6	0.75	100	50
32	2	1	91.67	78.57	9	0.92	90.91	47.62
33	2	0.92	90.91	71.43	7	0.83	90	47.37
34	1	0.83	100	76.92	8	1	91.67	55
35	2	1	83.33	71.43	5	0.83	70	41.18
36	1	0.83	80	61.54	8	0.92	81.82	45
37	6	0.58	85.71	33.33	10	0.58	71.43	22.73
38	0	1	100	100	8	0.92	90.91	50
39	1	1	91.67	84.62	3	1	83.33	66.67
AVG	2.90	0.85	86.72	62.34	7.90	0.82	82.91	44.09
STD	3.32	0.16	14.21	25.03	4.74	0.16	20.37	20.71

the activities and patient five only one third of them. In contrast, 12 patients recognize all detected activities perfectly and nine patients recognize more than or equal to 90% correctly. The result for the simple features improves from 78.21% to 82.91%. This signifies that the segments that were rejected based on the simple features in the combined experiment could not be recognized easily and were wrongly classified in the closed world experiment. In other words, misclassifications in the closed world experiment were traded for rejections in the combined experiment. Despite the Acc_p increase, HODA outperforms the simple features significantly ($p = 0.0326$).

The last column (Acc_a) of Table 7.3 shows the combined effects of the previous three columns and represent the complete performance of the system. For HODA, it drops with 24% compared to Acc_p to an average of 62.34%. As a general system, this is insufficient. Yet, for some patients, the performance is much better. The low performance is due to several badly-predicted patients. For 13 patients an actual accuracy below or equal to 50% is obtained. On the other side of the scale, for 11 patients Acc_a is higher than 70% and for four others higher than 90%. For three patients perfect detection and recognition is achieved, yielding a global performance of 100%. The performance difference with the simple features is immediately apparent and highly significant ($p < 0.01$). The large difference might seem surprising given the previous two columns with much smaller differences, but it can be explained by the high number of false detections. As an example, patient 38 has 12 target occurrences of activities. The simple features approach detects 11 of them. Ten out of these 11 are classified correctly. However, eight false positives are considered as wrongly classified, leading to an actual accuracy of 50%. This connects to the earlier remarks on the superiority of the HODA features over the simple features for rejection.

Performance can also be grouped by activity rather than by patient. Table 7.4

Table 7.4: Performance for the combined segmentation and classification, grouped by activity, expressed as the Detection True Positive Rate (DTPR), rejection rate (Rej), the pure accuracy (Acc_p) and the actual accuracy (Acc_a) using HODA features.

	DTPR	**Rej**	**Acc**$_p$ (%)	**Acc**$_a$ (%)
getup	0.91	0.05	95.77	87.18
liedown	0.88	0.07	100	87.5
maxreach	1	0	100	100
pen5	0.87	0.12	79.41	69.23
reach5	0.74	0.23	93.10	69.23
sts5	0.83	0.08	66.15	55.13

gives an overview for the HODA features. The same trends as in the confusion matrix in Table 7.2 can be observed. The non-repetitive activities perform better. This might be due to that repetition is closely linked to often-occurring activities such as walking. This hypothesis is strengthened by considering the rejection percentage. For this measure, activities have been segmented, but are predicted to be in the rejection class. The rejection rate of repetitive activities is higher, particularly for reach5. Overall though, the problem is mostly with sts5 which has an actual recognition performance of only 55.13%.

Looking more in detail at the results in Table 7.3 reveals a recurring issue. The data description mentioned that two datasets were merged to increase population size and strengthen the conclusions. In the analyses, the first 11 patients belong to the most recent data acquisition with the Shimmer, whereas the last 28 are recorded with the Sensewear Armband. In all recognition experiments, the Sensewear set outperforms the Shimmer, both with the HODA and the simple features. A summary of the most important results of all three experiments is provided in Table 7.5. In the closed world scenario, the HODA performance difference between the datasets is 10%. With the simple features, both datasets perform worse and the difference between them increases further to 20%. The results lead to the conclusion that HODA, apart from performing better, also succeeds in shrinking the performance gap between the two datasets. Also, in segmentation, the difference between the datasets is noticeable. The 11 Shimmer patients seem to have been segmented less accurately. This might be because the segmentation becomes more difficult if the activities under consideration are executed in a shorter time span without interruption or with immediate displacements afterwards. Ideally, the signal would be static in between activities and the activities themselves would be executed fluidly, without pauses. In (emulated) real-life settings, these ideal conditions are not always fulfilled. Finally, in the overall performance measure Acc_a, the gap grows to more than 40% for HODA due to worse rejection and recognition in the Shimmer dataset. For simple features, the gap is smaller (30%), but in the Shimmer dataset, only 22.19% of the data is classified correctly.

Table 7.5: A summary of the performance difference between SenseWear (28 patients) and Shimmer (11 patients) datasets for the three experiments in terms of, respectively, average closed world experiment accuracy (Acc), average segmentation quality (SDC) and average actual accuracy (Acc_a).

	HODA			**Simple**		
	Acc (%)	**SDC**	**Acc**$_a$ (%)	**Acc** (%)	**SDC**	**Acc**$_a$ (%)
SenseWear	97.02	0.91	74.46	83.93	0.91	52.70
Shimmer	87.50	0.81	31.49	63.64	0.81	22.19

7.4 Discussion

The results show that the newly proposed activity recognition approach with HODA outperforms a more standard direct comparison based on the distances derived from Dynamic Time Warping, both in a closed world scenario and with the inclusion of false and missed detections. The significant improvement is due to taking into account the entire signals, not only the reduced similarity information. Of course, it should be noted that the final feature space with HODA offers more degrees of freedom for the classifier since it is generally of higher dimensionality, around 30. Nevertheless, this is not only an advantage because it also increases the danger of overfitting.

It is also apparent that large differences between patients exist. As an example, Table 7.3 shows performances ranging from 18% to 100%. This shows that the effectiveness of the model greatly depends on the patient. In this study, training was performed in a leave-one-subject-out approach. Hence, a patient-independent model was produced. As an alternative, data from the patient to be tested could either be part of the training set or only patient-specific data could be used. A model constructed in this way is expected to perform better for a specific patient, but it is less generally applicable. Therefore, looking into adaptive methods might be beneficial. Such approaches start from a patient-independent model that is updated when sufficient specific data is available. Of course, it has the drawback of the need for patient-reported labelling during the update period [48].

Apart from between patients, the difference between datasets is also apparent, as explained before. There are several explanations: more activities are taken from the Sensewear set and segmentation seems more successful due to more standardized behaviour, e.g. pauses in between activities. A more important issue is likely the sensor itself. The Sensewear is an armband with a built-in strap, meaning that its positioning with respect to the arm is more likely to be similar between patients. The Shimmer has to be attached manually. Moreover, it contains three axes instead of two. Although care has been taken to attach the sensor and select the axes corresponding to the Sensewear, no further correction other than normalization with respect to the magnitude of gravity was applied to harmonize the two datasets. Many calibration parameters could introduce small differences. The performance difference is hence likely caused by three effects: differences within the Shimmer dataset are larger due to sensor setup, there are differences between the datasets and the Sensewear data has more influence in the training process due to its higher number of patients. In further studies, it would therefore be advised to use the same sensor for all recordings. This did not happen here because the Sensewear was no longer available. Also, a future study could compare the sensors more explicitly, which was not the aim here. In this study, data from both sets contribute to the classifier and

influence the results. In itself, this shows that HODA can obtain good results and is more robust against the heterogeneity than simple DTW features. Yet, performing the identical study on two separate sets of the same size with the different sensors might further clarify the impact of the sensor type.

Another factor influencing the performance is the manual labelling of the data. It is not exhaustive, so some false detections might actually be correct. These detections have not been checked manually afterwards. Moreover, in this case, correct segments would have ended up in the set of examples for rejection, leading to a less discriminative choice of features.

The most important drawback of the algorithm is its high computational cost. If more activities would be considered, it would be even higher. Training can be done in advance, but testing alone is also costly. In its current state, it is not suitable for real-time use. Structuring the data and applying HODA is not computationally expensive. The repeated application of DTW proves to be the biggest concern. It could be alleviated by using dedicated hardware e.g. based on memristors [178]. However, in the context of assessment of functional capacity, computational cost is not an issue. Current practice consists of a regular BASFI assessment. Disease evolution over time could be tracked in more detail by a daily automatic assessment, which can be reported to the clinician when available. This could detect fluctuations during the week, but also between meetings. Real-time recognition could be useful e.g. to detect and correct maladaptive poses through direct feedback, which is another possible aim. In that case, another approach is preferable.

The Chapter addresses the general problem of the limited applicability of pattern detection compared to sliding window approaches. This is achieved through the approach to rejection, first looking for dynamic segments and then rejecting non-informative segments as rejection class. The reason for rejection can be three-fold. Firstly, the rejection class serves as a model, but not in the same way as the activity classes. The tensorial data representation encodes similarity to the class patterns. No pattern is extracted from the rejection class because of its heterogeneity. However, HODA decides on discriminative features using the full seven class problem, including the rejection class. In other words, the random forest model part representing the rejection class is build on rejection data represented through its similarity with the classes. With this model, if similar non-informative segments are segmented, they can be rejected. Secondly, some segments are only partly segmented, which diminishes their resemblance to the class pattern. Hence, they are rejected by the final classifier. Thirdly, some activities are difficult to recognize, even when ideally segmented. This might lead to their features resembling the rejection features rather than their own class. This shows that, through segmentation, the continuous problem can be converted into a discrete one. Subsequently, the same approach as with sliding windows can be implemented, making the rejection class part of a

multi-class problem. The current approach has the advantage to explicitly aim for meaningful segments, whereas this is not guaranteed with basic windowing.

Of course, the choice of focusing on patterns alone also influences the performance. The methods employed in this study heavily rely on pattern matching, based on the hypothesis that the acceleration pattern captures the entire activity. A part of the variability in the data is captured through DTW and HODA-derived subspaces. Nevertheless, patterns also contain inherent noise in the sense that some parts of the movement are not defining aspects of a particular activity. Current results are promising, but, at first glance, seem inferior to the results obtained in the previous Chapter through the combination of statistical and simple pattern features. To verify this hypothesis, the results of the SenseWear patients should be looked at. As mentioned, performance is higher for this subgroup. Note that the effects of the other 11 patients is not completely negated since they were part of the cross-validation training data. Yet, it allows for a tentative comparison. A Wilcoxon signed rank test for zero median difference between the values of FD, DTPR, Acc_p and Acc_a for the combined and HODA approach significantly favours the former for all of them ($p < 0.01$). This indicates that incorporating statistical features is more powerful both for the correct rejection (measured by FD and DTPR) and the recognition (quantified by Acc_p and Acc_a). Yet, HODA features alone outperform simple pattern features such as the ones used in the combined approach. Therefore, in the future, it could be beneficial to study the effects of merging the feature set obtained by HODA with the statistical features in the combined approach, that is, replacing the simple features in the combined approach. This might lead to a further improvement.

Finally, after recognizing the activities, either with HODA or the combined approach of the previous Chapter, a next step would try to extract activity characteristics related to the functional capacity. In this regard, the low number of false detections for the majority of patients, in both approaches, is a key strength. False detections would introduce meaningless assessment characteristics. A lower detection rate is less important as long as enough data is available to still assess the functionality. The next Chapter discusses this assessment in detail.

7.5 Conclusion

This chapter attempted to classify data solely based on pattern information. After showing that its straightforward use is only viable in a very limited setting, a way of stretching patterns using an adaptation of the continuous wavelet transform was discussed. This led to a large number of false positives. Also,

increasing the number of activities would require a high computational effort since matches are checked for every position.

The usefulness of patterns could be increased by combining the well-known DTW with structural information of the data. This Chapter introduced a new multiway data representation encoding DTW similarity to all considered activities. Classification features were extracted by application of the HODA tensor decomposition. Segmentation was dealt with explicitly, such that a discrete problem was obtained from continuous data. The Chapter also contributed to the research on classification in the context of random movements through the introduction of an additional rejection class in the final classifier.

The developed approach was experimentally validated by segmentation and recognition of six activities from a single arm-worn accelerometer for 39 axSpA patients. The comparison to a simpler direct DTW similarity approach showed the added value of the structural information which led to significantly better results. HODA segmentation and classification showed good performance individually, but their combination led only to good results for part of the patients, in particular a subset of the data that was recorded with another device. For future work, the findings of this chapter could be combined with the merits of statistical features as established in the previous Chapter to improve performance.

The current and previous Chapter lay the foundation for a framework of automatic activity segmentation and recognition. It is the basis to build an assessment system, discussed in the next Chapter.

When you tell them you've made a new friend they never say to you "What does his voice sound like? What games does he love best?" Instead they demand "How old is he? How much does he weigh?" Only from these figures do they think they have learned anything about him.

A. de Saint-Exupéry, *Le Petit Prince*

Chapter 8

Activity assessment

Up till now, only activity recognition has been discussed. A next step in the support system pipeline is the investigation of approaches to quantify functional capacity. This Chapter builds on the previous Chapter's techniques to segment continuous acceleration signals and recognize the segments as activities from a predefined set. Using this information, objective quantification can be based on the assessment of the recognized segments, or on more general measures over the entire measured signals. Therefore, this Chapter has a double aim. On the one hand, it investigates parameter extraction. On the other hand, it uses extracted parameters in proof-of-concept decision support systems. In this way, it provides objective alternatives for e.g. the subjective BASFI questionnaire or similar systems. Nevertheless, the developed approaches are not intended as a replacement, but as a complement. The subjective information in itself should also be taken into account.

Section 8.1 focuses on the extraction of parameters from segmented activities. More specifically, it aims to capture interpretable characteristics for use in decision systems or for statistical analysis. The next Section proposes three ways to implement instrumented BASFI, an objective approximation of BASFI: two using regression and one using Interval Coded Scoring, discussed in Part I. Two of these approaches have been published in [21, 23]. Finally, Section 8.3 presents a more general example of quantification of disease activity via ICS based on hospital records.

8.1 Parameter extraction

Once activities have been segmented and recognized, parameters can be extracted to quantify them. They can be generic, that is, general statistical parameters such as the ones introduced in Chapter 6, Section 6.1.1. Alternatively, they can be activity-specific, that is, based on the measured activity pattern. Yet, the distinction is not always clear, because the interpretation of generic parameters can depend on the activity. A typical example is duration. It is used in performance-based tests as an indicator of functional capacity. It has already been shown that it can be extracted from the acceleration signal and that this automatic estimation is a statistically valid replacement for the use of a stopwatch [159]. Although it can be extracted without knowing which activity has been performed, its interpretation only makes sense when linked to an activity. For example, the durations of picking up a pen and performing five repetitions of sit-to-stand are not directly comparable.

This section contains two parameter extraction studies. The first consists of the extraction of motor control parameters based on the shape of the acceleration patterns of several activities. The second aims to extract duration and the number of steps of accelerometer acquisitions of patients performing a Timed Up and Go (TUG) test. Segmentation and recognition are supposed to be known, but could in principle be the result of the approaches of the previous Chapters.

8.1.1 Motor control parameters

This parameter extraction has been performed on the Motor Control dataset defined in Chapter 5, Section 5.5. The first session with three repetitions of five activities (flexion, extension, left and right lateral flexion, functional bow) has been used. Recall that the ViMove sensor yields orientation angles along the sagittal and the transverse axis. Here, only the main movement axis has been used for parameter extraction, despite the fact that rotations around the secondary axis also yield potentially important information. For example, medio-lateral sway has been linked to the progression of Parkinson's disease [105].

Each activity trial is characterized by a shape that is approximately unimodal, either convex or concave, depending on the activity. In what follows, the concave shape will be discussed, but convex shapes are processed similarly. Various parameters can be extracted to quantify the shape:

- The **overall maximum angle** θ_m indicates the typical range of motion of a patient. In case of convex shapes, it is a minimum angle.

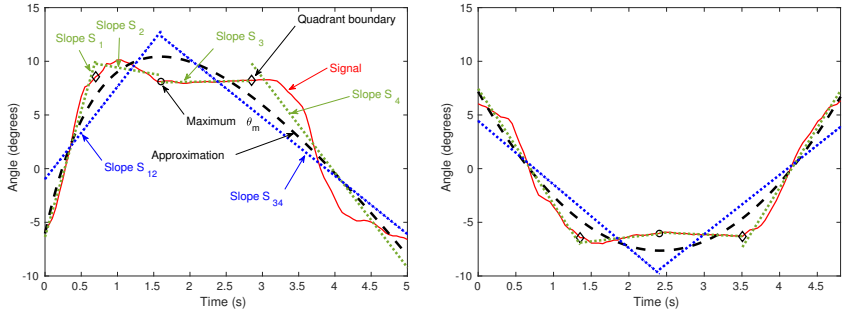


Figure 8.1: Examples of the computation of the motor control parameters for flexion (left) and extension (right) with exception of the quadrant averages and medians. The angle θ_m is calculated based on a rational approximation of the signal. The position of this angle is then further used to calculate the quadrant boundaries and slopes directly from the signal.

- The time position of the angle can be used to divide the movement into two phases, ascending and descending. The average **slopes** S_{12} **and** S_{34} of these halves allow to study the effectiveness of different muscles and the impact on the spine in the two movement phases.
- Both phases can be further split into two yielding a total of four quarters. The quarters are a more precise approximation capturing the onset and the end of the ascending phase and the onset and end of the descending phase, related to the responsiveness of the muscles. This can be characterized by the four **quarter slopes** $S_1, \dots S_4$, the **quarter averages** $\mu_1, \dots, \mu_4$ and the **quarter medians** $M_1, \dots M_4$.

These 15 parameters are extracted via an approximation of the shape as a rational function with a quadratic numerator and denominator. This function allows to take into account asymmetry, as compared to a more simple polynomial approximation. It is used to determine the angle θ_m because the approximation captures the main shape, neglecting any local deformations. The other parameters are directly calculated from the signal itself. Figure 8.1 displays the extracted parameters in case of flexion (concave shape, left) and extension (convex shape, right).

The Figure shows the signal in red and the black dashed rational approximation. Note that the movement does not start at a zero baseline. This can be corrected if necessary for a correct assessment of the range of motion. Based on the approximation, θ_m is computed first, dividing the signal in two phases. Least-squares linear approximations of the original signal indicated by blue dotted lines yield the slopes S_{12} and S_{34}. The two black diamonds represent the

splitting points for each phase, dividing it in two quadrants. This splitting point is defined as the point for which the linear approximation in the resulting quadrants yields the closest Euclidean distance to the original signals. Once all quadrants are known, their averages and medians (not shown) can be computed. Furthermore; they also yield the slopes S_1 to S_4 via the least-squares linear approximations of the quartered signals.

The information that has been extracted can be used in multiple ways.
On a single patient, as in the Figure, the motor control parameters already quantify several movement aspects. For example, the fact that both for flexion and extension, the patient first bends as far as possible (overshoot), before bending a bit back to find a stable position around the angle θ_m. After remaining static for a while as requested, the reverse effect can be observed.
Moreover, the parameters of a single patient can be compared over multiple trials executed one after another to see if fatigue has an effect. Another possibility is comparing different recording sessions over time to study the effect of therapy. Finally, parameters obtained from several patients can be jointly analysed to establish whether there is a statistically and clinically significant relationship with markers for disease activity and functional capacity in current clinical practice.

The extracted parameters have been studied in the context of master theses at the Division of Rheumatology, University Hospitals Leuven and the Faculty of Rehabilitation Sciences, KU Leuven. The parameters under study are already used in patients studies, showing their clinical relevance, but in most cases they are extracted via a static vision-based setup in a motion laboratory. Extracting them from IMU sensors is a more convenient and cheaper approach that is not limited to a laboratory environment. In order to do that, the IMU parameters should be valid and reliable. *Validity* answers the question whether one measures what one is supposed to measure, that is, the difference with ground truth data. Validity of the ViMove has been proven in literature [10]. *Reliability* answers the question whether the readings are and remain consistent after a longer monitoring period. The master thesis investigated the reliability of the parameters, extracted from the commercially available ViMove sensor as described above. The study concluded that the ViMove measured kinematics reliably within session movements, both expressed in absolute terms and relative to the maximal range of motion. Between-session results have also been shown to be reliable when expressed in absolute terms [47] .

8.1.2 Timed Up and Go

This second study focuses on the extraction of the duration and number of steps of hip fracture patients performing a Timed Up-and-Go. It aims to show that

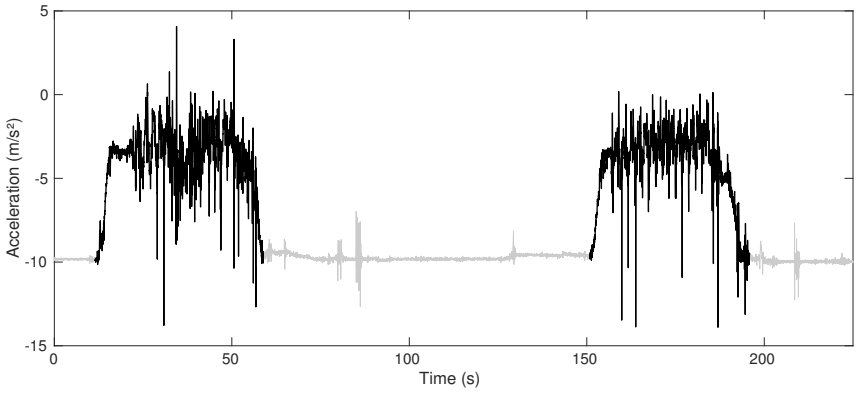


Figure 8.2: An example of segmented TUG trials of a patient, shown on a single acceleration channel. The signal is shown in grey, with the segmented regions in black.

these parameters can be validly extracted from controls and patient acceleration signals in an automated way.

8.1.2.1 Data and methods

The study is based on the TUG dataset described in Chapter 5, Section 5.5. Only the leg sensor has been used. The challenge of the task lies in the diversity of the walking for the patients. Many of them shuffle, which often masks the clear acceleration peaks. This is even more pronounced since all of them use a walking aid.

The duration is roughly delineated manually, but could also be extracted using the segmentation approach outlined in the previous Chapters e.g. in Chapter 7, Section 7.2.2.3. Since no other activities are performed, the segmented regions can be regarded as the activities under study. This segmentation does not require any training, other than setting the parameters for the variance differences distinguishing static from dynamic regions. Additionally, a further refinement based on the low-frequency wavelet components, adapted from [159], is applied for an even finer granularity. An example of a segmented TUG for a patient is shown in Figure 8.2. Only one of the three sensor channels is shown for clarity's sake.

Steps can be detected via the sudden acceleration peaks caused by the heel strike. Hence, the detection is implemented as peak finding on the bias-corrected magnitude of the detected TUG segments. This requires heuristic estimates

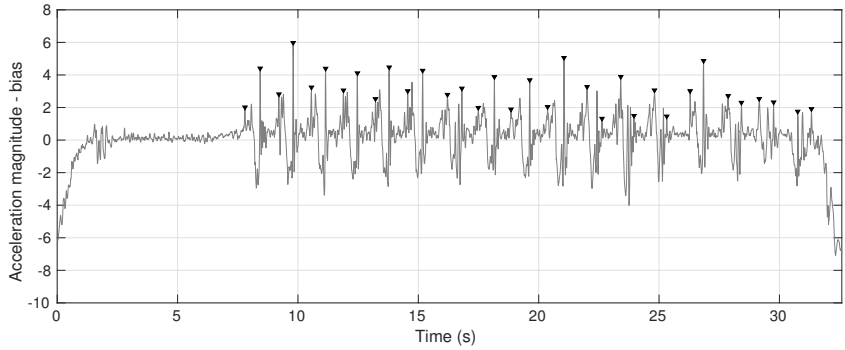


Figure 8.3: An example of the outcome of the peakfinding algorithm for a patient TUG trial. The black triangles indicate the sudden acceleration corresponding to the heel strike of a step.

for the minimum peak distance and the minimum peak height to reject local optima in the signal magnitude which would be false positives. For the healthy controls, the minimum distance is shorter and the peak height is larger than for patients. This prior information can be supplied as constraints in the peak finding algorithm. Figure 8.3 displays a TUG trial with detected steps. Notice how the difference between the alternating peaks corresponding to steps with the leg with and without sensor is large in the beginning and diminishes near the end. Also, the peakfinding does not find exact positions of the steps, which is evident from its selection of sometimes smaller peaks next to larger candidates which have been excluded due to the proximity constraint with regard to the previous peak. The positioning could be improved by a more aggressive low-pass filter to focus on the lowest frequencies of the signal, but this tends to remove the less prominent steps with the sensor-less leg.

8.1.2.2 Results and discussion

Table 8.1 lists the mean and standard deviation of the absolute differences of the duration and the number of steps for patients and control subjects. Additionally, the last row contains the p-value for a Wilcoxon signed rank test with the null hypothesis of equal distribution medians between the estimated and true values.

The Table shows that TUG duration estimation and step count are reasonably accurate for healthy controls. Yet, the estimated duration is significantly different from the ground truth measured manually via a chronometer. The full data reveals that the duration is overestimated for all subjects. This might be due to the segmentation approach assuming movement even after the actual

Table 8.1: Mean (MGTr) and standard deviation (STDGTr) of ground truth duration, number of steps, mean absolute error (MAE) and standard deviation (STDAE) in duration and number of steps for patients and control subjects with respect to the ground truth. A Wilcoxon signed rank test further compares the obtained error distribution to the ground truth and shows significant difference for the duration estimates ($p \leq 0.05$).

	Control		**Patients**	
	Duration	**Steps**	**Duration**	**Steps**
MGTr	7.92	12.35	59.93	37.06
STDGTr	1.09	1.29	49.45	13.69
MAE	1.68	0.9	3.38	5.03
STDAE	1.44	0.72	4.95	5.03
p-value	9e-5	0.47	0.05	0.24

TUG segment is finished. For the steps, the assumption that the count is on average correct cannot be rejected. On average, the step count deviates less than a single step from the actual average count of the observers.

The algorithm seems to perform worse for the hip fracture patients, but one should take into account that these are absolute errors. On average, the patients take about 7.5 times as long to complete the TUG (with a large variability) and take approximately three times as many steps. In that light, the relative duration estimation performs better for the patients. It is on the border significance for difference from the ground truth ($p = 0.05$). Also, there is no constant overestimation as was seen for the control group. In fact, the patient estimations have a very high correlation to the chronometer data ($\rho = 0.996$), whereas this is not the case for the control data ($\rho = 0.450$). This means the estimations are a valid means to compare patients to one another. In contrast, the estimated step count performs worse than for the control group even when correcting for a higher number of steps. The difference with the observers is still not significant. This is expected due to the less pronounced walking pattern of the patients with respect to the control group.

The differences between the healthy subjects and hip fracture patients are not only due to the surgery. They are probably mostly due to the large age difference of on average 54 years between both populations, as shown in Chapter 5, Table 5.6. Hence, no statistical comparison between groups has been performed. Nevertheless, conclusions can still be drawn for both groups individually based on the statistical evaluation of the difference with the ground truth.

Summarizing, the automatic step count can be used as a replacement for

observers' counts. Duration estimation is relatively close and, for patients, is highly correlated with the correct results. It can therefore be used for comparisons among subjects or between subject trials, though not with respect to thresholds used in literature to e.g. indicate an increased falling risk. Note that these are all statistical conclusions, whether the observed differences are clinically significant should be the subject of further study.

8.2 instrumented BASFI (iBASFI)

This Section explores a more direct approach to activity assessment than interpretable parameter extraction. It aims to produce models that approximate the BASFI questionnaire scoring, listed in Appendix B. These objective models are all different forms of instrumented BASFI (iBASFI). The questionnaire leads to a subjective result but it has been shown to be a reliable and valid measure, although its sensitivity to detect subtle changes has been questioned [185]. Hence, creating an objective model minimizing the overall error could be a valid complementary measure. The complementarity stems from the fact that the questionnaire is also directly influenced by factors such as expectations, whereas the model would only take into account the actual activity performance.

The models are constructed from the segments that can be extracted and identified using the approaches in the previous Chapters. Parameters are extracted for each segment and concatenated into a single feature vector for every patient, which will typically be high dimensional. The main aim of the Section is therefore to explore which activities, which parameters and which techniques yield good BASFI approximators, to assess their explanatory power and to investigate whether they outperform movement duration as a sole indicator of performance.

8.2.1 A stepwise approach

A first tentative way to quantify functional capacity from segments generates many well-known signal processing features. Then, it builds a linear model using stepwise selection to limit model complexity.

8.2.1.1 Data and methods

The experiment is carried out with the two-channel SenseWear dataset described in Chapter 5, Section 5.5. As mentioned earlier, the reliability and validity of automated algorithms to extract movement duration from accelerometer signals

have already been demonstrated for this data [159]. Here, more complex signal features from the signal processing community are investigated based on seven of the recorded activities: *getup*, *liedown*, *pen5*, *reach5*, *sts5*, *socks* and *stairs*, although the latter two have not been included in the recognition in the previous Chapters. For every activity, two trials are available, but only the first has been used.

The features are defined independent of the activity. They consist of a set of common measures in signal processing [98] including e.g. duration, standard deviation, root mean square, number of zero crossings, line length, kurtosis, skewness, signal magnitude area, spectral entropy, dominant frequency, features on the derivatives etc. All features are calculated on both acceleration axes and on the magnitude and phase signals of the acceleration vector. Moreover, the features that are length-dependent are normalized with respect to the signal length. This yields a total of 169 features for each activity.

To avoid overfitting problems with such a high feature space and a very small dataset of 28 patients a stepwise linear regression is performed for each activity separately. A linear model including an offset is constructed with all possible single features. Then, a forward pass determines whether it is an advantage to add another feature and which should be selected. Further in the process, a backward pass might also lead to the removal of a feature because the resulting smaller set is superior. This is possible due to the greedy nature of the forward pass which only looks a single step ahead.

The optimality criterion is the R^2 value which is also called the coefficient of determination. In linear models, it corresponds to the percentage of variance in the data explained by the model. Its adjusted version R^2_{adj} corrects for the number of terms in the model, since a high dimensional model is likely to have a better fit even when including many unnecessary variables. Hence, R^2_{adj} can only be lower than R^2 [61].

In the stepwise procedure, a feature is included in the model if the R^2 gain is at least 0.1. In the backward pass, features with less than 0.05 R^2 gain loss are removed. The Matlab function `stepwiselm` was used to build the model.

8.2.1.2 Results and discussion

The final feature selection results are shown in Table 8.2. For all performance-based tests with complex accelerometry-derived movement features the F-statistic for the difference with a constant model was significant ($p < 0.01$) and the models explained more variance (R^2_{adj}) than models including duration only. No single generic explanatory feature was detected across tests.

Table 8.2: Results of the stepwise feature selection for all activities. Explanatory power is expressed as standardized Beta values (stB) per feature and adjusted R^2 values, both for the model with accelerometry features, including duration (R^2_{adj}) and for models only based on duration ($dur R^2_{adj}$).

Activity	Feature	R^2_{adj}	durR^2_{adj}
getup	#maxima of angle of derivative (stB 0.463) spectral energy of angle of derivative (stB -0.415)	0.728	0.209
liedown	spectral energy of angle of derivative (stB -0.719) max absolute value of angle of derivative (stB -0.444)	0.567	0.204
pen5	minimum magnitude of derivative (stB 0.429) autocorrelation of longitudinal axis (stB -0.415)	0.471	0.209
reach5	maximum absolute value of angle (stB 0.606) skewness of angle (stB 0.617)	0.553	0.065
sts5	dominant freq of magnitude of derivative (stB -0.707)	0.481	0.376
socks	complexity of magnitude (stB 0.572) crosscorrelation between axes (stB 0.0.889)	0.509	0.078
stairs	10% spectral edge freq of magnitude (stB -0.747)	0.541	0.356

The results show that all activities can be linked to the overall (average) BASFI score. This implies that the BASFI could be approximated from a single activity using only one or two features. Moreover, even when only a single feature is selected, it is not the duration: the model with a single selected complex feature outperforms the duration model e.g. for *sts5* or *stairs*. The selected features are quite diverse, but many measures involve the angle or magnitude of the derivative of the signal channels. The biggest drawback of the study lies in the sample size: only 28 observations have been used, which severely limits the generalizability of the findings.

This study used a stepwise approach to limit the impact of the high dimensionality. Alternatively, a penalty term could be used to avoid high-dimensional models. This approach is used in the next Section.

8.2.2 Modelling with elastic net

Instead of stepwise linear regression, the elastic net can also be used for construction of a linear model while at the same time performing dimensionality reduction due to a penalty term. The method was already introduced in Chapter 2, Equation 2.1 (page 15) and used for the `enICS` approach in Chapter 4.

8.2.2.1 Data and methods

The constructed model is also based on the SenseWear dataset, but joined with the Shimmer data presented in Chapter 5, Section 5.5. The triaxial data of the latter's arm sensor was reduced to two axes to match the SenseWear setup, similar to the approach in Chapter 7. Across both recordings the total patient population consists of 39 patients. One patient had to be removed since the BASFI data for the Shimmer set were incomplete. As in the previous experiment, the total BASFI score, that is, the overall indicator calculated from the BASFI questionnaire, is used as labels for the model.

The model is based on five of the ten performed activities: *getup*, *pen5*, *reach5*, *socks* and *sts5*. Both trials of the selected activities are used such that the total number of available observations is 76, that is, the repetitions are considered as separate patients with identical BASFI score. The two trials then explicitly capture the within-patient variability contained within the same BASFI score.

The features are the same as in the stepwise regression but now, all activities' features are concatenated into a single data matrix $\mathbf{X} \in \mathbb{R}^{76\times 845}$ leading to a total of 845 features. Hence, the feature selection in elastic net gives direct insight into the importance of the activities as well as the features. The implementation of elastic net for this study is the standard implementation in Matlab. It has a slightly different notation than mentioned before in Equation 2.1. It solves

$$\min_{\mathbf{w},b} \frac{1}{2N}\|\mathbf{X}\mathbf{w} + b - \mathbf{y}\|_2^2 + \lambda\left[\frac{1-\alpha}{2}\|\mathbf{w}\|_2^2 + \alpha\|\mathbf{w}\|_1\right]$$

In this equation, N = 76, the number of observations and α is a hyperparameter making a trade-off between the pure ℓ_1 penalty (LASSO) and a larger impact of the ℓ_2 penalty (elastic net/ridge regression). The ℓ_2 norm improves the condition of the problem, but, as elsewhere in this work, it is further of little interest. Therefore, the value of α is fixed to 0.99. λ is the trade-off parameter between the goodness-of-fit corresponding to the first objective part before it and the penalty behind it, hence combining the roles of λ_1 and λ_2 in Equation 2.1.

The elastic net is applied on all features. For this case, an important criterion is whether the durations have been selected. This model will be compared to the case where only the five duration features are used. The elastic net yields a trace of possible models for 100 values of λ uniformly distributed between 0.01 and 1. Zero is left out because it corresponds to no regularization. The models will be selected and compared based on their leave-one-out cross-validation root mean squared error (RMSE) and their Bland-Altman plots. Such plots represent the agreement or disagreement between two data series supposedly containing the same information e.g. two scoring systems. It uses the average of the two series' data points as independent and the difference between them

as dependent variable. Hence, the vertical axis shows whether a systematic bias occurs and what the difference range between the series is [28].

8.2.2.2 Results and discussion

Figure 8.4 presents the traces of the elastic net: the value of λ has an influence on the number of variables included in the final model and on the RMSE obtained through leave-one-out cross-validation. Due to the nature of the method, the number of variables decreases for increasing value of λ. Consequently, the RMSE increases and the model becomes simpler. This also explains partly why the duration models (shown in dashed) obtains worse results than the models obtained from accelerometer features. The Figure further indicates three models that will be discussed more in detail: the RMSE-optimal duration model (indicated with a circle on the dashed line), the RMSE-optimal low-complexity feature model (indicated with a circle on the solid line) and the overall RMSE-optimal model (indicated with a star). Note that the simple accelerometer model still has three times more variables (12) than the duration model (4).

Before going into the details of the selected models the importance of the features will be discussed. Figure 8.5 displays the 25 most important features other than the durations, as assessed by elastic net, ranked according to their importance over the entire trace. Elastic net selection for a value of λ does not directly depend on the selection for other λ values. Hence, features can have a gap in their selection bar. The ones shown here have the highest number of selections over all values of λ. The five duration features have been added as

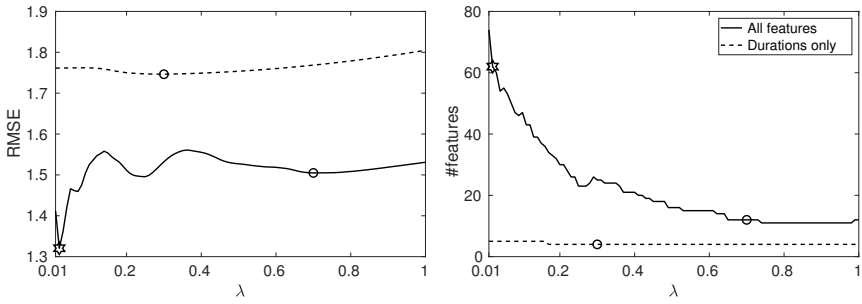


Figure 8.4: Elastic net traces showing the influence of λ on the RMSE (left) and the number of variables (right). The solid line shows the trace starting from all features, whereas the dashed line represents the use of duration features only. The circles indicate two models that will be described more in detail, whereas the star denotes the overall RMSE-optimal model. The duration models are all simpler, but less performant.

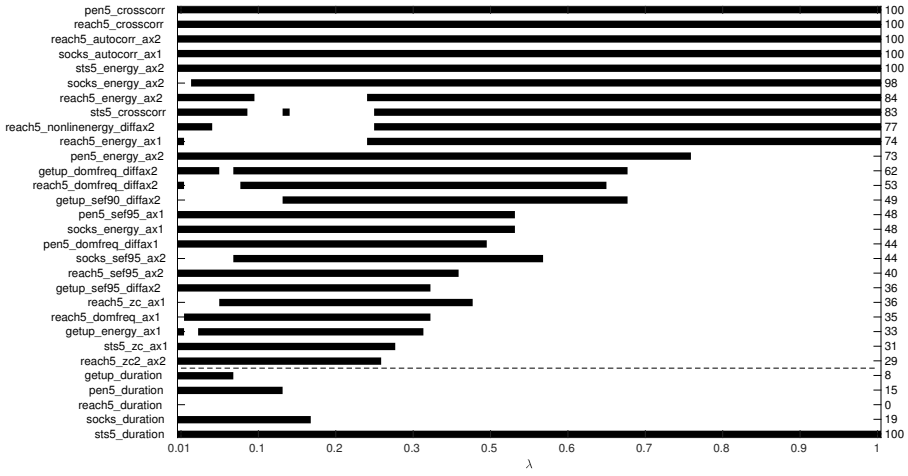


Figure 8.5: The 25 most important features based on their selection as a function of λ. The black bar shows for which values of λ each feature was selected. Below the dashed line, the importance of the duration features is highlighted as well. The numbers at the right of the bars show the percentage of selection over all values of λ.

well. The *sts5* duration appears as an important feature, selected for all models. In contrast, the other duration features are quickly abandoned in favour of other signal processing features. *reach5* duration is never selected at all. Among the other features, the normalized cross-correlations and autocorrelations are considered important, followed by the normalized energies, the nonlinear energy of the derivative (diffax), the dominant frequencies of the derivatives and various percentiles of the spectral edge frequencies (sef). Finally, zero crossings (zc) are also deemed important.

Based on these results the duration features seem unimportant, but several features, in particular the auto and cross-correlations are heavily correlated with them ($\rho \approx 0.75$). Yet, removing them before running the model yields overall worse models. Hence, they do contain other important information as well.

The optimal duration model is obtained for $\lambda = 0.3$. Leaving out one feature improves the results, so the optimal model only consists of the durations of *getup*, *pen5*, *socks* and *sts5*. Figure 8.5 already showed that *reach5* duration was never selected in any general model either. Its rejection is confirmed by the duration-only model. Its optimal RMSE is 1.746, but this only gives a partial

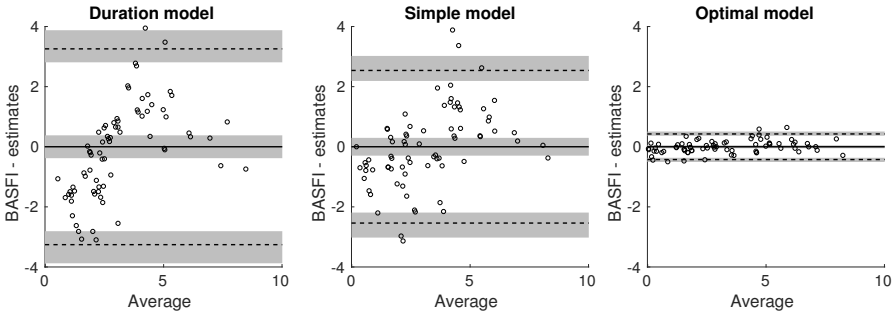


Figure 8.6: Bland-Altman plots for the best duration model (left), a simple feature model (middle) and the optimal feature model (right). The bold line denotes the bias of the model, the dashed lines the 1.96 std boundaries. The gray-shaded zones indicate their 95% confidence intervals.

result. The left part of Figure 8.6 shows the corresponding Bland-Altman plot for a more detailed analysis. On average, its bias is zero (solid line), but 95% of the data deviates between -3.257 and 3.257 from the ground truth BASFI. Also, that deviation is correlated with the average between the BASFI score and its estimate, indicating a dependency on the score itself. Indeed, the Pearson's correlation between the BASFI score and the difference between BASFI and its estimate is 0.772. In other words, the estimate aims too often for the average BASFI: low BASFI values are overestimated, high values are underestimated.

The middle part of Figure 8.6 displays the feature model indicated by the circle on the solid curve in Figure 8.4. It is the best feature model with low complexity, consisting of twelve features: the top 11 in Figure 8.5 and the duration of *sts5*. It is an improvement on the duration model, mostly caused by its increased dimensionality. The bias still remains around zero, but the 95% range is reduced to $[-2.539, 2.539]$. Consequently, its cross-validation RMSE of 1.505 is also lower than the duration model. The correlation can still be observed, but the data is more scattered. The correlation coefficient amounts to 0.6526.

The last model is overall optimal through cross-validation with an RMSE of 1.321. Its γ value of 0.02 leads to 62 variables. Due to much more information being available, the estimations are much more precise. The 95% interval of the deviations is $[-0.426, 0.426]$. The correlation has also decreased further ($\rho = 0.354$).

In the overall optimal model, 12 variables come from *getup*, 13 from *pen5*, 17 from *reach5*, 8 from *socks* and the remaining 12 from *sts5*, so all activities remain necessary. Also, it is important to mention that for this detailed model, all durations except *reach5* have also been included. Other than that, 15

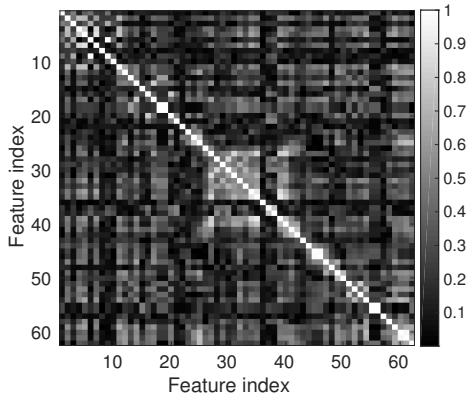


Figure 8.7: Grayscale representation of the absolute correlation coefficient between all selected variables in the optimal model.

types of features were included: the autocorrelation, crosscorrelation, dominant frequency, energy, kurtosis, line length, number of local optima, nonlinear energy, spectral energy fraction with cutoff 50%, 85%, 90% and 95%, the spectral entropy and the number of zero crossings. Moreover, both axes and their derivatives were indicated as important.

In this final model, several variables are correlated. This is due to the nature of elastic net, which favours grouping. If no ℓ_2 term is present (LASSO), it will randomly pick one of the correlated variables, whereas all are selected with lower weights in the full elastic net. Figure 8.7 gives an overview of the absolute values of the correlations. As can be seen, most variables are uncorrelated. However, a few are perfectly correlated, indicated by light squares off the main diagonal, and should better be removed in a future model.

The selected models show that all durations except for *reach5* contain important information with regard to functional capacity. Yet, the assessment remains relatively poor and can be improved by introducing more features or picking alternative features, which have been shown to be correlated with the durations.

Overall, this study shows that accelerometer features are a promising approach to objectively model the BASFI scoring system. Duration features can be complemented by, or replaced with, more sophisticated features to improve the linear model. Some correlations occur, but the majority of the selected features is uncorrelated. In the future, the model should be validated on external test data to assess its generalization power more correctly, since the current results are only based on cross-validation.

The previous study showed that simple models based on complex features

outperform duration-only models. The findings in this study somewhat contradict this. This can be explained by a number of reasons. First of all, with elastic net, all activities are taken together. In the stepwise study, the aim was to see if anything could already be derived from a single activity. Secondly, the modelling criterion is different, elastic net aimed to minimize the cross-validation RMSE, whereas stepswise selection was based on the R^2 value, the percentage of explained variance. Finally, the elastic net study uses 38 instead of 28 patients and both trials of all selected activities. Nevertheless, both studies put an emphasis on the use of autocorrelation and cross-correlation features, as well as the usefulness of the derivative signal. These insights could be used as leads for further research.

8.2.3 Approximation as classification

The previous studies to create an iBASFI system were intended to create a continuous objective model by means of regression to approximate the current BASFI questionnaire. Alternatively, one can also model it as a classification-based scoring system by thresholding the BASFI score, where the ICS score or, equivalently, the predicted risk can be used as a BASFI surrogate. This study applies `lpICS`, discussed in Chapter 3, to obtain a scoring system that can be used as such. It has been developed for application with the activity classification approach described in Chapter 6.

8.2.3.1 Data and methods

As in Chapter 6, the assessment is carried out on the 28 patients from the SenseWear dataset based on six activities: *getup*, *liedown*, *maxreach*, *pen5*, *reach5* and *sts5*. The average BASFI score is 3.14. Based on this value, the classification criterion becomes $BASFI < 3$ *vs* $BASFI \geq 3$ to create a balanced dataset. The target class BASFI ≥ 3 corresponds to a decreased functional capacity.
Since features used for assessment need to be interpretable they have been directly defined on the acceleration signals. Most of the information of interest is contained in the longitudinal channel due to the nature of the movement. This gives as features, with the number between parentheses (#activities×#channels):

- The duration of each activity (6×1).
- For *getup*, *liedown* and *maxreach*, the maximum acceleration (3×1), the slope of a linear approximation of the signal (3×1) and the variance

around this slope (3×1) for the longitudinal channel and the variance of the sagittal channel (3×1).

- *pen5* and *reach5* contain local maxima in the acceleration signals. As features, one can take the average peak values (2×1) and the average (2×1) and standard deviation (2×1) of the peak-to-peak duration. This captures their acceleration capability, as well as the variability in the execution of the repetitive activities. Furthermore, this is quantified even further by also looking at the average foot-to-peak duration (2×1) and the average slopes of these subsegments (2×1). This is all measured on the longitudinal channel. For *reach5*, the peaks in the sagittal channel are clear as well, hence, the average peak value is yet another feature (1×1).
- *sts5* has a more complicated acceleration pattern in both channels with two kinds of local peaks: the locally maximal acceleration towards sitting and towards standing. Therefore, on both channels, the average 'sit peak' (1×2) and 'stand peak' value (1×2) are defined. Moreover, the mean foot-to-peak durations can be derived for sit (1×2) and stand (1×2). Finally, the non-smoothness of the movement is measured by the variance of the channels (1×2) filtered with a high-pass fourth order Butterworth filter (cutoff at 2.4Hz).

The features defined on all activities are merged into a single vector of 39 assessment features. This also implies that the dataset for assessment only consists of 28 observations. In order to still derive some useful information, a leave-one-subject-out test procedure was used. Still, even with this approach, only 27 observations are available to build a model. Therefore, a bootstrap-like approach explained below further minimizes the impact of a single observation.

Furthermore, the settings for `lpICS` also take into account the limited data. The maximum number of intervals is limited to seven and no interactions are allowed to further simplify the data analysis and avoid an increase of the data dimensionality. No preselection was performed and the trade-off between model complexity and performance was decided automatically with the default AUC cutoff of 0.95.

The total approach is summarized in Algorithm 8.1.

8.2.3.2 Results and discussion

The goal is to predict a patient's class label, that is, whether his BASFI value is higher or lower than 3. The repetition of step 3, the variable selection, yielded 28 sets of selected variables. Most of these selections only kept 2 out of 39 variables (43% of the cases), but also 3 variables (32%), 1 variable (18%) or

Algorithm 8.1 iBASFI via ICS classification

1. Select one observation for testing, the other 27 make up the training set.
2. Make all possible $\binom{27}{25} = 351$ subsets of 25 observations out of the full training set. Derive an `lpICS` model, barring interactions, for each of these subsets. Each obtained ICS model indicates which variables have been selected.
3. Retain only the variables that have been selected in more than 25% of all the models. Variable selection based in this way on all combination sets allows a more stable decision than running the selection procedure only once on all 27 observations. The remaining number of selected variables is low (usually 2 to 4). Hence, ICS is used as a variable selector.
4. Build a final `lpICS` model from all 27 training observations using the selected variables from the previous step and predict the class of the test observation.
5. Restart the procedure from step 2, until all observations have been classified.

more than 3 variables (7%) occurred as outcome. The variables that were selected most often were the *sts5* longitudinal average foot-to-peak time (27 out 28 cases) and the duration of *liedown* (16 out of 28 cases). The former quantifies the control one exerts over the acceleration, how long it takes to go from maximum deceleration to acceleration. The latter confirms duration as an indicator for objective performance. The selection of *liedown* probably relates to it being a relative complex series of actions including possibly turning, bowing, sitting and finally lying down.

In the fifth step of Algorithm 8.1, 28 new models were trained using the obtained sets of selected variables, each time on 27 training observations, to predict the remaining test observation's class. This corresponds to a classical leave-one-subject-out approach. Comparing the predicted labels with the actual labels obtained in step 1 yielded a final test accuracy of 64.3%. The accuracy on the training data is on average 75.9% with a standard deviation of 4.1%. This is the average accuracy of the 28 models of step 5 on their respective training data. The relative small gap between testing and training accuracies and the not excessively high training accuracies imply that severe overfitting has been avoided.

To conclude, Figure 8.8 shows an example model trained on all 28 observations, where only the two most important variables have been considered. Its (training) accuracy is 82.1%. Note that this is different from the aforementioned leave-one-out (testing) accuracy of 65.3% which was obtained by averaging over all 28 models. To gain further insight in this model, one can also look at the attributed scores. They are obtained by calculating the feature values and verifying whether they lie above or below the data-derived thresholds. Then, summing the corresponding number of points yields a score on a three-point

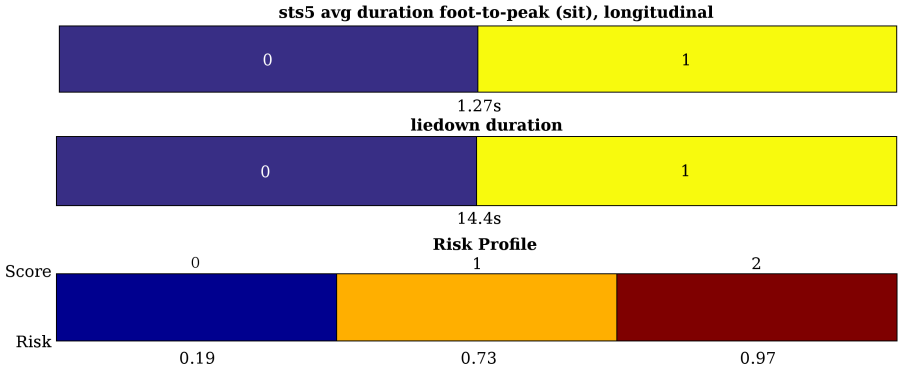


Figure 8.8: ICS model for the classification approach of iBASFI. It shows the selected variables, their intervals with corresponding weights and the Risk Profile mapping the total score to a risk on decreased functional capacity (≥ 3 on the BASFI scale).

scale. This granularity is decided by the algorithm itself. If more information is available, more intervals can be defined or additional variables could be included, which could also lead to more outcome values for the score. This is of particular importance to go beyond mere detection of diminished functional capacity towards tracking changes during treatment or due to intervention. Figure 8.9 indicates how well the ICS score, calculated from objective features, matches the subjective, patient-reported BASFI. Individual observations are shown as black asterisks (*). Box plots capture their general behaviour. One can indeed observe a trend, a positive correlation ($\rho = 0.67$), although the system could still be improved. A more elaborate study with more data would allow to investigate the scoring system's reliability and validity in a proper statistical evaluation.

8.3 Example: a scoring system for patient assessment

The study in this Section focuses on assessment of *disease activity* from general patient characteristics. Therefore, it differs from the previous studies. It is not directly related to the assessment of physical activity, but focuses instead on a characterization of the disease activity, its clinical counterpart. The goal is to medically assess the progression of spondyloarthritis, in general, not merely based on activities. This involves various patient parameters and laboratory

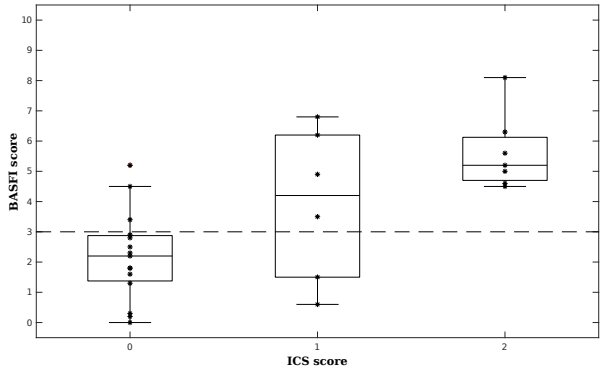


Figure 8.9: A comparison of objective ICS and subjective BASFI.

tests as discussed in Chapter 5, Section 5.1.1. Current clinical practice uses the Ankylosing Spondylitis Disease Activity Score (ASDAS) developed by the Asssessment of SpondlyoArthritis international Society (ASAS) [100]. It can be seen as the counterpart of BASFI: ASDAS focuses on disease activity and BASFI on functional capacity. Of course, they are related since they model the same disease from a different perspective. ASDAS was developed based on a Delphi study with 85 ASAS member experts. They selected 11 candidate variables covering six domains of disease activity. A statistical training procedure on a dataset of 708 patients finally yielded four candidate regression disease scores based on five variables listed in Table 8.3. In this initial dataset, the need to start TNF-α blocking medication was used as ground truth to distinguish between patients with mild and severe progression. However, in the validation study on a separate dataset the patients' and physicians' global assessment was used instead to make this distinction. Details on how the regression systems were obtained can be found in [100].

As mentioned, several scoring systems were constructed. They serve as

Table 8.3: Disease activity-related variables used for the construction of ASDAS modelling the need for TNF-α blocking drug therapy.

Total back pain
Patient global assessment (PGA)
Duration of morning stiffness
C-reactive Protein (CRP)
Erythrocyte Sedimentation Rate (ESR)

Table 8.4: Disease activity variables used for the construction of an ICS ASDAS equivalent with their average, range and percentage of missing values before data reduction. BASDAI: Bath Ankylosis Spondylitis Disease Activity Index, FACIT: Functional Assessment of Chronic Illness Therapy, MASES: Maastricht Ankylosing Spondylitis Enthesitis Score.

Variable (/max)	**Average**	**Range**	**Missing (%)**
Age	49.07	21-82	0.35
Gender	/	191(m), 93(f)	0
Spinal pain (BASDAI2) (/10)	5.73	0-10	0
Peripheral pain (BASDAI3 (/10))	3.87	0-10	0
Morning stiffness (BASDAI5) (/10)	4.96	0-10	0
BASFI global (/10)	4.39	0-10	9.50
CRP (mg/l)	12.55	0-138.8	3.17
ESR (mm/h)	23.16	0-122	4.93
FACIT (/52)	33.27	2-52	15.85
MASES (/39)	0.37	0-19	3.87
Back pain (BPAIN) (/10)	4.20	0-10	1.76
Patient global pain (PAIN) (/10)	5.03	0-10	2.11
Tender Joint Count (TJC) (/70)	1.16	0-24	3.87
Swollen Joint Count (SJC) (/78)	0.27	0-14	10.56

alternatives to be used in case of missing values. For example, ERS and CRP are not always recorded for all patients. The resulting scoring systems suffers from similar drawbacks as discussed in Chapter 2, Section 2.4. They are regression systems with a limited number of effects. They are more advanced than most systems since not only linear effects were discovered. However, the variable transformations were decided beforehand, not discovered through training. The training can still only yield a linear model of the transformed variables. Moreover, the statistical analysis is not founded on optimization criteria. In contrast, ICS could yield a variable selection and arbitrary piecewise constant distribution of the coefficients over their range. Therefore, such a model will be constructed for comparison.

8.3.1 Data and methods

Data of 284 spondyloathritis patients, 191 male, 93 female, has been obtained from the Division of Rheumatology, University Hospitals Leuven. They have an average age of 49.1 years (std 13.2). Fourteen variables have been extracted from the database by an expert rheumatologist. They are listed in Table 8.4. Removing all missing values in these variables, only 195 patients, 136 male, 59 female, remain.

The need for TNF-α blockers, the target variable for ASDAS construction, is not available for this dataset. However, the PGA, used to assess disease severity in the ASDAS validation dataset is present. Therefore, it is used to divide patients into two groups. Low values indicate patients with almost no impact of the disease, whereas high values point to high impact. The target class was defined as patients having a PGA ≥ 3. This yields a balanced classification with 119 patients in the target class (severe disease) and 122 patients in the non-target (relatively mild disease) class. Because PGA is used as target, it will not be included in the ones considered for inclusion in the model. This contrasts with ASDAS. Nevertheless, all other variables contributing can still be discovered.

ICS only yields a single model based on given training and test data. Therefore, to indicate the impact of the train-test sampling, ten-fold cross-validation results will be presented. The models are trained with a maximum of 10 intervals per variable, automatic complexity trade-off and without interactions. For each fold, the complexity of the model can be assessed via the number of selected variables and the total number of intervals. The performance is evaluated by means of the AUC and the slope and bias of the calibration curve.

8.3.2 Results and discussion

The results for `lpICS` are given in Table 8.5. The following variables have been selected over the 10 folds: BASDAI5 (7/10), BASFI (5/10), SJC (2/10), CRP (2/10), FACIT (1/10) and PAIN (1/10).

It is clear from the Table that simple models are preferred, mostly based on only one or two variables. Also, the general performance in terms of AUC is often below 0.7. Furthermore, the calibration slope of 0.76 and bias of 0.12 are far from the ideal unit slope and zero bias. Notice also the large standard deviation, pointing to an unstable model. This indicates the dependence on the exact data used in the fold, that is, the fold does not accurately represent the total population.

Despite the differences between the ASDAS study and our approach, some of the important variables overlap. BASDAI5, selected in most of the folds, corresponds to the duration of morning stiffness. CRP has been selected in some folds as well. In contrast, some other ASDAS indicators have not been found. PGA was used as target, not as independent variable, which excluded its selection. Also, BPAIN and ESR have not been selected.

The experiment can be repeated with the same folds and settings using `enICS`. It selects slightly different variables: BASDAI5 (7/10) and FACIT (7/10) are

Table 8.5: 10-fold cross-validation ($n = 195$) results for disease activity classification with `lpICS`.

Fold	#variables	#intervals	AUC	Slope	Bias
1	4	8	0.71	0.58	0.22
2	2	4	0.63	0.44	0.25
3	1	2	0.79	1.38	-0.15
4	1	2	0.58	0.40	0.30
5	2	4	0.55	0.18	0.37
6	1	2	0.64	0.59	0.22
7	1	2	0.64	0.87	0.10
8	2	4	0.74	0.94	-0.02
9	1	2	0.83	1.52	-0.16
10	1	2	0.69	0.76	0.07
AVG	1.6	3.2	0.67	0.76	0.12
STD	0.97	1.93	0.09	0.43	0.19

Table 8.6: 10-fold cross-validation ($n = 195$) results for disease activity classification with `enICS`.

Fold	#variables	#intervals	AUC	Slope	Bias
1	2	5	0.69	0.60	0.16
2	2	4	0.80	0.89	0.07
3	3	6	0.90	1.20	-0.07
4	2	4	0.67	0.52	0.24
5	3	8	0.83	1.28	-0.15
6	1	5	0.64	0.56	0.23
7	2	5	0.50	0.05	0.40
8	3	6	0.81	0.94	-0.01
9	5	11	0.91	1.15	-0.08
10	2	4	0.67	0.55	0.23
AVG	2.5	5.8	0.7	0.77	0.10
STD	1.08	2.2	0.13	0.39	0.18

followed by global pain (4/10), SJC (2/10), BASDAI2 (2/10), BASDAI3 (1/10), BASFI (1/10) and CRP (1/10). The performance is summarized in Table 8.6

More variables are selected for `enICS` but the correspondence in variable selection with ASDAS does not improve. In contrast, the functional FACIT score is deemed important. The generally more complex models compared to the ones obtained with `lpICS` lead to a slight increase in all performance metrics.

However, a stable model has not been obtained. One can only assume that BASDAI5 plays an important role, based on the two ICS approaches. More conclusive findings require a larger dataset to obtain a more stable model.

8.4 Conclusion

The Chapter presented various ways to quantify patient disease information. A particular focus was on functional capacity starting from activities as recognized in the previous Chapters.

Firstly, two parameter extraction studies were discussed. Motor capacity parameters were presented as an informative and interpretable way to quantify acceleration patterns. Then, the duration and steps from patients performing the TUG test were automatically derived from the acceleration signals. Although the extracted duration differed significantly it was still reasonably accurate. The extracted number of steps could not be shown to be statistically different from the ground truth. Both studies combined show that it the developed approach is a viable way to extract the parameters.

Secondly, the Chapter introduced approaches to implement instrumented BASFI from features rooted in the signal processing field. The regression approaches highlighted the fact that, although duration is informative, more detailed models can be built when additional features are included. This goes for very simple models obtained via stepwise regression, but also for more complex elastic net models. The latter were most accurate, reaching an RMSE below 1.5. An attempt to approach iBASFI as an ICS classification system yielded relatively poor results but still obtained a correlation of 0.67 between the three-point prediction scores and the BASFI ground truth. The iBASFI implementations show that an objective quantification of activity from acceleration data can be performed, although larger studies should confirm the specific models.

Finally, a last study constructed an `lpICS` and `enICS` model for comparison with the ASDAS score with slight changes in the study setup due to variable availability. Cross-validation results were highly variable. Also, the detected effects only partially overlapped with the ASDAS set. This can partly be explained by the different setup. Still, BASDAI5, corresponding to the duration of morning stiffness, and CRP were selected as informative variables.

Overall, the presented results are a proof of concept for objective quantification of disease activity or functional capacity. Together with the activity recognition from previous Chapters, they are an important step towards the construction of a system to support clinicians in the treatment of axSpA patients.

> He looked over the edge. And below him, only a little way below, he saw the beach of ivory sand; the white and amber waves were curling and breaking in foam on it, and across the sea the sun was setting in a haze of gold.
>
> U.K. LeGuin, *The Farthest Shore*

Chapter 9

Conclusions and future work

This final Chapter summarizes the findings of this thesis in Section 9.1 before highlighting the specific contributions corresponding to the thesis's objectives in Section 9.2. Finally, Section 9.3 presents some perspectives on future research.

9.1 Overview

As mentioned in the introduction in Chapter 1 this thesis is part of the SPARKLE multidisciplinary project. Its aim involves the development of hardware and algorithms for the monitoring of kine(ma)tics of axSpA patients in a home environment. This thesis contributes to this larger aim from the engineering point of view in terms of signal processing and development of algorithms in close collaboration with the Division of Rheumatology, University Hospitals Leuven, and the department of Rehabilitation Sciences, KU Leuven. Other KU Leuven faculties were involved in the project as well, but much of our work did not involve direct collaboration.

The two Parts of this thesis fit into the SPARKLE framework. The first Part explores mathematical formulations for interpretable classification systems. The development of ICS is based on former research and was not initially part of SPARKLE. Yet, Chapter 8 shows how ICS can be used for patient assessment where interpretability is a key concern. The second Part mostly presents algorithms directly involving specific activity-related problems. Therefore, it is more application-oriented than the first Part.

9.1.1 Interpretable classification

The first Part of the thesis hinges on the assumption that interpretability is a primary objective in clinical decision support. Many data-driven approaches such as Support Vector Machines or the recently re-emerging (deep) neural networks do not support interpretability. Furthermore, it should be possible to incorporate expert knowledge into a training procedure. In that way, a support system can not only provide new insights, but can also come up with models that do take into account current practice, inspiring trust.

Interval Coded Scoring consists of dividing all observation variables into intervals, leading to a binary encoding of the data. Then, weights for the intervals are assigned based on optimization theory. These weights are further refined to reach a desired performance-simplicity trade-off. Note that this general formulation in terms of intervals does not impose restrictions of e.g. linearity. Any piece-wise constant model is possible. Such models have the ability to approach any possible weight distribution at the given level of detail corresponding to the number of intervals. The summation of the final integer weights for an observation leads to a score that can be linked to a risk value.

The original formulation adapts the SVM framework to take into account total variation regularization. Due to its solution via linear programming, it is called `lpICS`. It was extended to support any kind of interactions between variables. The resulting algorithm proved robust with regard to noise and non-informative variables. Yet, its solution is not unique and large datasets need a very long training time, although this can be improved via the support of IBM's CPLEX optimization algorithms.

Therefore, Chapter 4 introduced preselection as a light-weight interval informed feature selection process to reduce the dimensionality of the original problem. Further attempts to improve uniqueness properties and execution speed lead to `enICS`. Its speed improvement stems from the solution of the elastic net via a transformation to a dual SVM. Unfortunately, it does not support interactions. Adding constraints or decomposing the interactions with Dykstra's approach made interactions possible but this reformulation nullified the improvement in execution speed.

A statistical comparison on six publicly available clinical machine learning problems showed the feasibility of using ICS. It produced simple interpretable models that could, for the given datasets, not be shown to be inferior to standard machine learning algorithms such as Decision Trees, Naive Bayes or SVMs.

The developed algorithms have been incorporated in a Matlab toolbox. It offers an easy formulation of new problems starting from demo files. This allows to use ICS as an exploratory tool or for the formulation of new models.

Its strengths include an interface for manual intervention for preselection and iterative reweighting, although automatic tuning is possible as well. Furthermore, the toolbox comes with a colour-coded visualization of the model to improve interpretability and allow for manual usage, if so desired. It also offers various model performance criteria on cross-validation training and on test data, both for classification and for risk calibration.

9.1.2 Activity recognition

The activity recognition Chapters were primarily focused on two datasets of axSpa patients performing an activity protocol. The activities were selected based on the BASFI questionnaire, a current standard in clinical practice. It mostly features transitory activities, since they yield more insights on activity capacity. The simple single wearable sensor setup, particularly in case of the SenseWear, allows to perform this protocol easily at home as well. The first goal for home monitoring is the recognition of these activities from accelerometer data.

On a larger scale, the Chapters assumed that activity recognition of transitory activities could be improved by using features derived from pattern matching. This contrasts with the sliding window approach with statistical features that is more common in literature. However, this is mostly true for general signal-agnostic sliding window approaches.

Chapter 6 stayed close to literature by using statistical features, but trying to use early fusion to merge them with pattern matching features obtained from DTW. A closed world experiment showed that, assuming the segmentation is known, a linear classifier with common statistical features has similar performance as the same classifier with pattern features. Moreover, combining both feature types yielded significantly better performance. At the same time, it was also shown that a naive attribution of a segment to the closest pattern's class was significantly inferior to using similarity to all class patterns as features in a classifier.

A more elaborate experiment also took into account the segmentation from continuous recordings. This is of vital importance for correct recognition. An energy-based segmentation approach with sliding windows assumed a different variance of activities, with small resting periods in between. One also has to take into account false detections belonging to the NULL class. They have to be rejected. To this end, a hierarchical framework was constructed. It consists of a random forest biased toward the NULL class, a closed world linear classifier and binary rejection classifiers for each class. Segmentation was shown to perform well, both in the detection of segments and in their overlap with the

ground truth. Classification was also shown to give good results, even taken into account missed, rejected and false detections.

Chapter 7 offered an alternative for Chapter 6. It strayed further from literature by focusing solely on pattern-based approaches without additional features. An introductory step detector on a public dataset showed how static patterns can be used in restricted settings. In order to allow some more flexible matching, CWT matching was introduced. Since this is largely dependent on amplitude, the convolution was replaced by correlation. This flexibility however comes with a price: as the specific example showed, much more false detections could be expected. Moreover, without further adaptations, it requires matching the pattern on multiple scales to all possible starting positions.

Therefore, a more complex approach with dynamic time warping exploiting multilinear data structure was developed for the aforementioned protocol setup. Two patient studies were merged for this study. The segmentation approach remained the same. Each segment was transformed to a tensor through matching it to all class patterns via DTW. Hence, the multilinear data structure has a time, channel and class dimension. Next, HODA applied a joint Tucker decomposition on all training tensors to obtain factor matrices and training core tensors. The latter contain the compressed class-discriminative information expressed in function of the subspaces defined by the factor matrices. Vectorizing them yielded the training features. A random forest including the NULL class performed the classification with rejection, this time in a single step.

A closed world experiment showed that HODA features obtained good classification accuracy and significantly outperform simple pattern features expressing the DTW similarity with the class patterns. Segmentation also performed well. However, missed, rejected and false detections led to a low final classification accuracy. This is at least partly due to the merging of the two datasets which have slightly different characteristics.

Still, when comparing both Chapters, it is not easy to decide overall whether patterns are superior to statistical features for recognition of transitory activities. The difference in dataset and classification approach makes it impossible to judge solely based on the large gap in performance. On the one hand, Chapter 6 indicates that similarity to all class patterns significantly improves statistical feature-based classification. On the other hand, Chapter 7 shows HODA features significantly outperform the pattern similarity features. Therefore, a combination of statistical and HODA features would be an interesting topic for future research.

9.1.3 Activity assessment

Activity recognition is primarily meant to serve as input for assessment. Currently patients come to the hospital for assessment of their disease status through medical examination e.g. via MRI or blood samples. On top of that, their functional capacity can be estimated. Typically this is done via the patient-reported BASFI questionnaire, but performance-based tests offer an objective complement, mostly restricted to the manual measurement of activity duration.

Chapter 8 first investigated if accelerometer features other than duration could be extracted to build a more insightful assessment measure. A first study extracted specific interpretable motor control parameters from patients and healthy controls performing typical flexion movements. Further investigation at the faculty of Rehabilitation showed the reliability of the features compared within a session and between sessions. A second study obtained duration and number of steps from hip fracture patients performing a TUG test. For both patients and healthy controls, the statistical evaluation showed that the extracted number of steps was not significantly different from the ground truth, but for the durations the slight difference was significant.

Then, the Chapter studied three ways to construct instrumented BASFI (iBASFI), an objective model of the patient-reported BASFI, a measure for functional capacity. A first approach with stepwise regression concluded that a single or two-feature linear model based on a single activity can explain more variance in the data than a duration model. This was expressed via the R^2_{adj} which were consistently higher for the non-duration features.
A second approach yielded three models based on the elastic net. A duration-only model once again proved worse than the model based on more complex features. However, in the RMSE-optimal models based on complex features, all but one durations had been included as well. This suggests that, although complex features can be used to replace duration in small models, its inclusion still yields added value if high-dimensional models are constructed.
Thirdly, an `lpICS` classification approximation of BASFI was created. Due to the small dataset size, extensive resampling was used to increase the reliability of the findings. Still, the average LoSo classification accuracy was only 64.3%. Also here, a duration parameter had been selected, stressing once more its importance. Despite the low accuracy, the three-point ICS score showed a clear correlation with the ground-truth BASFI scores.

The last study in the Chapter went beyond functional capacity. It is a comparison with the study that led to the ASDAS score, a widely accepted system for disease activity in axSpA. An ICS scoring system was trained on a dataset obtained from the University Hospitals with a thresholded PGA as class criterion. An average AUC of 0.67 ± 0.09 (`lpICS`) and 0.7 ± 0.13 (`enICS`) was obtained via a 10

fold cross-validation. Large standard deviations, particularly for the calibration slope imply the model is not very stable due to the limited size of the dataset. Nevertheless, the study confirmed the importance of the duration of morning stiffness and (to a lesser extent) CRP.

9.2 Contributions

The contributions of this thesis are aligned with the objectives presented in Chapter 1, Section 1.3.

Part I: Interval Coded Scoring

Contribution 1 Starting from earlier work, the ICS framework to infer interpretable models for binary classification from data with optional guidance via expert knowledge has been extended. Variable interactions have been added as well as CPLEX support to solve the underlying Linear Programming problem. The visualization module has been extended to support interactions. Furthermore, a unified representation of the model assessment also quantifies risk calibration quality. Chapter 3 covers this contribution.

Contribution 2 The basic ICS formulation suffers from non-uniqueness and limited scalability to high-dimensional datasets. Preselection has been added to lower the problem's complexity. Uniqueness has been investigated via a quadratic formulation. Alternatively, the newly developed `enICS` yields a strictly convex problem formulation and obtains an equally performant model compared to `lpICS` with a reduced time complexity, largely independent of the dimensionality of the problem. This is covered in Chapter 4.

Contribution 3 Interpretation comes presumably at the cost of reduced performance. ICS has been applied on several public datasets to validate its performance. A comparison with established machine learning techniques showed no significant difference in performance. This is discussed in Chapters 3 and 4.

Contribution 4 A Matlab toolbox has been implemented to facilitate the use of ICS algorithms in many fields. It is discussed in detail in Appendix A. It provides an easy interface for problem formulation, solving, expert intervention, visualization, assessment and application to new data. The algorithms implemented in the toolbox are discussed throughout Chapters 3 and 4.

Part II: Activity recognition and assessment

Contribution 5 Window-based statistical features are commonly used for long term monitoring. Yet, patterns are hypothesized to be more suitable for transitory activities. Chapter 6 showed that patterns and statistical features complement each other. An activity pipeline combining both proved significantly superior to approaches based on both modalities separately.

Contribution 6 The impact of complementing patterns with the multilinear structure of the data has been tested through the use of HODA. Although the newly developed segmentation and recognition pipeline outperformed simple pattern features, the final accuracy is too low for practical use. Chapter 7 reports this approach.

Contribution 7 Activity recognition has been complemented with assessment. Chapter 8 has listed two cases of parameter extraction and reported on their statistical significance. Furthermore, three proof-of-concept iBASFI systems have been developed. Finally, ICS was applied for comparison with ASDAS, yielding only tentative results due to the limited sample size.

Contribution 8 Clinical recordings from wearable sensors need to be segmented and annotated. A toolbox has been developed for easy customizable visualization, organization, annotation and labelling of the data. It has been implemented in Matlab and Python. The Matlab version also supports video synchronization and proof-of-concept detection of basic transitions between poses. This toolbox is presented in Appendix C.

9.3 Future research perspectives

The overview given above opens several possibilities for future research. On the one hand, alternatives for the given techniques could be used. On the other hand, the work could be continued and expanded. These future research perspectives are discussed here for both parts of this thesis.

9.3.1 Interval Coded Scoring

`enICS` interactions Section 4.1.3 in Chapter 4 described how `enICS` improved execution time but does not support interactions. Further research might find a new decomposition, similar to Dykstra, but compatible with SVEN. Alternatively, further studies in efficient ways to solve the constrained elastic net could also open new possibilities.

Other languages The current implementation of ICS is coded in Matlab. It is well suited for academic research, but many companies and clinical research institutes make use of R or Python instead. The latter have the advantage of being freely available. Matlab has the advantage of easy notation and good performance for matrix operations. However, the performance aspect is partly related to the underlying C-coded optimization libraries. These are also available for other programming languages such as R or Python. Hence, it is conceivable to translate ICS to other languages without performance penalty. Experiments in Python already lead to a crude core implementation able to interface the CPLEX optimization. The resulting preliminary implementation of `lpICS` proved to have roughly the same execution speed for the core algorithm as its Matlab counterpart. The further development of this implementation would widen the applicability of ICS.

Regression Many processes that could be modelled have a continuous output. A typical example is the BASFI questionnaire. ICS currently only supports binary classification labels. Therefore, regression would be a valuable ICS extension. The easiest way to accomplish this is via `enICS` which is already based on regression. Yet, this requires a reinterpretation of ICS's optimization and assessment measures as well as the visualization and the role of the risk profile.

Multi-class Another potential ICS extension is support for multiple classes. This arises for example in tumour classification in oncology. With the current implementation, one could train multiple independent scoring systems in a one versus all approach. A joint optimization could be beneficial, but leads to conceptual problems. For example, each class needs a separate risk profile. If a single score is used, these risk profiles will probably not (all) be monotonic which violates the current assumption of using a logistic function. Hence, first, a new concept for multi-class needs to be defined.

Validation and applications This thesis validates ICS on multiple public databases. It has also been applied on a large dataset in the context of a master's thesis. This yielded a scoring system for risk assessment for acquired long QT syndrome, tied to the occurrence of Torsade de Pointes and sudden cardiac death [88]. More and larger similar studies would be further proof of the viability, reliability and power of ICS. This validation is necessary in order for ICS to function as a general toolbox to generate scoring systems for use in any application field that requires interpretability.

Clinical validation and use The ICS validation study in this work was a comparison with established machine learning methods. It investigated the statistical significance of ICS's performance and calibration. However, it is even

more important to look at the clinical validity and significance of the results. For examples, do differences in ICS risk outcomes correspond to actual clinical differences? Moreover, feedback of the expert users, that is, the clinicians, is required to start a discussion that might lead to better design, more powerful models and easier adoption. So far, expert feedback was provided by engineers with medical background rather than medical doctors.

9.3.2 Activity recognition and assessment from wearable sensors

Scaling up The activity recognition approaches were tested on 28 to 39 patients. This is a relatively small group which does not capture the total variability in axSpA patients. Larger studies could be used to further validate the recognition approaches that have been developed.

Daily life recognition The approaches in this thesis can be used in a home environment but are still based on an activity protocol. Ideally, transitory activities would be measured in daily routine, extracting parameters throughout the day. This type of recognition has hardly been attempted in this manuscript. As a single example, a master thesis has been conducted to detect only sit-stand transitions in a home environment [169]. Future research should expand the algorithms in this work to function in a free-living environment.

Assessment studies Chapter 8 described parameter extraction and several approaches to iBASFI. These pilot studies should be repeated with larger populations to improve the statistical strength of the findings, particularly when considering the large variabilities. A possible larger study aim could include a web-based interface for data collection over a longer time period with automatic processing and visualization of the evolving statistics. The final goal would be the implementation of these algorithms and parameters for clinical use.

Real-time feedback The activity recognition and assessment techniques in this thesis have all been developed for remote monitoring, not for intervention. Another aim could be the development of similar methods for e.g. pose recognition and correction. This is particularly useful to extract slumping behaviour during sitting. Dual-sensor experiments have already been performed in collaboration with the Faculty of Rehabilitation Sciences to extract the orientation of the spinal region as a first step in that direction.

9.3.3 Other considerations

The following considerations are not a direct continuation or extension of the work in this thesis, but are important to take into account for the future.

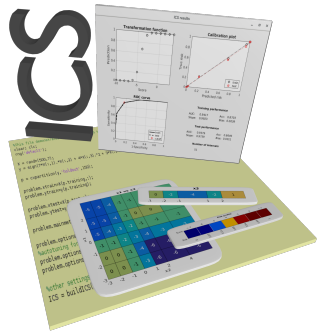
Sensor development Many algorithms are sensor-dependent, although this can often be remedied via a calibration procedure. As Chapter 7 showed, merging data from different sensors might nevertheless be a problem. Many sensors are commercially available, but are not always convenient for long term monitoring due to e.g. lost connections, a bulky encasing etc. A standardized sensor should be developed to remedy these issues and provide a reliable building block for a body area network with multiple sensors.

Interdisciplinarity The activity recognition and assessment approaches have been developed in close collaboration with researchers and practitioners from engineering and medical fields. This collaboration is of the utmost importance for future work. On the one hand, it ensures that new approaches are directly discussed and verified from several points of view. This guarantees actual (medical) problems are addressed, not merely technical abstract counterparts. On the other hand, the involvement of all stakeholders makes the concrete application and adoption of new methods on short term more likely.

Collaboration with industry Similar to the previous consideration, contact with industry might improve the focus of the research and increase the possibility of adoption in clinical practice.

Patient-centred design This is an aspect that has only be touched upon in this manuscript. Any system for monitoring and assessment should be convenient for the user, hence e.g. the single sensor armband. However, particularly when direct feedback is involved, the interface and communication style are very important as well. All these aspects of a system contribute to a patient's adherence to therapy and hence, the effectiveness of the monitoring or treatment. In the context of SPARKLE this has been studied by the eMedia Lab (Group T). It remains an important aspect to take into account for any future research direction.

Appendices



Appendix A

Interval Coded Scoring Toolbox

Chapters 3 and 4 discussed the Interval Coded Scoring data transformations, the two different models `lpICS` *and* `enICS`*, the use of preselection and also the general ICS algorithm. All functionality is implemented as a Matlab toolbox, which also provides an interface layer for easy access.*

This appendix describes the ICS toolbox. It explains how to install it, use it to effectively derive a scoring system in different stages, from the setup of a problem over running it automatically or interactively. It also discusses visualization, assessment and application of an obtained model on new data. This overview was previously published as part of the toolbox manual available at `https://www.esat.kuleuven.be/stadius/icstoolbox`*.*

A.1 Setup

A.1.1 Prerequisites

The ICS toolbox should be used with a Matlab version greater than or equal to 8.4 (release 2014b). It requires the *Optimization* and *Statistics (and Machine Learning)* toolboxes to be installed. Some functionality also needs external packages. The core functionality requires LS-SVMlab. Optionally, ICS can take advantage of the Support Vector Elastic Net (SVEN) and IBM CPLEX toolbox. You can add the toolboxes manually to the path or simply put them in ICS's main folder when running the setup file.

- **LS-SVMLab (required)** is a toolbox for solving Least-squares Support Vector Machines [158]. It is subject to the GNU GPL license and available at `http://www.esat.kuleuven.be/stadius/lssvmlab`. It is primarily used to perform variable preselection, but some of its functionality is also used elsewhere e.g. to tune hyperparameters.
- **IBM CPLEX (optional)** is an optimization suite with a Matlab interface [73]. It is freely available for academic use at `https://www-01.ibm.com/software/websphere/products/optimization/cplex-studio-community-edition`. Without CPLEX, the core optimization problem is solved using Matlab's Optimization toolbox. At the moment of writing, CPLEX's implementation is one or two orders of magnitude faster.
- **Support Vector Elastic Net (optional)**, abbreviated as SVEN, is a toolbox developed by Zhou et al [184]. It is freely available at `https://bitbucket.org/mlcircus/sven`. It can be used as a library to solve elastic net problems very efficiently by restating them as SVMs. This can be exploited in the `enICS` formulation discussed in Chapter 4.

A.1.2 Installation

The ICS paths can be properly set by running *setup.m*. The script will check for the presence of the three packages in the Matlab path by simply detecting their name or add them if they are in the current folder.

A.2 Setting up a problem

The ICS toolbox uses a struct *problem* as main carrier of information. Hence, while setting up a problem, one needs to set its initial fields. The following fields are mandatory:

- *xtrain*: the training data, with N observations as rows and N_d variables as columns.
- *ytrain*: the training labels for the binary classification problem as a $N \times 1$ vector of any two numerical values. The lowest value will always be used as the negative class, the highest as the positive (target) class.

All other fields are optional. Default values are used if they are undefined.

- *groups*: the effects to be considered in the model. They are specified as a cell array. E.g. {1, 2, [1, 2], 3 } defines main effects for variables 1, 2 and 3 to be considered, together with the interaction between variables 1 and 2.
  Default: all main effects and two-way interactions for `lpICS`, all main effects for `enICS`.

- *mainmethod*: the ICS algorithm used, either '*en*' or '*lp*'.
  Default: '*lp*'.

- *var*: struct with definition of all variables as continuous (field *cont*), ordinal (*ord*), binary (*bin*) or categorical (*cat*).
  Default: derived from the values.

- *varnames*: array with the names of the variables.
  Default: $\{v_1, \ldots, v_{N_d}\}$.

- *xtest*: the $N_2 \times N_d$ test data, with the same features as *xtrain*.
  Default: [ ].

- *ytest*: the $N_2 \times 1$ vector of test labels. Classes should be represented by the same two numerical values as in the training labels.
  Default: [ ].

- *options*: a struct with options fields, namely:

  - *show*: whether or not to show the final model.
    Default: true.
  - *preselect*: whether or not preselection should be applied.
    Default: false.
  - *maxbins*: the maximum number of intervals for any given variable. It can be defined as a vector of length N_d with a value for every variable. Alternatively, a single scalar can be used.
    Default: 10 for continuous and ordinal variables, 2 for binary variables and the number of categories in case of categorical data. Note that the number of intervals for binary and categorical data always reverts to default, even if another value is supplied.
  - *auto*: whether or not the automatic mode should be used for preselection and/or reweighting. A single boolean value or a boolean vector can be used as input.
    Default: [false, false].
  - *cutoff*: relative AUC threshold for automatic preselection and reweighting. A single value or a vector can be used as input.
    Default: [0.95, 0.95].

To train the model one simply needs to call the `buildICS` function with the *problem* as input. See the demos in Section A.7 for concrete examples.

A.3 Automatic or Interactive ICS

ICS performs iterative reweighting to simplify the model. In automatic mode, it selects the weight parameter a based on a cutoff on the AUC value. First, it obtains estimates for the impact of each parameter value based on 10-fold cross-validation on the training data. At the default values, it then chooses the weight that leads to an AUC of around 95% of the maximum AUC. However, in manual mode, the user can intervene at this point. The information obtained through cross-validation is presented to the user via the interface shown in Figure A.1. Subsequently, he or she can then make the trade-off between model complexity and performance.

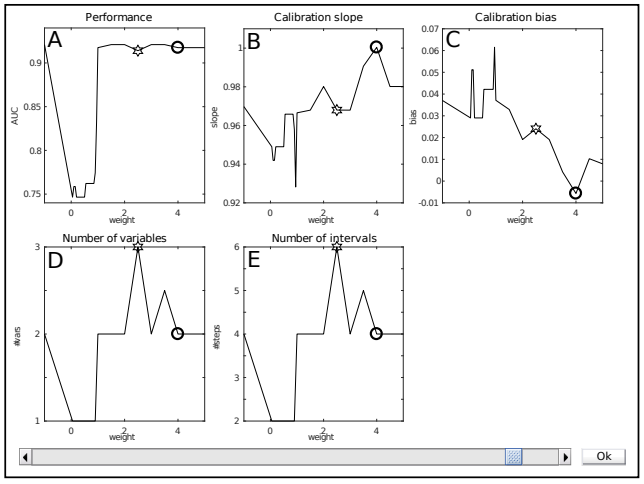


Figure A.1: The interactive weight selection window allows the user to intervene during iterative reweighting in the ICS core. It presents cross-validation results for the training data as a function of the selected weight parameter a. The model performance is expressed by the Area under the ROC curve (A), calibration is assessed using the slope and bias of the calibration curve (resp B, C). The complexity of the model is indicated by the number of variables (D) and intervals (E). The circle corresponds to the current selection, the hexagon to the solution used in automated reweighting. The selected weight can be adjusted by clicking in any of the graphs or by means of the slider at the bottom.

The Figure displays five measures of model performance. The first one (A, top left) represents the Area under the ROC curve. The second and third (B and C, resp. top middle and top right) refer to the calibration via the slope and bias of the calibration curve. This is a least squares linear fit of the model-based estimated risk against the empirical risk. Ideally this fit should have a slope of one and a bias (offset) of zero. The second row of the Figure shows two measures for model complexity: the number of selected variables (D) and the associated number of intervals in the model (E). The optimal model has as few variables and intervals as possible, while still obtaining good performance and calibration. Most often, a trade-off will have to be made.

Selection of a weight is possible by using the slider or by clicking in any of the subfigures. The choice can be confirmed by clicking the OK button or by a carriage return. The current selection is indicated by a circle, whilst the hexagon denotes the suggested weight based on the cutoff variable that would be used in automatic mode. One can also choose to perform no reweighting at all. This can be enforced by selecting a weight value of -1.

A.4 Preselection

Preselection can be used as a step to reduce the dimensionality of the problem before applying the actual ICS model. It involves a common Least Squares Support Vector Machine model to construct a new data set, followed by an elastic net for feature selection. The selection can be done automatically, using the cutoff on the AUC defined while setting up the model. Alternatively, the user can indicate the desired trade-off between performance and model simplicity. In that case, he or she is presented with a selection window as shown in Figure A.2.

The presented information is based on 5-fold cross-validation on the training set. The penalization weight λ allows to make a trade-off between performance and model complexity. The left side of the Figure shows the expected sparsity of the solution. By using the labels on the left, one can see that the sparsity increases for increasing values of λ. Equivalently, this can be expressed in terms of the variables kept in the model, presented on the right. The right part of the Figure shows the performance as a function of λ.

The proposed solution is indicated by a hexagon. The circle indicates the current selection. It can be changed via the slider or by clicking in one of the graphs. A final choice is made by either clicking the button or by a carriage return.

It is important to realize that this step does not show the final performance of the ICS model, but only a relative assessment of the current preselection step.

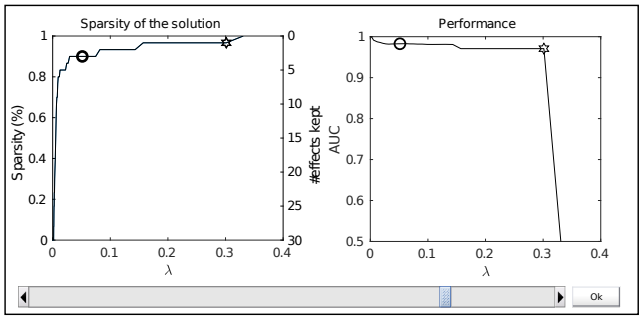


Figure A.2: The interactive preselection window contains estimates for the sparsity of the model (left) and its performance (right) based on cross-validation results on the training data. The circle shows the current selection, the hexagon the value that would be used in automated preselection. The value of the weight parameter lambda can be adjusted by clicking in one of the graphs or by means of the slider.

A.5 Visualization and Assessment

The training procedure yields an ICS model that can be used for further classification and study. By default, the visualization interface in Figure A.3a is shown at the end of the training. Alternatively, one can invoke it with the `showICS` function with the ICS model structure as only parameter. By visualizing the obtained model, that is, its selected effects and intervals, one could apply it manually on new data. Furthermore, the interface can also be used to assess the model performance.

A.5.1 Visualization

The interface lists all selected effects (main effects or interactions) in the listbox. One can look at the details of an effect by clicking the *Show selected* button. Alternatively, *Show all* displays all effects. The checkboxes allow to add the numerical scores or generate optimized grayscale instead of coloured representations (e.g. for printing). The bottom part of the window contains the color legend common to all score point representations. As an example, Figure A.3b displays a simple effect with highlighted numerical scores.

The model can be applied by looking at the contributions of all effects to obtain a score. In the case displayed here, if the *pulse_3_R* value is above 0.39, 1 point is added to the total score. The final score incorporating all effects corresponds to a risk. The risk profile is available via the *Show risk profile* button. An



Figure A.3: Visualization of the ICS model: (a) the main window lists the selected effects and allows to visualize them in more detail, (b) an example of an effect, (c) the risk profile.

example is shown in Figure A.3c. At the top, it lists the scores, at the bottom, the associated predicted risks. For example, an observation with a score value of 0 has a risk of 23% of belonging to the target (positive) class.

A.5.2 Assessment

The *Model assessment* button in the main interface (Figure A.3a) invokes the assessment window displayed in Figure A.4. It reports performance characteristics of the model and is divided in four quadrants.

- The top left is the transformation function from scores to predicted risks, that is, another way to represent the risk profile.
- At the top right one finds the calibration plot, a linear fit of predicted and empirical risk with the plotted risk points corresponding to all possible score outcomes. If test data was provided at the start, one can observe two sets of points, for training (black) and test (red) set.
- The bottom left displays the ROC curve for both training (full line) and test (dashed line) data. It quantifies the classification performance of the model. A summarizing characteristic is the Area Under the Curve (AUC). The closer it is to 1, the better a model's performance, with 0.5 (the main diagonal) corresponding to chance level. The cutoff value, shown as a red circle, corresponds to the ROC point closest to the (ideal) upper-left corner of the graph.
- The last quadrant provides several summarizing statistics for training and test data (if available). They are summarized in Table A.1.

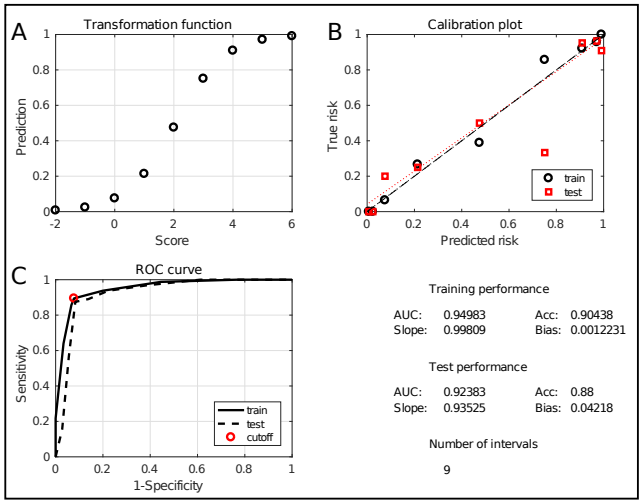


Figure A.4: The model assessment interface. It displays the mapping of scores to risks (top left), the calibration curve of the model (top right), its ROC curve (bottom left) and a numerical evaluation of model performance and calibration, as well as its complexity expressed as the number of intervals (bottom right).

Table A.1: ICS performance indicators.

Parameter	Optimal	Explanation
AUC	1	The area under the ROC curve
Acc	1	Class accuracy (cutoff at ROC top left)
Slope	1	Slope of the calibration curve
Bias	0	Intercept of the calibration curve
Nr of intervals		The total number of intervals in the model.

In case no test data was provided, the test performance indicators will be omitted in Figure A.4 and no test calibration plot or test ROC curve will be shown. Alternatively, even if no test data was supplied in the problem formulation stage, it is always possible to pass new test results (scores and ground truth labels) to be displayed as test data. These results can be obtained by applying the trained model on new data as discussed in the next part.

A.6 Applying a model on new data

Once a model has been obtained through training, it can be used either manually, via the visualization, or automatically, by supplying both the model and the test data as arguments of the `applyICS` function. In the latter case, the function yields predicted labels, obtained scores and associated risks for every new observation.

If one knows the ground truth labels *ytest2*, one can use `showICS` with additional parameters to assess the model's performance on this new data. By calling `showICS`(*model*, *scores*, *ytest2*), with *scores* obtained from `applyICS` and $ytest2_i \in \{-1, 1\}$. The test data performance assessment contained in the model (see Figure A.4) is now computed with respect to the new data.

A.7 Demos

The toolbox contains three demo scripts to showcase its use. It is a quick way to experiment with the settings and observe their effects.

A.7.1 Fisher's Iris dataset

Fisher's Iris dataset contains four characteristics of three types of iris flowers (setosa, virginica and versicolor): the length and width of the petals and sepals. This dataset with 150 observations is directly available in Matlab. A minimal example of how to build an ICS model for this dataset is shown in Snippet A.1. A more extended version is available in the Toolbox's demo folder as `irisdemo.m`.

Snippet A.1: A minimal ICS setup

```
load fisheriris; %built-in dataset containing 'meas' and 'species'

problem.xtrain = meas; %we use the entire set for training
problem.ytrain = 2*strcmp(species,'virginica')-1;

ICS = buildICS(problem);
```

The code snippet above first loads the data and then initializes the problem fields with the data and labels ('virginica' vs 'setosa/versicolor'). These are the only fields required. All other settings are initialized to their default values. Of course, this has some drawbacks. For example, no test data has been assigned, so the performance can only be judged by cross-validation. Furthermore, the default model considers all main effects and interactions, a brute-force approach

that is mostly suitable only if nothing is known about the data, but quickly leads to long execution time. Despite this drawback, it offers the advantage of completeness. Moreover, user interaction is enabled for the reweighting stage, which allows for a quick exploration of possible models. Finally, be aware that even this small example might take some time with Matlab's builtin solver due to the interactions. IBM's CPLEX significantly speeds up its execution (on our 3.6GHz system from 4min55s to 2min32s).

Overall though, this dataset is very simple and the impact of the user's choices on performance is therefore rather limited: only one or two main effect(s) remain.

A.7.2 Ionosphere dataset

This dataset contains 351 instances of 34 antenna readings, classified as either 'good' or 'bad'. If we only consider main effects, we could easily use `enICS`, as show in Snippet A.2. It can also be found in the demo folder as `ionospheredemo.m`.

Snippet A.2: Using `enICS` on the *ionosphere* dataset

```
load ionosphere; %built-in dataset containing 'X' and 'Y'
y = 2*strcmp(Y,'g')-1;
p = cvpartition(y, 'holdout', 100); %stratified sampling

problem.xtrain = X(p.training,:);
problem.ytrain = y(p.training);
problem.xtest = X(p.test,:);
problem.ytest = y(p.test);

problem.mainmethod = 'en'; %enICS
problem.options.maxbins = 7;
problem.option.auto = true;

ICS = buildICS(problem);
```

Some more concepts are shown here. We introduce stratified sampling to create a training and test set. Then, the `enICS` method is selected. It will automatically disable interactions. Finally, the maximum number of intervals is set to 7 instead of the standard 10 and automatic tuning of the reweighting is enabled.

A.7.3 Synthetic dataset

This demo uses synthetic data to showcase the use of preselection and discovering interactions. It is outlined in Snippet A.3 and can also be found in the demo folder as `syntheticdemo.m`. The data was previously discussed in Chapter 3,

Snippet A.3: Using ICS on synthetic data

```
%generate random data, binary labels and partitions
X = randn(500,7);
y = sign(7*X(:,1).*X(:,2) + 4*X(:,3).^2 + 3*X(:,4) - 3 );

p = cvpartition(y, 'holdout', 100); %stratified sampling

problem.xtrain = X(p.training,:);
problem.ytrain = y(p.training);
problem.xtest = X(p.test,:);
problem.ytest = y(p.test);

problem.options.preselect = true;
problem.options.maxbins = 10;
problem.option.auto = true;

ICS = buildICS(problem);
```

The Snippet starts with the generation of random data and the application of a sign function to generate binary labels. The partitioning of the data is the same as in Snippet A.2. Afterwards however, the options differ. Line 12 enables preselection, other options are similar as before. Other settings are implicit. Recall that the default algorithm is `lpICS` and the default grouping considers all main effects and two-way interactions.

In the result, we expect to find an interaction between variable 1 and 2, a quadratic effect of variable 3 and a linear effect of variable 4. Other variables should be discarded since they do not contribute to the solution. The parameters presented here obtain a result similar to the one shown in Figure 3.2 in Chapter 3. However, the model can be further simplified by changing the tuning. For example, one could set *problem.options.auto = [true, false]*, implying automatic tuning for the preselection and manual tuning for `lpICS` reweighting.

A.8 Implementation details

This section alphabetically lists all functions in the toolbox with a short description, grouped as top-level, visualization or core functions. This listing is

for the sake of completeness. For an average user, the top-level functions and `showICS` are sufficient. Hence, for these functions, the inputs and outputs are described in more detail.

A.8.1 Top-level functions

`applyICS` - applies an existing ICS model on new data

In	*ICS*	struct, valid ICS model
	x	data, same dimensionality as the training set
Out	*pred*	predicted labels for the new data $\in [-1, 1]$
	scores	ICS scores for the new data
	risks	the associated estimated risks

`buildICS` - top level function and main access point to its functionality

In	*problem*	struct containing all information of an ICS problem
Out	*ICS*	struct, a trained ICS model
	problem	updated problem structure
	hist	struct with information about the training flow

`setup` - script to check ICS requirements and initialize the paths if necessary

A.8.2 Visualization functions

`dispGroupScore` - displays a single effect of an ICS scoring system (main effect or interaction)

`getTextColor` - calculates text colour (black or white) based on YIQ value

`plotColorLegend` - plots colour legend at the bottom of the interface

`plotResults` - launches the model assessment interface

`plotRiskProfile` - generates the risk profile window

`setupClrInfo` - computes the colour values based on the range of scores

`showICS` - launches the visualization interface

In	*ICS*	struct, valid ICS model
	scores	predicted scores obtained with `applyICS` (optional)
	ytest	ground truth labels for the scores $\in \{-1, 1\}$ (optional)
Out	/	

A.8.3 Core functions

`calcPerf` - calculates training and test performance measures

`chooseWeight` - launches the interface for weight selection during iterative reweighting

`convert2FeatureSpace_en` - converts original data to the binary feature space used by `enICS`

`convert2FeatureSpace_lp` - converts original data to the binary feature space used by `lpICS`

`convertW2FuncForms` - splits a full coefficient vector in the effects' subcomponents, reshaping them to matrices if applicable

`convertWshape` - simplified version of `convertW2FuncForms` for subcases

`createVectArray` - creates the struct with information about variables and thresholds.

`csa` - performs simulated annealing

`generate_block` - generates a block of the difference matrix for `lpICS`

`get_performance_ics_class` - computes k-fold performance metrics

`get_pred` - computes predictions based on training and test scores

`ics_class_en` - ICS core function for `enICS`

`ics_class_lp` - ICS core function for `lpICS`

`makeICS` - contains the main ICS flow to be executed on a problem. Called by buildICS

`optimize_feature_sel` - computes kFold influence of weighting

`post_ics_class` - postprocessing of the ICS model

`pre_ics_class` - preprocessing of the model

`preselectICS` - contains the preselection procedure

`preselectICSgui` - launches the interface for manual preselection weight selection

`refold` - reshapes a vector-unfolded matrix or tensor to its original shape

`reweigh` - iterative reweighting procedure

`simplify` - simplifies a finished ICS model e.g. by interval reduction

`unfold` - reshapes a matrix or tensor to a vector

A.8.4 Problem parameters

Problem is a struct to encapsulate all aspects of an ICS problem. It is the input for buildICS which performs some checks, preselection, training (via `makeICS`) and eventually visualization of the final ICS model. For more information, see Section A.2.

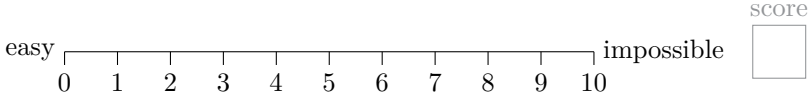
Appendix B

Bath Ankylosing Spondylitis Functional Index

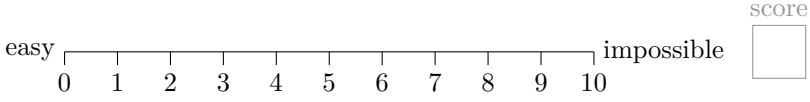
This appendix contains a full version of the BASFI questionnaire as currently used in clinical practice [35], distributed by the Ankylosing Spondylitis International Federation (ASIF) [9].

Please draw a mark on each line below to indicate your level of ability with each of the following tasks.

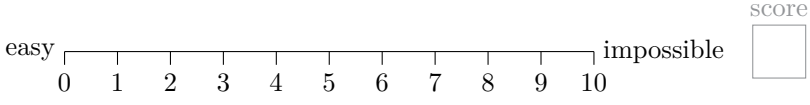
1. Putting on socks or tights without help or aids (e.g. sock aid).



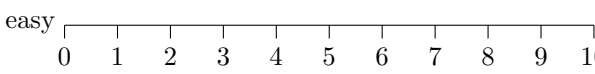
2. Bending forward from the waist to pick up a pen from the floor without an aid.



3. Reaching up to a high shelf without help or aids (e.g. helping hand).

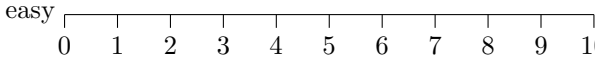


4. Getting up out of an armless dining room chair without your hands or any other help.



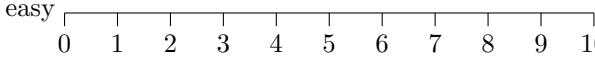
score ☐

5. Getting up off the floor without help from lying on your back.



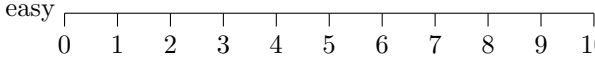
score ☐

6. Standing unsupported for 10 minutes without discomfort.



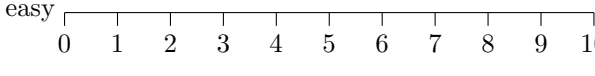
score ☐

7. Climbing 12-15 steps without using a handrail or walking aid, one foot on each step.



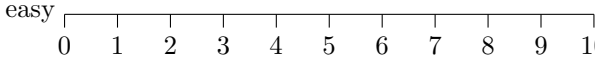
score ☐

8. Looking over your shoulder without turning your body.



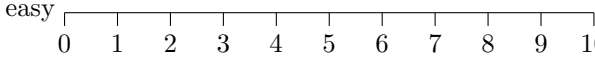
score ☐

9. Doing physically demanding activities (e.g. physiotherapy exercises, gardening or sports).



score ☐

10. Doing a full day's activities whether it be at home or at work.



score ☐

(sum of answers 1 to 10 divided by 10) BASFI = ☐

Appendix C

SegmentLab toolbox

In the course of this thesis, several datasets have been measured in the movement laboratory, at the Division of Rheumatology in the University Hospitals or in subjects' home. The development of suitable algorithms for activity recognition and assessment, explained in Chapters 6, 7 and 8 requires the availability of ground truth segmentation and activity labels. Moreover, the algorithms were implemented in Matlab, an unfamiliar program for the labelling experts.

With this aim in mind segmentLab, a Matlab/Python toolbox, has been developed. it supports the import of data from various types of text files, several data visualization aids, segmentation and labelling options. Furthermore, data-video synchronization has been implemented in Matlab. Finally, the Matlab version also supports some basic automatic activity recognition approaches. It has been tested in version 2016b.

In what follows, the Matlab toolbox will be discussed from a user's point of view. First, it summarizes the interface, followed by a discussion on data loading and representation. Then, it continues by explaining the various visualization tools. Next, it instructs on how to annotate the visualized data before discussing some functionality extensions. The final Section of this Appendix introduces the Python equivalent, briefly highlighting the differences for the user. Neither the Matlab nor the Python version of the toolbox has been released for public use.

C.1 A brief overview of the interface

To get started, simply execute the file *segmentLab.m*. This triggers the interface shown in Figure C.1. It consists of five parts, from top to bottom. On top,

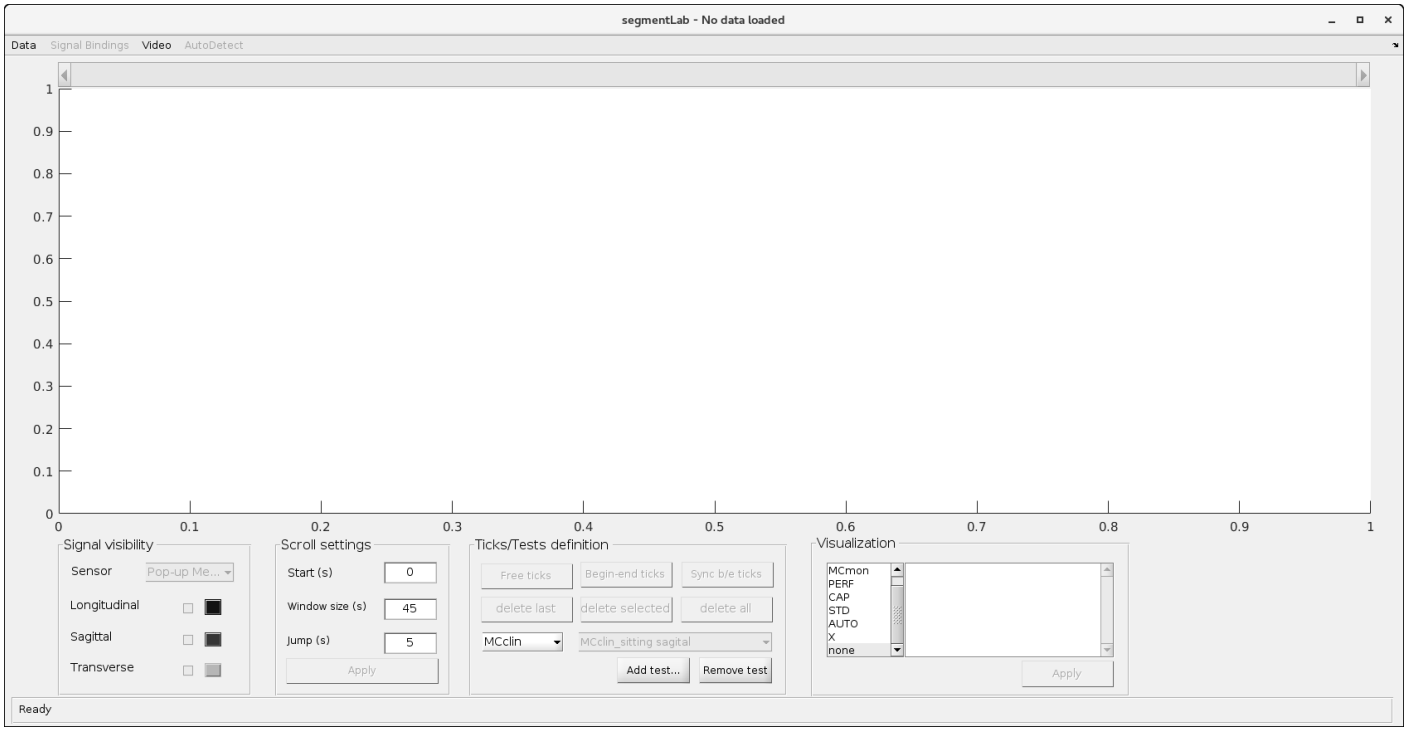


Figure C.1: The starting window of segmentLab.

the window title bar indicates the current file. In case of this Figure, no data has been loaded yet. Below is the menu bar. All menu items will be discussed throughout this Appendix. The next layer is the graphical axis with a scrollbar. Here, all signals and annotations will be visualized. Underneath one can see all the controls to select which signals should be shown, how large the visualization window should be, annotation tools and a way to limit which tests should be visualized to avoid clutter. They are organized in four panels. Finally, the status bar at the bottom gives status reports during the execution of certain algorithms.

C.2 Data representation

A Matlab data file can be loaded via the menu: *Data* → *Open*. This will trigger a file selection window. In a data file, data is organized in *sensors*. This should not be understood as one physical device, but as a single triaxial sensing unit. For example, an IMU can be represented as three sensors: triaxial accelerometry, triaxial gyroscope and triaxial magnetometer. A data file consists of minimally three variables, but, as data is being segmented or default options are changed, other variables can be added. All possible variables in a data file can be listed as follows, assuming one has N sensors:

- **data** (required) The data matrix. It will always have a $3N$ columns, since data is assumed to come from triaxial sensors. Uni- or biaxial sensors are represented via the addition of columns of NaN (not a number) values. The data is assumed to be sampled at an exact sampling rate, without any delays.
- **fs** (required) The sampling rate of the data.
- **ids** (required) A cell array of length $3N$ with a string identifier of each sensor, given for each axis.
- **annots** Cell array of textual annotations that should be shown in the graphical interface.
- **eventss** Array of x coordinates, expressed in seconds, corresponding to the textual annotations.
- **sensorids** A cell array of length N containing the same information as ids, but with a single entry for each sensor for convenience.
- **sensorbinding** A matrix consisting of three rows and N columns. It indicates which axis of each sensor is longitudinal, sagittal or transverse.

- **clrs** A matrix of three columns and $3N$ rows. It contains an RGB colour triplet for each axis to indicate the colour of its visualization. At first loading, this variable is initialized using the configuration file *clrs.mat*
- **tests** A struct array containing the annotation of all tests. Each struct has the following fields:
  - **name** The name of the test. All available tests are listed in the configuration file *testnames.mat*.
  - **beginend** A matrix with two rows containing the x coordinates (in seconds) for the begin (first row) and end (second row) of a segmented region. Every column defines a new segment.
  - **ticks** A vector of x coordinates (in seconds) for single annotations, when no region is needed.

The approach of loading the data and its annotations assumes a Matlab file is available. This will mostly not be the case, since IMUs often export data in a raw text format. In that case, a Matlab file can be constructed manually or through conversion. The latter is accessible via the menu *Data→Convert from...*

C.2.1 Data conversion

Converting data can be done from two types of files: excel workbooks or text files. In both cases, a file layout configuration has to be provided first. The menu item leads to the interface shown in Figure C.2. It has buttons to build, save or load configuration as well as two conversion buttons, one for Excel and one for text files.

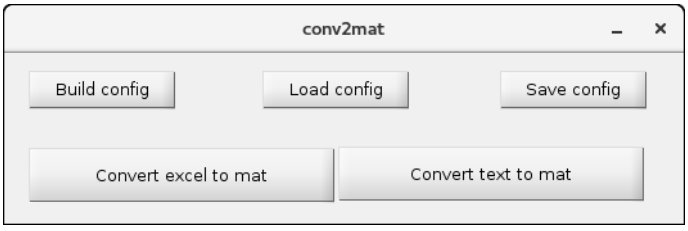


Figure C.2: Interface to convert Excel or text files to segmentLab-readable Matlab files.

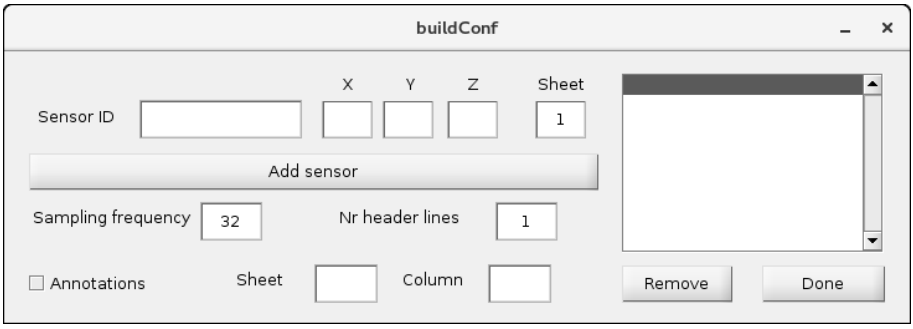


Figure C.3: Interface to describe a file layout configuration.

C.2.1.1 File layout configurations

Any file, Excel or text-based, is supposed to consist of rows and columns. The file layout configuration indicates what every column represents and which ones are of interest, whether the file has any header lines etc. This can all be indicated via the interface shown in Figure C.3.

The top left of the Figure is used to define new sensors, that is, sensor units. A user has to provide a sensor identifier (e.g. 'acc', 'gyr' or 'mag'), the columns in the file that correspond to the X, Y and Z axis and the workmap sheet. The sheet and any column numbers can be left out if not applicable e.g. in case of text files (no sheets) or biaxial sensors (no Z axis). Then, the sensor is added via the *Add sensor* button. It will appear in the listbox on the right and can be removed if needed using the button underneath it.

Continuing in this way one can define several sensors. For example, columns three to five could contain accelerometry data, columns six to eight of the same file gyroscope data etc. Nevertheless, a single file only has a single sampling rate and number of header lines. These are indicated in the text boxes below with the respective labels. Finally, a data file can also contain annotations meant for the *eventss* and *annots* variables defined earlier. In this case, the checkbox should be ticked and the sheet (if applicable) and column number should be filled out.

Finally, once everything is finished the *Done* button in the lower right of the Figure confirms the configuration and closes the window.

Back in Figure C.2, one can save the configuration for future use of load one instead of constructing it by the respective buttons.

C.2.1.2 Converting Excel files

The conversion of an Excel file (or 'workbook') can be executed via the respective button in Figure C.2. Importing an Excel file is only possible on a Windows system which has Microsoft Excel installed since the toolbox uses the ActiveX server. Support for Mac has not been tested. Note that you can circumvent the dependence on Excel by exporting data from a workbook to a text file using any compatible workbook program.

The configuration file assumes each sheet in the workbook has the same sampling rate and the same number of headers. Moreover, the sampling rate is supposed to be exact: no timing information available in the worksheets will be used.

The final output is a Matlab file which is saved through a Save file dialog. This can subsequently be loaded via the menu command *Data→Load* in the main interface (Figure C.1).

C.2.1.3 Converting text files

A more elaborate conversion can be performed via the *Convert text to mat* button in Figure C.2. Where Excel supports different sheets, text conversion supports merging multiple text files. All files are assumed to be identical, as defined by the file layout configuration. For example, all files contain accelerometer and gyroscope data in the same columns, they have the same headers and units etc. This is a common situation when data has been captured on multiple devices in a (wireless) body area network.

The selection of one or multiple files for conversion happens through a new window which allows you to select a target directory. Hence, all files to be converted jointly should be in the same directory. Then, the supported files in this directory are listed. Supported extensions include *txt*, *csv* and *dat*. The user can select one or more files for conversion.

In a next step, the interface will prompt the user for prefixes for all selected files. Imagine a configuration file with two sensors 'acc' and 'mag'. Imagine further two files corresponding to the IMU on the arm and on the chest which the user denotes with prefixes 'arm' and 'leg'. Then, the final lists of four sensors in the application will be listed with the identifiers 'arm_acc', 'arm_mag', 'leg_acc' and 'leg_mag' to differentiate them.

Using multiple files from multiple devices raises the problem of synchronization. SegmentLab assumes the IMUs have a synchronized clock. The first column of their output text files should contain the timestamps. All string formats recognized by Matlab's `datestr` are supported, as well as a numeric representation in milliseconds. The timestamps allows each file to be resampled

with the given sampling frequency. The latest start time of any of the IMUs involved is taken as time zero, any data measured before is discarded. In contrast, for every IMU, all data at the end is kept, even if some others have already been switched off. The data for the sensors related to the desactivated IMUs are replaced by NaNs.

The final output of the procedure is a single data matrix containing the measurements of all files, synchronized and sampled at a uniform sampling rate. At this point, a dialog window will prompt the user for a filename to save the generated Matlab file. Next, it can be loaded via the menu of the main interface, identical as for the Excel conversion.

C.3 Visualization

The visualization of data and segmentations can be controlled via the panels on the bottom of Figure C.1. They are repeated in an active state in Figure C.4.

Displaying signals Once data has been loaded the desired signals corresponding to the coordinate axes of the desired sensors can be visualized. By default, the sensor identifiers supplied in the process described above are used as sensor. In Figure C.4, the left panel labelled *Signal visibility* shows a dropdown list of all available sensors. For each sensor, it allows to check the signals one wants to show: the longitudinal, transverse or sagittal. These names are used since they are more informative than the X, Y and Z. The sensor names and default binding of coordinate axes to the anatomical labels can be changed if desired, as explained in the next paragraph. Each signal has its own colour to identify it with in the graph above. The colour can be changed by clicking the coloured square next to the anatomical label, prompting a colour selection window to appear.
The coloured signals are normally shown in the graph part of Figure C.1. The

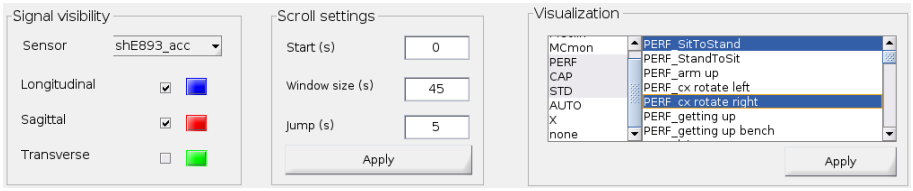


Figure C.4: The signal visibility panel, the scroll settings panel and the (test) visualization panel. The first controls which signals should be shown, the second defines the viewport of the graphs and the third indicates which test segmentations should be displayed.

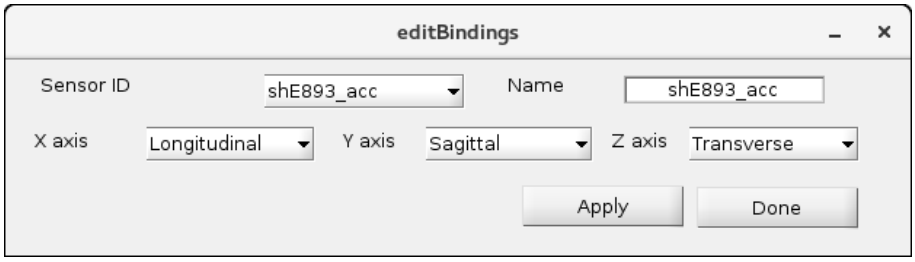


Figure C.5: The interface to edit signal names and bindings.

scrollbar on top simplifies signal exploration. Apart from by their colour, signals can be identified by clicking on them. This will render a tooltip with the name of the sensor and the axis. The option is mostly useful when several sensors are shown at the same time, such that not all signal colours are visible.

Changing default binding settings By default, the X axis is dubbed longitudinal, the Y axis sagittal and the Z axis transverse. Furthermore, the sensor identifiers are used in the dropdown box. It is possible to change the binding and the name using the menu command *Signal Bindings→Edit*. This spawns the window shown in Figure C.5. A sensor can be selected via its identifier in the dropdown list. Then, for that sensor, the name can be changed via the text field. The binding of coordinate axes to anatomical axis can also be defined. Note that the changes are only saved when clicking the *Apply* button before selecting the next sensor identifier. An error dialog will appear if an anatomical axis has been selected multiple times.
Clicking the *Done* button closes the interface and applies its settings in segmentLab. Consequently, the changes can be exported for future use with the menu command *Signal Bindings→Save as default*. This creates a file *defaultbinding.mat*. This default can be loaded again via the menu *Signal Bindings→Load default*. However, it can only be applied to the exact same setup: the same sensors in the same order. Hence, it might serve to restore the bindings in a segmentation session on a single file after they have been changed or when multiple subjects have been measured with the same sensor setup, provided the sensor identifiers match.

Scroll settings Figure C.4 also shows the *Scroll settings* panel next to the signal selection panel. It makes it possible to quickly define a view window on the graph by supplying the start of the window and its length in the text fields. Moreover, the step size of the scrollbar when using the arrows can also be set in this settings panel. The changes only apply when clicking the *Apply* button.

Displaying tests The last visualization settings can be applied in the panel *Visualization*. It refers to the visibility of the segmented tests. Tests are divided in fixed categories displayed in the listbox in the left part of the panel. Multiple categories can be selected. All tests in the selected categories are displayed in the right listbox. Once again, multiple selections are possible. Clicking the *Apply* button fixes the tests for display in the graph. Single ticks are rendered as vertical lines, segments as gray-shaded rectangles. They can be identified by the tooltip content when clicking on them.
None is a special category. On selecting it, even in combination with other categories, no tests are displayed. Hence, selecting *None* and clicking *Apply* is a shortcut to hide the visualization of all tests.
An example of a fully functional segmentLab with signals and segmented tests can be found in Figure C.7, which is used as an example for video synchronization discussed further.

C.4 Test annotations

The *Ticks/Tests definition* panel, visualized in Figure C.6, is the interface to segment and label activities. Which activity (or 'test') is being segmented is selected via the two dropdown lists at the bottom. The left-most selects the category, the right-most the specific test. New tests can be added or existing tests can be removed from the currently selected category via the corresponding buttons. Also, note that the currently selected test is always visualized, as if it was selected in the Visualization panel discussed in the previous section.

SegmentLab supports two types of segmentations. *Free ticks*, placed via the respective button in the top left of the panel, pinpoint a single time point. When visualized, they appear as vertical lines. *Begin-end ticks* represent regions. When segmenting, one simply clicks once for the beginning and again for the end of a region. For both free and begin-end ticks, clicking the button enters a segmentation loop: one can keep on segmenting without clicking the button every time. To exit this modus, use the right mouse button.

The second row of buttons deals with deletion of ticks or regions. The functionality is self-explanatory, but take into account that they work with respect to the currently selected test. For example: delete the lastly indicated region or free tick.

The *Sync b/e ticks* button will be covered when discussing video synchronization in the next Section.

The segmentations can be saved, together with the signal bindings, names and colours, via the menu *Data→Save*.

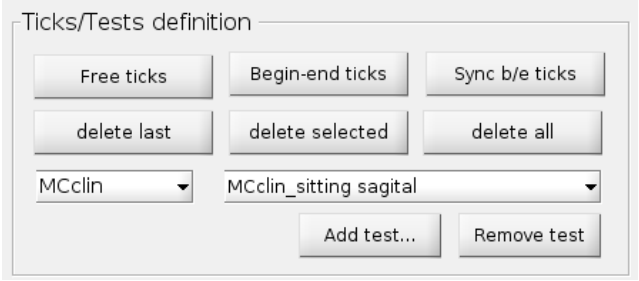


Figure C.6: The segmentLab panel to segment and label activities.

C.5 Extensions

All functionality discussed up until now suffices for data visualization and segmentation. To improve the ease and accuracy of the segmentation, video synchronization has been implemented. Moreover, a basic form of activity recognition is also available.

C.5.1 Video synchronization

A video can be loaded via the menu command *Video→Open*. The supported video types depend on the Matlab version and operation system support. Selecting an unsupported file format will raise an error. If successful, loading a video starts a new *Video* panel, visible in the bottom right of Figure C.7. The graph area will be shortened as well to make space for the video itself.

The video can be closed again via the *Video* menu. This menu also gives the possibility to rotate the video counterclockwise by 90, 180 or 270 degrees. Playing the video can be done using the *Play* and *Stop* buttons, as well as the scrollbar on top.

Of course, the video is not automatically synchronized to the loaded signal data. Synchronizing requires two steps.

1. Move the video toward a significant scene e.g. when a subject starts sitting down. Use the menu: *Video→Mark video sync*. The time point in the video is stored and mentioned in the lower right corner of the panel, as can be seen in Figure C.7.

2. Make sure the signal part representing the same movement is visible in the graph. Then, use the menu: *Video→Mark data sync*. A crosshair

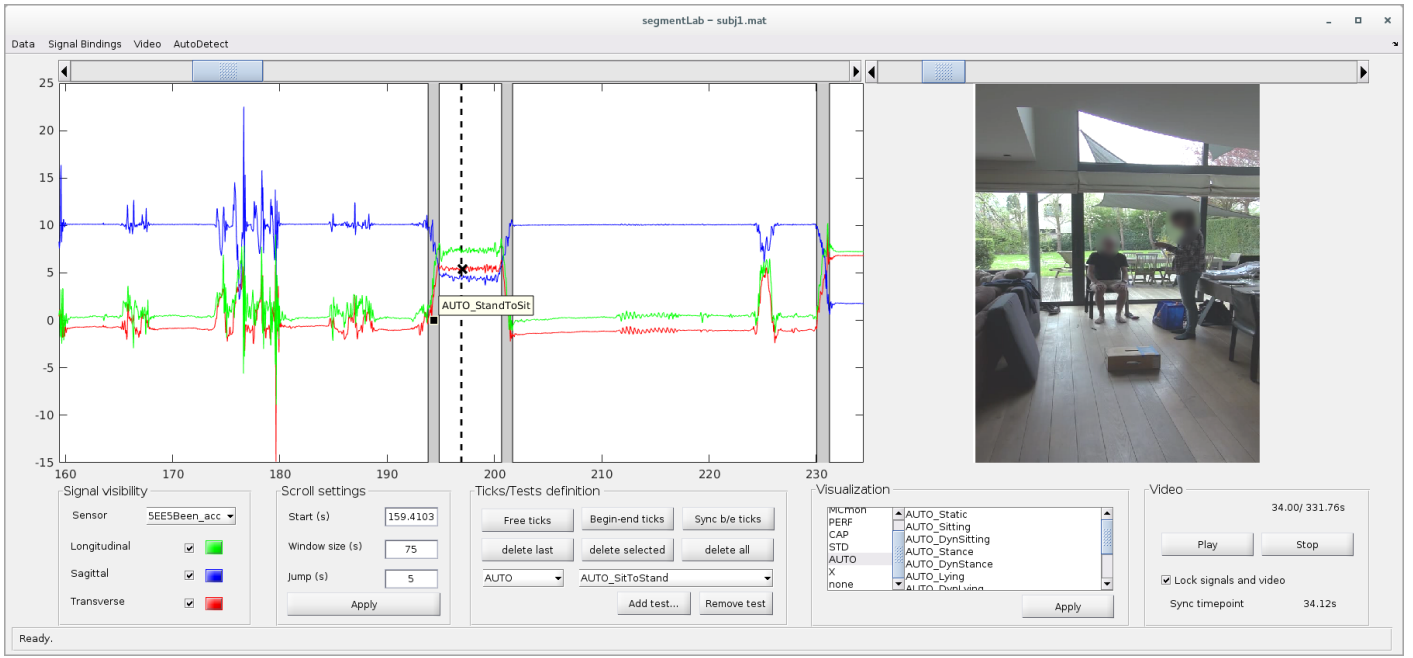


Figure C.7: A fully functional example of segmentLab including video synchronization and automatically recognized activities.

cursor helps to pinpoint an exact time point which is consequently marked with a black **X**.

The order of these steps is not important. Also, any of the two steps can be repeated to refine the synchronization.

A user can still browse the video and graphs freely, independently from one another. However, he or she can lock the graph and video with the *Lock signals and video* checkbox in the *Video* panel. In Figure C.7, this box has been ticked. When playing the video the corresponding time point in the data is marked with a thick dashed vertical line, called a syncline. The viewport of the graphs automatically moves together with the video.

This locking state between video and signals can be exploited for a precise segmentation. Going back to the *Ticks/Tests definition* panel e.g. in Figure C.4, a *Sync b/e ticks* button can be seen. It segments regions by defining begin and end ticks, just as the *Begin-end ticks* button discussed earlier, but its working differs slightly. When clicked for the first time, it marks the current position of the syncline (if one is available) as the start of a region. When clicked again, it marks the end of the same region on the (changed) position of the syncline. In contrast to the other segmentation buttons, start and end are marked by the button, not by mouse clicks on the required positions due to the fact that this position is defined by the syncline. This approach allows to simply select the test one wants to segment, let the video play and click the button twice at the appropriate times for start and end, as derived from the video.

C.5.2 Activity recognition

A last extension is a basic implementation of recognition of transitory activities. It can be triggered using the menu command: *AutoDetect→Pose/Pose transitions*. The process is guided via messages in the status bar and proceeds in five steps. Full mathematical details are left out.

1. As a first step, the user is requested to mark a sitting and standing region, similarly to begin-end tick segmentation, but right-clicking to exit the segmentation is not needed or possible. These regions will be used for data recalibration.

2. Next, the recognition can only be executed if two accelerometers can be detected: one at the (upper) leg and one at the back, e.g. around the third thoracic vertebra (T3). This assumes the user named the sensors correspondingly. For the leg, the sensor name should contain both 'acc' and 'leg' (or 'been', in Dutch). For the thoracic sensor, the name should

contain both ‘acc’ and ’T3’. The matching is case insensitive. If these names are not detected, the user is presented a list of sensors to pick them manually.

3. The indicated sitting and standing regions of the two sensors allow to calibrate the data in an anatomical coordinate system. For the leg sensor, gravitation during standing and sitting yields two orthogonal axes, complemented by their cross-product.
   For the back sensor, the gravitation in stance yields one axis. Another one is derived as the axis of greatest variability in the plane perpendicular to this gravity by means of Principal Component Analysis (PCA). The last back axis is the cross-product of the two others.

4. Once the data has been transformed to the new anatomical coordinate systems static and dynamic regions are separated based on the energy segmentation approach discussed in Chapters 6 and 7.

5. The next step classifies standing, sitting and lying poses based on rules for the orientation of the axes of the leg and back sensor. E.g. when standing, gravity of both sensors points downward, along the anatomically defined axis.

6. Finally, transitions are derived as the dynamic regions between identified poses.

The automatically detected activities can be visualized via the *Visualization* panel. They are listed under the category *AUTO*. The recognized activities can be categorized as static, dynamic or transitory, although they are presented in a single list in segmentLab.

- **Static**: contains *Static* (all static areas), *Lying*, *Sitting*, *Stance*.
- **Dynamic**: contains *DynLying*, *DynSitting* and *DynStance*. Here, movement has been detected for some sensors, but the main pose is still known. For example, walking is classified as *DynStance*.
- **Transitory**: contains *Getup*, *LieDown*, *SitToStand* and *StandToSit*.

The visualized *AUTO* tests behave identically to any other tests. For example, a tooltip is shown when they are selected. An example of an automatically detected *StandToSit* test is highlighted in Figure C.7.

C.6 Python implementation

A Python version of segmentLab has been implemented to circumvent the severe limitation of the dependency on Matlab. The interface looks and functions almost exactly the same. Video synchronization and activity recognition are not implemented at the moment. Also, the menu bar contains an *About* item triggering a small information window. This Section first discusses the package requirements followed by loading, saving and importing data. Then, signals and annotations are covered.

C.6.1 Package requirements

SegmentLab has been built using Python 3.4.5, combined with a Qt v5 graphical interface. Tested on Linux and Windows, it should in principle be perfectly cross-platform. It requires the packages listed below. Compatibility with other library or Python versions cannot be guaranteed, although the requirements should be tolerant in practice.

- From the standard libraries (installed with Python)
  - *configparser* configuration file handling
  - *os* operating systems commands
  - *pickle* saving and loading data
  - *re* handle regular expressions
- External libraries:
  - *matplotlib 2.0.0* data visualization
  - *numpy 1.11.3* support for matrix operations
  - *pandas 0.19.2* data import and visualization
  - *PyQt5 5.6* Python bindings for Qt
  - *scipy 0.18.1* scientific computing algorithms
  - *xlrd 1.0.0* Excel library for use with pandas

C.6.2 Loading, saving and conversion

Loading and saving data works the same as in the Matlab version, via the *Data* menu. Apart from Matlab files, a custom Python file format with extension '.segl' can also be selected. The *Convert from...* command has been replaced

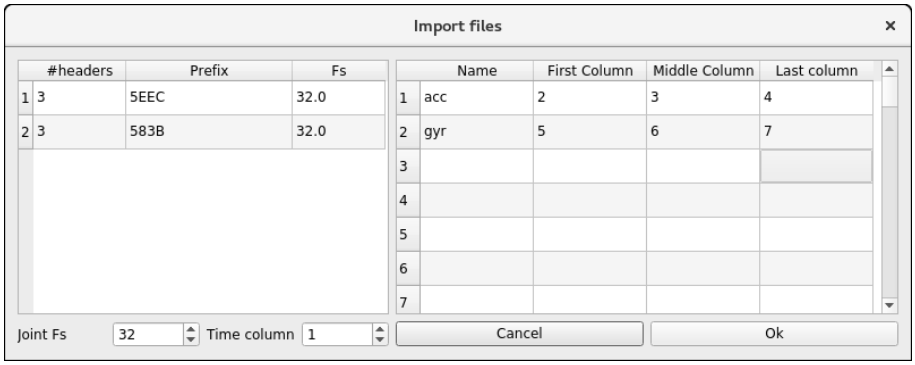


Figure C.8: The Python interface to import Excel or text files.

with *Import...*, however. The conversion process has a different interface as well, unified for Excel and text files. After selecting one or multiple files via a file dialog window, the interface in Figure C.8 is shown.

The selected import files are shown in the left table. In contrast to the Matlab version, each file can have a different number of headers and sampling frequency. The sampling frequency is estimated from the files themselves but can be set manually as well. Moreover, the prefix is directly defined. In the Matlab version, a user was only prompted for it afterwards when joining data files. Another difference is the fact that there is no option to select another sheet of an Excel workbook - all information has to be available on the first sheet. This allows to treat text files and workbooks uniformly. No installation of Microsoft Excel is required.

Below the left table one can define the sampling frequency after synchronizing and merging the files (joint Fs). Also, the time column has to be the same for all the files. It is estimated from the file headers by checking for the presence of the string 'time'.

The right table defines the sensors: their name and the columns for the three axes. They can be left blank if the device is not triaxial. Note that these sensor definitions should apply for all files in the left table.

The *Cancel* and *Ok* button cancel or confirm these settings, respectively. In contrast to the Matlab version, converted data is not saved, but directly opened in segmentLab.

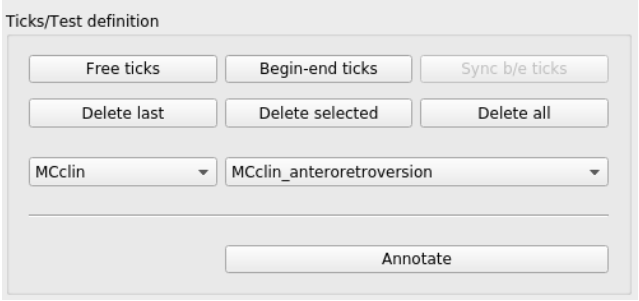


Figure C.9: The Python Ticks/Test definition panel.

C.6.3 Signals and annotations

The *signal visibility* panel is identical to its Matlab counterpart, both in appearance and functionality. Its bindings can be set via the *Edit* menu, as before. The Python *Ticks/Test definition* is shown in Figure C.9. A small difference for segmentation is the fact that begin-end ticks are now defined by clicking and dragging instead of two separate clicks. While dragging, a gray rectangle is shown. Also, due to technical reasons, the test category that is being segmented is not automatically visualized. This should be switched on manually in the Visualization panel.

A more notable difference is the absence of the interface to define new tests compared to Figure C.6. This is now done via the *Edit*→*Tests*... menu command. It shows a new window which offers more options than in Matlab. One can create new categories or tests, load or save test configuration under a custom name and even append a saved configuration to the one currently loaded. The interface is self-explanatory and will not be covered in detail.

Instead of the test definitions, an *Annotate* button is available in the panel. The Matlab version allows to load and visualize text annotations, but no interface is provided to enter them. In Python, this button enables the user to indicate a position and enter any text. The position is marked with a question mark which can be clicked to reveal the text. Text annotations can be removed via the *Delete selected* button.

In conclusion, a representation of the full Python user interface in action can be seen in Figure C.10.

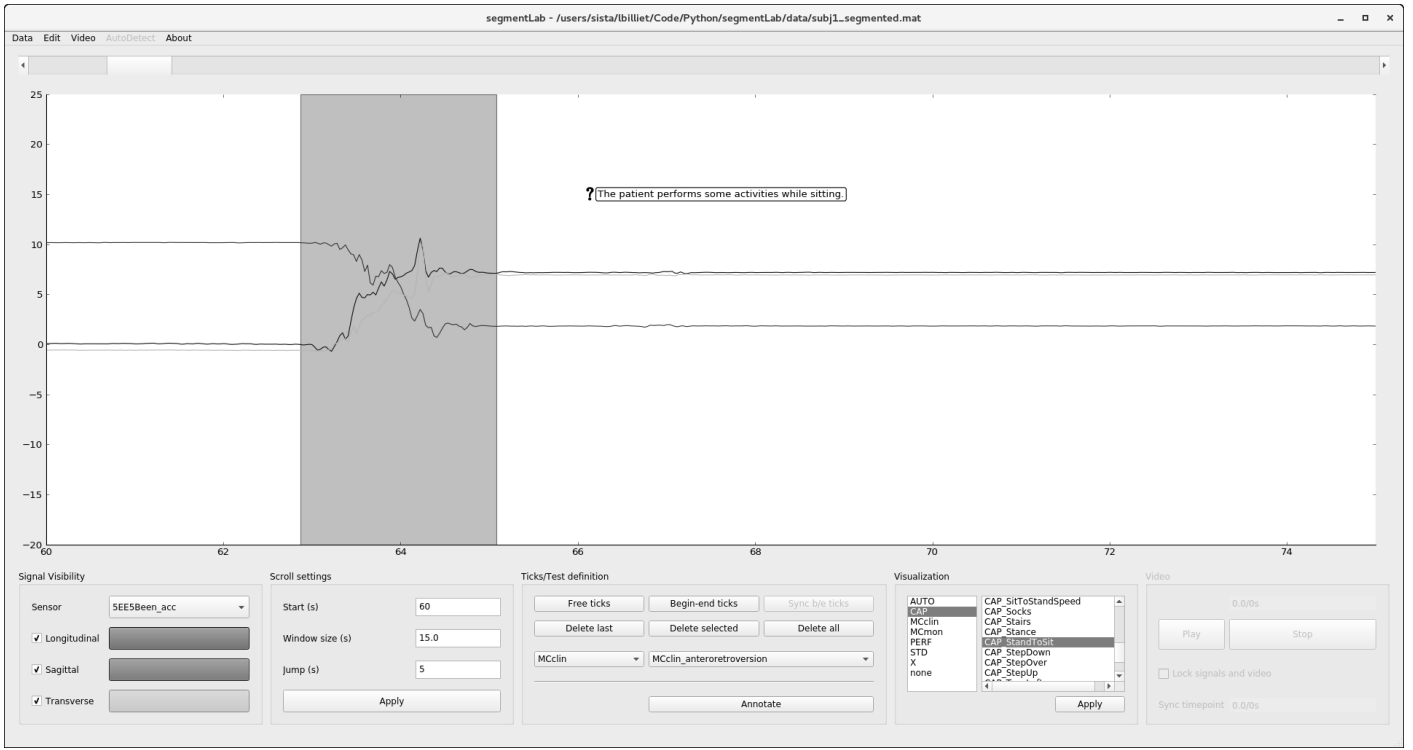


Figure C.10: The full Python interface showing a segmented sit to stand activity. A text annotation can also be observed at the question mark.

Bibliography

[1] Aggarwal, J., and Cai, Q. Human Motion Analysis: A Review. *Computer Vision and Image Understanding 73*, 3 (1999), 428–440.

[2] Ajam, T., and Mehdirad, A. A. CHADS2 and CHA2DS2-VASc Score for Stroke Risk Assessment in Atrial Fibrillation. `https://emedicine.medscape.com/article/2172597-overview`. Accessed: April 30, 2018.

[3] Albert, M. V., Toledo, S., Shapiro, M., and Koerding, K. Using Mobile Phones for Activity Recognition in Parkinson's Patients. *Frontiers in Neurology 3*, 158 (2012).

[4] Alberts, J. L., Hirsch, J. R., Koop, M. M., Schindler, D. D., Kana, D. E., Linder, S. M., Campbell, S., and Thota, A. K. Using Accelerometer and Gyroscopic Measures to Quantify Postural Stability. *Journal of Athletic Training 50*, 6 (2015), 578–588.

[5] Ali, H. H., Moftah, H. M., and Youssif, A. A. Depth-based Human activity Recognition: A Comparative Perspective Study on Feature Extraction. *Future Computing and Informatics Journal 3*, 1 (2017), 51–67.

[6] Alsina-Pagès, R. M., Navarro, J., Alías, F., and Hervás, M. Homesound: Real-time Audio Event Detection based on High Performance Computing for Behaviour and Surveillance Remote Monitoring. *Sensors 17*, 4 (2017), 854.

[7] Altini, M., Penders, J., Vullers, R., and Amft, O. Estimating Energy Expenditure Using Body-Worn Accelerometers: A Comparison of Methods, Sensors Number and Positioning. *IEEE Journal of Biomedical and Health Informatics 19*, 1 (2015), 219–226.

[8] Alvarado, A. A Practical Score for the Early Diagnosis of Acute Appendicitis. *Annals of Emergency Medicine 15*, 5 (1986), 557–564.

[9] Ankylosing Spondylitis International Federation. `http://www.asif.info/en/assets/File/basfi-eng.pdf`. Accessed: April 6, 2018.

[10] Aranda-Valera, I., Garrido-Castro, J., Martinez-Sanchez, I., Gonzalez, C., Gardiner, P., Machado, P., and Collantes, E. Inertial Motion Sensors using the Vimove System is a Valid Method to Assess Spinal Mobility in Patients with Axial Spondyloarthritis. *Annals of the Rheumatic Diseases 77*, Suppl 2 (2018), 642–643.

[11] Atallah, L., Lo, B., King, R., and Yang, G.-Z. Sensor Positioning for Activity Recognition Using Wearable Accelerometers. *IEEE Transactions on Biomedical Circuits and Systems 5*, 4 (2011), 320–329.

[12] Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., and Amirat, Y. Physical Human Activity Recognition using Wearable Sensors. *Sensors 15*, 12 (2015), 31314–31338.

[13] Avci, A., Bosch, S., Marin-Perianu, M., Marin-Perianu, R., and Havinga, P. Activity Recognition using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey. In *Proc 23rd International Conference on Architecture of Computing Systems (ARCS10)* (Hannover, Germany, Feb 2010), pp. 1–10.

[14] Ayres-de Campos, D., Bernardes, J., Garrido, A., Marques-de Sa, J., and Pereira-Leite, L. SisPorto 2.0: A Program for Automated Analysis of Cardiotocograms. *Journal of Maternal-Fetal Medicine 9*, 5 (2000), 311–318.

[15] Bader, B. W., Kolda, T. G., et al. MATLAB Tensor Toolbox Version 3.0-dev. Available online, Oct 2017. https://www.tensortoolbox.org.

[16] Banos, O., Damas, M., Pomares, H., Rojas, I., Toth, M. A., and Amft, O. A Benchmark Dataset to Evaluate Sensor Displacement in Activity Recognition. In *Proc 14th International ACM Conference on Ubiquitous Computing* (New York,NY, USA, 2012), pp. 1026–1035.

[17] Banos, O., Galvez, J.-M., Damas, M., Pomares, H., and Rojas, I. Window Size Impact in Human Activity Recognition. *Sensors 14*, 4 (2014), 6474–6499.

[18] Bao, L., and Intille, S. S. Activity Recognition from User-Annotated Acceleration Data. In *2nd International Conference on Pervasive Computing* (Linz and Vienna, Austria, Apr 2004), pp. 1–17.

[19] Barakat, N. H., and Bradley, A. P. Rule Extraction from Support Vector Machines: A Review. *Neurocomputing 74*, 1-3 (2010), 178–190.

[20] Berner, E. S., and La Lande, T. J. Overview of Clinical Decision Support Systems. In *Clinical Decision Support Systems: Theory and Practice*. Springer, Cham, 2016, pp. 1–17.

[21] Billiet, L., Swinnen, T., Milosevic, M., Dankaerts, W., Van Huffel, S., Westhovens, R., and de Vlam, K. Further Development of the Instrumented Bath Ankylosing Spondylitis Functional Index (iBASFI) in Axial Spondyloarthritis : the Added Value of Complex Accelerometry-derived Movement Features for Activity Capacity Assessment. In *Proc 4th International Conference on Ambulatory Monitoring of Physical Activity and Movement Abstract Booklet (ICAMPAM)* (Limerick, Ireland, Jun 2015), p. 31.

[22] Billiet, L., Swinnen, T., Westhovens, R., de Vlam, K., and Van Huffel, S. Activity Recognition for Physical Therapy - Fusing Signal Processing Features and Movement Patterns. In *3nd International Workshop on Sensor-based Activity Recognition and Interaction (iWOAR16)* (Rostock, Germany, Jun 2016), pp. 5:1–6.

[23] Billiet, L., Swinnen, T. W., Westhovens, R., de Vlam, K., and Van Huffel, S. Accelerometry-based Activity Recognition and Assessment in Rheumatic and Musculoskeletal Diseases. *Sensors 16*, 12 (2016), 2151.

[24] Billiet, L., Van Huffel, S., and Van Belle, V. Interval Coded Scoring Index with Interaction Effects: a Sensitivity Study. In *5th International Conference on Pattern Recognition Applications and Methods (ICPRAM)* (Rome, Italy, Feb 2016), pp. 33–40.

[25] Billiet, L., Van Huffel, S., and Van Belle, V. Interval Coded Scoring Extensions for Larger Problems. In *Proc 22nd International Symposium on Computers and Communication (ICSS17), ICTS4eHealth Workshop* (Heraklion, Greece, Jul 2017), pp. 198–203.

[26] Billiet, L., Van Huffel, S., and Van Belle, V. Interval Coded Scoring: a Toolbox for Interpretable Scoring Systems. *PeerJ Computer Science 4* (2018), e150.

[27] Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[28] Bland, J. M., and Altman, D. G. Statistical Methods for Assessing Agreement between Two Methods of Clinical Measurement. *The Lancet 327*, 8476 (1986), 307 – 310.

[29] Bodor, R., Jackson, B., and Papanikolopoulos, N. Vision-based Human Tracking and Activity Recognition. In *Proc 11th Mediterranean Conference on Control and Automation (MED03)* (Rhodos, Greece, Jun 2003), pp. 18–20.

[30] Bonomi, A. G., Plasqui, G., Goris, A. H., and Westerterp, K. R. Improving Assessment of Daily Energy Expenditure by Identifying Types of Physical Activity with a Single Accelerometer. *Journal of Applied Physiology 107*, 3 (2009), 655–661.

[31] Bottou, L., and Lin, C.-J. Support Vector Machine Solvers. *Large Scale Kernel Machines 3*, 1 (2007), 301–320.

[32] Breiman, L., Friedman, J., Stone, C., and Olshen, R. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984.

[33] Brionez, T. F., Assassi, S., Reveille, J. D., Learch, T. J., Diekman, L., Ward, M. M., Davis, J. C., Weisman, M. H., and Nicassio, P. Psychological Correlates of Self-reported Functional Limitation in Patients with Ankylosing Spondylitis. *Arthritis Research & Therapy 11*, 6 (2009), 1–9.

[34] Bulling, A., Blanke, U., and Schiele, B. A Tutorial on Human Activity Recognition using Body-worn Inertial Sensors. *ACM Computing Surveys 46*, 3 (2014), 33:1–33:33.

[35] Calin, A., Garrett, S., Whitelock, H., Kennedy, L. G., O'Hea, J., Mallorie, P., and Jenkinson, T. A New Approach to Defining Functional Ability in Ankylosing Spondylitis: the Development of the Bath Ankylosing Spondylitis Functional Index. *Journal of Rheumatology 21*, 12 (1994), 2281–2285.

[36] Callegaro, L., Pennecchi, F., and Spazzini, P. G. Comparison of Calibration Curves using the Lp Norm. *Accreditation and Quality Assurance 14*, 11 (2009), 587–592.

[37] Cantú-Paz, E., Newsam, S., and Kamath, C. Feature Selection in Scientific Applications. In *Proc 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Seattle, WA, USA, Aug 2004), pp. 788–793.

[38] Caso, F., Costa, L., Del Puente, A., Di Minno, M. N. D., Lupoli, G., Scarpa, R., and Peluso, R. Pharmacological Treatment of Spondyloarthritis: Exploring the Effectiveness of Nonsteroidal Anti-inflammatory Drugs, Traditional Disease-modifying Antirheumatic Drugs and Biological Therapies. *Therapeutic Advances in Chronic Disease 6*, 6 (2015), 328–338.

[39] Chapelle, O. Training a Support Vector Machine in the Primal. *Neural Computation 19*, 5 (2006), 1155–1178.

[40] Chaudhry, B., Wang, J., Wu, S., Maglione, M., Mojica, W., Roth, E., Morton, S. C., and Shekelle, P. G. Systematic Review: Impact of Health Information Technology on Quality, Efficiency and Costs of Medical Care. *Annals of Internal Medicine 144*, 10 (2006), 742–752.

[41] Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., and Sun, J. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. In *Proc. 1st Machine Learning for Healthcare Conference* (Boston, MA, USA, Aug 2016), pp. 301–318.

[42] Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., and Phan, H. A. Tensor Decompositions for Signal Processing Applications: From Two-way to Multiway Component Analysis. *IEEE Signal Processing Magazine 32*, 2 (2015), 145–163.

[43] Combettes, P. L., and Pesquet, J.-C. Proximal Splitting Methods in Signal Processing. In *Fixed-point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2011, pp. 185–212.

[44] Czerniak, J., and Zarzycki, H. Application of Rough Sets in the Presumptive Diagnosis of Urinary System Diseases. In *Proc 9th International Conference on Artificial Intelligence and Security in Computing Systems (ACS02)* (Miedzyzdroje, Poland, Oct 2002), pp. 41–51.

[45] da Rocha Neto, A. R., Sousa, R., de A. Barreto, G., and Cardoso, J. S. Diagnostic of Pathology on the Vertebral Column with Embedded Reject Option. In *Proc 5th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA11)* (Las Palmas de Gran Canaria, Spain, Jun 2011), pp. 588–595.

[46] Dax, A. The Adventures of a Simple Algorithm. *Linear Algebra and its Applications 361* (2003), 41–61.

[47] De Boeck, B., and Vandekerckhove, H. Applying Wearable Inertial Measurement Units to Assess Spinal Motor Control in Healthy Controls and Patients with Axial Spondyloarthritis: Reliability and Pilot Clinical Implementation. Master's thesis, KU Leuven, Faculty of Rehabilitation Sciences, Tervuursevest 101, 3001 Leuven, Belgium, Jun 2017.

[48] De Cooman, T., Kjær, T. W., Van Huffel, S., and Sorensen, H. B. Adaptive Heart Rate-based Epileptic Seizure Detection using Real-time User Feedback. *Physiological measurement 39*, 1 (2018), 014005.

[49] Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research 7*, 1 (2006), 1–30.

[50] Díaz-Uriarte, R., and Alvarez de Andrés, S. Gene Selection and Classification of Microarray Data using Random Forest. *BMC Bioinformatics 7*, 1 (2006), 3:1–3:13.

[51] Dice, L. R. Measures of the Amount of Ecologic Association between Species. *Ecology 26*, 3 (1945), 297–302.

[52] Dodd, R., Schieber, G., Cassels, A., Fleisher, L., and Gottret, P. Aid Effectiveness and Health. Tech. rep., World Health Organization, Geneva, Switzerland, 2007.

[53] Duch, W., Adamczak, R., Grabczewski, K., Ishikawa, M., and Ueda, H. Extraction of Crisp Logical Rules using Constrained Backpropagation Networks. In *Proc European Symposium on Artificial Neural Networks (ESANN)* (Bruges, Belgium, Apr 1997), pp. 109–114.

[54] Duda, R., Hart, P., and Stork, D. *Pattern Classification.* Wiley, 2001.

[55] Duda, R. O., and Shortliffe, E. H. Expert Systems Research. *Science 220*, 4594 (1983), 261–268.

[56] Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognition Letters 27*, 8 (2006), 861–874.

[57] Gaines, B. R., Kim, J., and Zhou, H. Algorithms for Fitting the Constrained Lasso. *Journal of Computational and Graphical Statistics 27*, 3 (2018).

[58] Ganea, R., Paraschiv-lonescu, A., and Aminian, K. Detection and Classification of Postural Transitions in Real-world Conditions. *IEEE Transactions on Neural Systems and Rehabilitation Engineering 20*, 5 (2012), 688–696.

[59] Garg, A. X., Adhikari, N. J., McDonald, H., Rosas-Arellano, M. P., Devereaux, P. J., Beyene, J., Sam, J., and Haynes, R. B. Effects of Computerized Clinical Decision Support Systems on Practitioner Performance and Patient Outcomes: A Systematic Review. *Journal of the American Medical Association 293*, 10 (2005), 1223–1238.

[60] Gevarter, W. B. An Overview of Computer Vision. Tech. rep., U.S. Department of Energy Office of Scientific and Technical Information, 1982.

[61] Glantz, S. A., Slinker, B. K., and Neilands, T. B. *Primer of Applied Regression and Analysis of Variance*, vol. 309. McGraw-Hill, 1990.

[62] Glaros, C., and I. Fotiadis, D. Wearable Devices in Healthcare. In *Intelligent Paradigms for Healthcare Enterprises*. Springer, 2005, pp. 237–264.

[63] Gorton, G. E., Hebert, D. A., and Gannotti, M. E. Assessment of the Kinematic Variability among 12 Motion Analysis Laboratories. *Gait & Posture 29*, 3 (2009), 398–402.

[64] Greene, B. R., Rutledge, S., McGurgan, I., McGuigan, C., O'Connell, K., Caulfield, B., and Tubridy, N. Assessment and Classification of Early-stage Multiple Sclerosis with Inertial Sensors: Comparison against Clinical Measures of Disease State. *IEEE Journal of Biomedical and Health Informatics 19*, 4 (2015), 1356–1361.

[65] Guenterberg, E., Ostadabbas, S., Ghasemzadeh, H., and Jafari, R. An Automatic Segmentation Technique in Body Sensor Networks based on Signal Energy. In *Proc 4th International Conference on Body Area Networks (BodyNets09)* (Los Angeles, CA, USA, Apr 2009), pp. 21:1–21:7.

[66] Gupta, P., and Dallas, T. Feature Selection and Activity Recognition System using a Single Triaxial Accelerometer. *IEEE Transactions on Biomedical Engineering 61*, 6 (2014), 1780–1786.

[67] Guvenir, H. A., Acar, B., Demiroz, G., and Cekin, A. A Supervised Machine Learning Algorithm for Arrhythmia Analysis. In *Computers in Cardiology* (Lund, Sweden, Aug 1997), pp. 433–436.

[68] Higashi, H., Rutkowski, T. M., Tanaka, T., and Tanaka, Y. Multilinear Discriminant Analysis with Subspace Constraints for Single-Trial Classification of Event-related Potentials. *IEEE Journal of Selected Topics in Signal Processing 10*, 7 (2016), 1295–1305.

[69] Higgins, W. T. A Comparison of Complementary and Kalman Filtering. *IEEE Transactions on Aerospace and Electronic Systems 11*, 3 (1975), 321–325.

[70] Ho, T. K. Random Decision Forests. In *Proc 3rd International Conference on Document Analysis and Recognition* (Montreal, Canada, Aug 1995), pp. 278–282.

[71] Hügli, A. S., Ernst, M. J., Kool, J., Rast, F. M., Rausch-Osthoff, A.-K., Mannig, A., Oetiker, S., and Bauer, C. M. Adherence to Home Exercises in Non-specific Low Back Pain. a Randomised Controlled Pilot Trial. *Journal of Bodywork and Movement Therapies 19*, 1 (2015), 177–185.

[72] Huisinga, J. M., Mancini, M., St. George, R. J., and Horak, F. B. Accelerometry Reveals Differences in Gait Variability Between Patients with Multiple Sclerosis and Healthy Controls. *Annals of Biomedical Engineering 41*, 8 (2012), 1670–1679.

[73] IBM. CPLEX Optimizer: High-performance Mathematical Programming Solver for Linear Programming, Mixed Integer Programming, and Quadratic Programming. `https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`, Accessed: Oct 3, 2017.

[74] Izenman, A. J. Linear Discriminant Analysis. In *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer, 2008, pp. 237–280.

[75] James, G. M., Paulson, C., and Rusmevichientong, P. The Constrained Lasso. Tech. rep., University of South California, 2012.

[76] Janssen, W. G. M., Bussmann, J. B. J., Horemans, H. L. D., and Stam, H. J. Analysis and Decomposition of Accelerometric Signals of Trunk and Thigh Obtained during the Sit-to-stand Movement. *Medical and Biological Engineering and Computing 43*, 2 (2005), 265–272.

[77] Jebara, T. *Machine Learning: Discriminative and Generative*. Springer Science & Business Media, 2012.

[78] Jeong, B.-H., Koh, W.-J., Yoo, H., Um, S.-W., Suh, G. Y., Chung, M. P., Kim, H., Kwon, O. J., and Jeon, K. Performances of Prognostic Scoring Systems in Patients with ealthcare-associated Pneumonia. *Clinical Infectious Diseases 56*, 5 (2013), 625–632.

[79] Jette, A. M., Haley, S. M., and Kooyoomjian, J. T. Are the ICF Activity and Participation Dimensions Distinct? *Journal of Rehabilitation Medicine 35*, 3 (2003), 145–149.

[80] Jette, M., Sidney, K., and Blümchen, G. Metabolic Equivalents (METS) in Exercise Testing, Exercise Prescription, and Evaluation of Functional Capacity. *Clinical Cardiology 13*, 8 (1990), 555–565.

[81] Jiménez, A. B., and Sra, S. Fast Newton-type Methods for Total Variation Regularization. In *Proc 28th International Conference on Machine Learning (ICML11)* (Bellevue, WA, USA, Jun 2011), pp. 313–320.

[82] Johnston, M. E., Langton, K. B., Haynes, R., and Mathieu, A. Effects of Computer-based Clinical Decision Support Systems on Clinician Performance and Patient Outcome: A Critical Appraisal of Research. *Annals of Internal Medicine 120*, 2 (1994), 135–142.

[83] Kale, N., Lee, J., Lotfian, R., and Jafari, R. Impact of Sensor Misplacement on Dynamic Time Warping Based Human Activity Recognition using Wearable Computers. In *Proc 3rd Conference on Wireless Health (WH12)* (La Jolla, CA, USA, Oct 2012), pp. 7:1–7:8.

[84] Keogh, E. J., and Pazzani, M. J. Derivative Dynamic Time Warping. In *Proc 1st SIAM International Conference on Data Mining* (Chicago, IL, USA, Apr 2001), pp. 1–11.

[85] Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. Multisensor Data Fusion: A Review of the State-of-the-Art. *Information Fusion 14*, 1 (2013), 28–44.

[86] Khan, A. M., Lee, Y.-K., Lee, S. Y., and Kim, T.-S. A Triaxial Accelerometer-based Physical Activity Recognition via Augmented Signal Features and a Hierarchical Recognizer. *IEEE transactions on Information Technology in Biomedicine 14*, 5 (2010), 1166–1172.

[87] Kim, T.-K., Wong, S.-F., and Cipolla, R. Tensor Canonical Correlation Analysis for Action Classification. In *Proc 17th IEEE Conference on Computer Vision and Pattern Recognition (CVPR07)* (Minneapolis, MN, USA, Jun 2007), pp. 1–8.

[88] Lambrechts, A. Development of a Risk Profile for the Acquired Long QT Syndrome in the Presence of Missing Data. Master's thesis, KU Leuven, Faculty of Engineering Science, Kasteelpark Arenberg 1, box 2200, May 2017.

[89] Lara, O. D., and Labrador, M. A. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys and Tutorials 15*, 3 (2013), 1192–1209.

[90] Lee, Y.-S., and Cho, S.-B. Activity Recognition using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer. In *Proc 4th International Conference on Hybrid Artificial Intelligence Systems (HAIS11)* (Wroclaw, Poland, May 2011), pp. 460–467.

[91] Letham, B., Rudin, C., McCormick, T. H., and Madigan, D. Interpretable Classifiers using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model. *Annals of Applied Statistics 9*, 3 (2015), 1350–1371.

[92] Leutheuser, H., Schuldhaus, D., and Eskofier, B. M. Hierarchical, Multi-Sensor based Classification of Daily Life Activities: Comparison with State-of-the-Art Algorithms using a Benchmark Dataset. *PLoS ONE 8*, 10 (2013), e75196.

[93] Lewis, D. D. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proc 10th European Conference on Machine Learning (ECML98)* (Chemnitz, Germany, Apr 1998), pp. 4–15.

[94] Liang, H.-N. Overview of the Health Informatics Research Field: A Bibliometric Approach. In *Proc 5th World Computer Congress (WCC10), 1st IMIA/IFIP Joint Symposium (E-Health10)* (Brisbane, Australia, Sep 2010), pp. 37–48.

[95] Lichman, M. UCI Machine Learning Repository, 2013.

[96] Lip, G. Y., Nieuwlaat, R., Pisters, R., Lane, D. A., and Crijns, H. J. Refining Clinical Risk Stratification for Predicting Stroke and Thromboembolism in Atrial Fibrillation using a Novel Risk Factor-based Approach: The Euro Heart Survey on Atrial Fibrillation. *Chest 137*, 2 (2010), 263–272.

[97] Liu, G., Xu, Y., Wang, X., Zhuang, X., Liang, H., Xi, Y., Lin, F., Pan, L., Zeng, T., Li, H., Cao, X., Zhao, G., and Xia, H. Developing a Machine Learning System for Identification of Severe Hand, Foot, and Mouth Disease from Electronic Medical Record Data. *Scientific Reports 7*, 1 (2017), 16341.

[98] Logesparan, L., Casson, A. J., and Rodriguez-Villegas, E. Optimal Features for Online Seizure Detection. *Medical & Biological Engineering & Computing 50*, 7 (2012), 659–669.

[99] Long, X. *On the Analysis and Classification of Sleep Stages from Cardiorespiratory Activity.* PhD thesis, TU Eindhoven, 2015.

[100] Lukas, C., Landewé, R., Sieper, J., Dougados, M., Davis, J., Braun, J., van der Linden, S., and van der Heijde, D. Development of an ASAS-endorsed Disease Activity score (ASDAS) in Patients with Ankylosing Spondylitis. *Annals of the Rheumatic Diseases 68*, 1 (2009), 18–24.

[101] Maity, N. G., and Das, S. Machine Learning for Improved Diagnosis and Prognosis in Healthcare. In *Proc 21st IEEE Aerospace Conference* (Big Sky, MT, USA, Mar 2017), pp. 1–9.

[102] Majumder, S., Mondal, T., and Deen, M. J. Wearable Sensors for Remote Health Monitoring. *Sensors 17*, 1 (2017), 130.

[103] Mallat, S. *A Wavelet Tour of Signal Processing: the Sparse Way.* Academic Press, 2008.

[104] Mallat, S. G. A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 7 (1989), 674–693.

[105] Mancini, M., Carlson-Kuhta, P., Zampieri, C., Nutt, J. G., Chiari, L., and Horak, F. B. Postural Sway as a Marker of Progression in Parkinson's Disease: a Pilot Longitudinal Study. *Gait & posture 36*, 3 (2012), 471–476.

[106] Mangasarian, O. L. Normal Solutions of Linear Programs. In *Mathematical Programming at Oberwolfach II*. Springer, 1984, pp. 206–216.

[107] Mangasarian, O. L., Street, W. N., and Wolberg, W. H. Breast Cancer Diagnosis and Prognosis via Linear Programming. *Operations Research 43*, 4 (1995), 570–577.

[108] Mani, S., Ozdas, A., Aliferis, C., Varol, H. A., Chen, Q., Carnevale, R., Chen, Y., Romano-Keeler, J., Nian, H., and Weitkamp, J.-H. Medical Decision Support using Machine Learning for Early Detection of Late-onset Neonatal Sepsis. *Journal of the American Medical Informatics Association 21*, 2 (2014), 326–336.

[109] Mannini, A., Intille, S. S., Rosenberger, M., Sabatini, A. M., and Haskell, W. Activity Recognition using a Single Accelerometer Placed at the Wrist or Ankle. *Medicine and Science in Sports and Exercise 45*, 11 (2013), 2193.

[110] Margarito, J., Helaoui, R., Bianchi, A. M., Sartor, F., and Bonomi, A. G. User-Independent Recognition of Sports Activities from a Single Wrist-worn Accelerometer: A Template-Matching-based Approach. *IEEE Transactions on Biomedical Engineering 63*, 4 (2016), 788–796.

[111] Martens, D., Baesens, B., Gestel, T. V., and Vanthienen, J. Comprehensible Credit Scoring Models using Rule Extraction from Support Vector Machines. *European Journal of Operational Research 183*, 3 (2007), 1466–1476.

[112] Martis, R. J., Chakraborty, C., and Ray, A. K. Wavelet-based Machine Learning Techniques for ECG Signal Analysis. In *Machine Learning in Healthcare Informatics*. Springer Berlin Heidelberg, 2014, pp. 25–45.

[113] Maurer, U., Smailagic, A., Siewiorek, D., and Deisher, M. Activity Recognition and Monitoring using Multiple Sensors on Different Body Positions. In *Proc 3rd International Workshop on Wearable and Implantable Body Sensor Networks (BSN06)* (Cambridge, MA, USA, Apr 2006), pp. 113–116.

[114] McDonald, J. H. *Handbook of Biological Statistics*, vol. 3. Sparky House Publishing, 2014.

[115] McKibbon, K. Evidence-based Practice. *Bulletin of the Medical Library Association 86*, 3 (1998), 396–401.

[116] Menze, M., and Geiger, A. Object Scene Flow for Autonomous Vehicles. In *Proc 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR15)* (Boston, MA, USA, Jun 2015), pp. 3061–3070.

[117] Minor, L. B. Stanford Medicine 2017 Health Trends Report: Harnessing the Power of Data in Health. Tech. rep., Stanford University School of Medicine, 2017. Available online: `http://med.stanford.edu/school/leadership/dean/healthtrends.html`.

[118] Mounzer, R., Langmead, C. J., Wu, B. U., Evans, A. C., Bishehsari, F., Muddana, V., Singh, V. K., Slivka, A., Whitcomb, D. C., Yadav, D., Banks, P. A., and Papachristou, G. I. Comparison of Existing Clinical Scoring Systems to Predict Persistent Organ Failure in Patients with cute Pancreatitis. *Gastroenterology 142*, 7 (2012), 1476–1482.

[119] Mourcou, Q., Fleury, A., Franco, C., Klopcic, F., and Vuillerme, N. Performance Evaluation of Smartphone Inertial Sensors Measurement for Range of Motion. *Sensors 15*, 9 (2015), 23168–23187.

[120] Müller, M. Dynamic Time Warping. In *Information Retrieval for Music and Motion*. Springer, 2007, pp. 69–84.

[121] Muscillo, R., Conforto, S., Schmid, M., Caselli, P., and D'Alessio, T. Classification of Motor Activities through Derivative Dynamic Time Warping Applied on Accelerometer Data. In *Proc 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS07)* (Lyon, France, Aug 2007), pp. 4930–4933.

[122] Needleman, S. B., and Wunsch, C. D. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins. *Journal of Molecular Biology 48*, 3 (1970), 443–453.

[123] Ozdalga, E., Ozdalga, A., and Ahuja, N. The Smartphone in Medicine: A Review of Current and Potential Use Among Physicians and Students. *Journal of Medical Internet Research 14*, 5 (2012), e218.

[124] O'Dwyer, T., O'Shea, F., and Wilson, F. Exercise Therapy for Spondyloarthritis: A Systematic Review. *Rheumatology International 34*, 7 (2014), 887–902.

[125] Papageorgiou, E. I. A New Methodology for Decisions in Medical Informatics using Fuzzy Cognitive Maps based on Fuzzy Rule Extraction Techniques. *Applied Soft Computing 11*, 1 (2011), 500–513.

[126] Pearson, K. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572.

[127] Pham, M. H., Elshehabi, M., Haertner, L., Del Din, S., Srulijes, K., Heger, T., Synofzik, M., Hobert, M. A., Faber, G. S., Hansen, C., et al. Validation of a Step Detection Algorithm during Straight Walking and Turning in Patients with Parkinson's Disease and Older Adults using an Inertial Measurement Unit at the Lower Back. *Frontiers in Neurology 8* (2017), 457.

[128] Phan, A. H. NFEA: Tensor Toolbox for Feature Extraction and Applications. Tech. rep., Lab for Advanced Brain Signal Processing, Brain Science Institute RIKEN, 2011.

[129] Phan, A. H., and Cichocki, A. Tensor Decompositions for Feature Extraction and Classification of High Dimensional Datasets. *Nonlinear Theory and Its Applications 1*, 1 (2010), 37–68.

[130] Ponce, H., Martínez-Villaseñor, M. d. L., and Miralles-Pechuán, L. A Novel Wearable Sensor-based Human Activity Recognition Approach using Artificial Hydrocarbon Networks. *Sensors 16*, 7 (2016), 1033.

[131] Poppe, R. A Survey on Vision-based Human Action Recognition. *Image and Vision Computing 28*, 6 (2010), 976–990.

[132] Preece, S., Goulermas, J., Kenney, L., and Howard, D. A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities from Accelerometer Data. *IEEE Transactions on Biomedical Engineering 56*, 3 (2009), 871–879.

[133] Preece, S., Goulermas, J., Kenney, L., Howard, D., Meijer, K., and Crompton, R. Activity Identification using Body-mounted Sensors – a Review of Classification Techniques. *Physiological Measurement 30*, 4 (2009), R1–R33.

[134] Protti, D. J. The Synergism of Health/Medical Informatics Revisited. *Methods of Information in Medicine 34*, 5 (1995), 441–445.

[135] Quinlan, J. R. Induction of Decision Trees. *Machine Learning 1*, 1 (1986), 81–106.

[136] Ramana, B. V., Babu, M. P., and Venkateswarlu, N. A Critical Comparative Study of Liver Patients from USA and India: An Exploratory Analysis. *International Journal of Computer Science Issues 9*, 2 (2012), 506–516.

[137] Roetenberg, D., Luinge, H., and Slycke, P. Xsens MVN: Full 6DOF Human Motion tracking using Miniature Inertial Sensors. Tech. rep., Xsens Motion Technologies BV, 2009.

[138] Roohi, S. F., Zonoobi, D., Kassim, A. A., and Jaremko, J. L. Dynamic MRI Reconstruction using Low Rank Plus Sparse Tensor Decomposition. In *Proc 23rd IEEE International Conference on Image Processing (ICIP16)* (Phoenix, AZ, USA, Sep 2016), pp. 1769–1773.

[139] Rubini, L. J., Eswaran, P., and Soundarapandian, P. Chronic Kidney Disease Dataset. Available online, 2015.

[140] Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear Total Variation based Noise Removal Algorithms. *Physica D: Nonlinear Phenomena 60*, 1-4 (1992), 259–268.

[141] Ryali, S., Chen, T., Supekar, K., and Menon, V. Estimation of Functional Connectivity in fMRI Data using Stability Selection-based Sparse Partial Correlation with Elastic Net Penalty. *NeuroImage 59*, 4 (2012), 3852–3861.

[142] Sachs, J. D. America's Health Crisis and the Easterlin Paradox. In *World Happiness Report 2018*. Sustainable Development Solutions Network, 2018, ch. 7.

[143] Schlimmer, J. C. *Concept Acquisition through Representational Adjustment*. PhD thesis, University of California, Irvine, 1987.

[144] Semanik, P. A., Lee, J., Song, J., Chang, R. W., Sohn, M.-W., Ehrlich-Jones, L., Ainsworth, B. E., Nevitt, M. M., Kwoh, C. K., and Dunlop, D. D. Accelerometer-monitored Sedentary Behavior and Observed Physical Function Loss. *American Journal of Public Health 105*, 3 (2015), 560–566.

[145] Sharma, S., Ayyaswamy, J., and Kishore, J. Technology Driven Healthcare. *Advanced Techniques in Biology & Medicine* (2015), 1–4.

[146] Shen, D., Wee, C.-Y., Zhang, D., Zhou, L., and Yap, P.-T. Machine Learning Techniques for AD/MCI Diagnosis and Prognosis. In *Machine Learning in Healthcare Informatics*. Springer, 2014, pp. 147–179.

[147] Shen, L., Kim, S., Qi, Y., Inlow, M., Swaminathan, S., Nho, K., Wan, J., Risacher, S. L., Shaw, L. M., Trojanowski, J. Q., Weiner, M. W., and Saykin, A. J. Identifying Neuroimaging and Proteomic Biomarkers for MCI and AD via the Elastic Net. In *Proc. 1st International Conference on Multimodal Brain Image Analysis (MBIA11)* (Toronto, Canada, Sep 2011), pp. 27–34.

[148] Shibl, R., Lawley, M., and Debuse, J. Factors Influencing Decision Support System Acceptance. *Decision Support Systems 54*, 2 (2013), 953–961.

[149] Shumway-Cook, A., Brauer, S., and Woollacott, M. Predicting the Probability for Falls in Community-dwelling Older Adults using the Timed Up & Go Test. *Physical Therapy 80*, 9 (2000), 896–903.

[150] Sieper, J., van der Heijde, D., Landewe, R., Brandt, J., Burgos-Vagas, R., Collantes-Estevez, E., Dijkmans, B., Dougados, M., Khan, M., Leirisalo-Repo, M., van der Linden, S., Maksymowych, W., Mielants, H., Olivieri, I., and Rudwaleit, M. New Criteria for Inflammatory Back Pain in Patients with Chronic Back Pain: A Real Patient Exercise by Experts from the Assessment of Spondyloarthritis International Society (ASAS). *Annals of the Rheumatic Diseases 68*, 6 (2009), 784–788.

[151] Sørensen, T. A Method of Establishing Groups of Equal Amplitude in Plant Sociology based on Similarity of Species Content and its Application to Analyses of the Vegetation on Danish Commons. *Det Kongelige Danske Videnskabernes Selskab Biologiske Skrifter 5*, 4 (1948).

[152] Speier, C., Valacich, J. S., and Vessey, I. The Influence of Task Interruption on Individual Decision Making: An Information Overload Perspective. *Decision Sciences 30*, 2 (1999), 337–360.

[153] Sra, S. Efficient Large Scale Linear Programming Support Vector Machines. In *Proc 17th European Conference on Machine Learning (ECML 2006)* (Berlin, Germany, Sep 2006), pp. 767–774.

[154] Steins, D., Dawes, H., Esser, P., and Collett, J. Wearable Accelerometry-based Technology Capable of Assessing Functional Activities in Neurological Populations in Community Settings: A Systematic Review. *Journal of NeuroEngineering and Rehabilitation 11*, 1 (2014), 1–13.

[155] Steyerberg, E. W., Eijkemans, M. J. C., Frank E. Harrell, J., and Habbema, J. D. F. Prognostic Modeling with Logistic Regression Analysis. *Medical Decision Making 21*, 1 (2001), 45–56.

[156] Suárez-León, A. A., Varon, C., Goovaerts, G., Vázquez-Seisdedos, C. R., and Van Huffel, S. Irregular Heartbeat Detection using Sequentially Truncated Multilinear Singular Value Decomposition. In *Proc 44th Annual Conference on Computing in Cardiology* (Rennes, France, Sep 2017), vol. 44, p. 1.

[157] Suryadevara, N. K., and Mukhopadhyay, S. C. Wireless Sensor Network based Home Monitoring System for Wellness Determination of Elderly. *IEEE Sensors Journal 12*, 6 (2012), 1965–1972.

[158] Suykens, J. A., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. *Least Squares Support Vector Machines*. World Scientific, 2002.

[159] Swinnen, T. W., Milosevic, M., Van Huffel, S., Dankaerts, W., Westhovens, R., and de Vlam, K. Instrumented BASFI (iBASFI) Shows Promising Reliability and Validity in the Assessment of Activity Limitations in Axial Spondyloarthritis. *Journal of Rheumatology 43*, 8 (2016), 1532–1540.

[160] Tapia, E. M., Intille, S. S., and Larson, K. Activity Recognition in the Home using Simple and Ubiquitous Sensors. In *2nd International Conference on Pervasive Computing* (Linz and Vienna, Austria, Apr 2004), pp. 158–175.

[161] Taurog, J. D. The Role of HLA-B27 in Spondyloarthritis. *Journal of Rheumatology 37*, 12 (2010), 2606–2616.

[162] Taurog, J. D., Chhabra, A., and Colbert, R. A. Ankylosing Spondylitis and Axial Spondyloarthritis. *New England Journal of Medicine 374*, 26 (2016), 2563–2574.

[163] Thorp, A. A., Healy, G. N., Winkler, E., Clark, B. K., Gardiner, P. A., Owen, N., and Dunstan, D. W. Prolonged Sedentary Time and Physical Activity in Workplace and Non-work Contexts: A Cross-sectional Study of Office, Customer Service and Call Centre Employees. *International Journal of Behavioral Nutrition and Physical Activity 9*, 1 (2012), 128.

[164] Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology 58*, 1 (1996), 267–288.

[165] Tibshirani, R. J., Taylor, J., et al. The Solution Path of the Generalized Lasso. *The Annals of Statistics 39*, 3 (2011), 1335–1371.

[166] Ulaş, A., Yildiz, O. T., and Alpaydin, E. Cost-conscious Comparison of Supervised Learning Algorithms over Multiple Data Sets. *Pattern Recognition 45*, 4 (2012), 1772–1781.

[167] Ustun, B., and Rudin, C. Supersparse Linear Integer Models for Optimized Medical Scoring Systems. *Machine Learning 102*, 3 (2016), 349–391.

[168] Van Belle, V. M. C. A., Van Calster, B., Timmerman, D., Bourne, T., Bottomley, C., Valentin, L., Neven, P., Van Huffel, S., Suykens, J. A. K., and Boyd, S. A Mathematical Model for Interpretable Clinical Decision Support with Applications in Gynaecology. *PLoS ONE 7*, 3 (2012), e34312.

[169] Van Damme, M. Robust Detection of Sit-Stand Transitions using Inertial Sensors. Master's thesis, KU Leuven, Faculty of Bioscience Engineering, Kasteelpark Arenberg 20, box 2300, May 2017.

[170] Van Tubergen, A., Landewé, R., Van Der Heijde, D., Hidding, A., Wolter, N., Asscher, M., Falkenbach, A., Genth, E., Thè, H. G., and van der Linden, S. Combined Spa–Exercise Therapy is Effective in Patients with Ankylosing Spondylitis: A Randomized Controlled Trial. *Arthritis Care & Research 45*, 5 (2001), 430–438.

[171] van Weely, S. F., van Denderen, J. C., Steultjens, M. P., van der Leeden, M., Nurmohamed, M. T., Dekker, J., Dijkmans, B. A., and van der Horst-Bruinsma, I. E. Moving Instead of Asking? Performance-based Tests and BASFI Questionnaire Measure Different Aspects of Physical Function in Ankylosing Spondylitis. *Arthritis Research & Therapy 14*, 2 (2012), R52.

[172] Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[173] Williamson, J., Liu, Q., Lu, F., Mohrman, W., Li, K., Dick, R., and Shang, L. Data Sensing and Analysis: Challenges for Wearables. In *Proc 20th Asia and South Pacific Design Automation Conference (ASP-DAC15)* (Chiba and Tokya, Japan, Jan 2015), pp. 136–141.

[174] Woolf, A. D., and Pfleger, B. Burden of Major Musculoskeletal Conditions. *Bulletin of the World Health Organization 81*, 9 (2003), 646–656.

[175] World Health Organisation. Towards a Common Language for Functioning, Disability and Health: ICF. Tech. rep., World Health Organisation, 2002.

[176] World Health Organization. Current Health Expenditure as a Percentage of Gross Domestic Product. `http://www.who.int/gho/health_financing/health_expenditure/en/`, 2015.

[177] Xu, H., Eis, D. J., and Ramadge, P. J. The Generalized Lasso is Reducible to a Subspace Constrained Lasso. In *Proc 38th International Conference on Acoustics, Speech and Signal Processing (ICASSP13)* (Vancouver, Canada, May 2013), IEEE, pp. 3268–3272.

[178] Xu, X., Lin, F., Wang, A., Yao, X., Lu, Q., Xu, W., Shi, Y., and Hu, Y. Accelerating Dynamic Time Warping with Memristor-based Customized Fabrics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 37*, 4 (2018), 729–741.

[179] Yang, A. Y., Iyengar, S., Kuryloski, P., and Jafari, R. Distributed Segmentation and Classification of Human Actions using a Wearable Motion Sensor Network. In *Proc 5th IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW08)* (Anchorage, AK, USA, Jun 2008), pp. 1–8.

[180] Zhang, M., and Sawchuk, A. A. Motion Primitive-based Human Activity Recognition using a Bag-of-features Approach. In *Proc of 2nd ACM SIGHIT International Health Informatics Symposium (ICI12)* (Miami, FL, USA, Jan 2012), pp. 631–640.

[181] Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., and Li, Z. A Review on Human Activity Recognition using Vision-based Method. *Journal of Healthcare Engineering 2017* (2017), 3090343.

[182] Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. Face Recognition: A Literature Survey. *ACM Computing Surveys (CSUR) 35*, 4 (2003), 399–458.

[183] Zhou, F., and De la Torre, F. Generalized Time Warping for Multi-modal Alignment of Human Motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR12)* (Providence, RI, USA, Jun 2012), pp. 1282–1289.

[184] Zhou, Q., Chen, W., Song, S., Gardner, J. R., Weinberger, K. Q., and Chen, Y. A Reduction of the Elastic Net to Support Vector Machines with an Application to GPU Computing. In *Proc 29th Conference on Artificial Intelligence (AAAI15)* (Austin, TX, USA, Jan 2015), pp. 3210–3216.

[185] Zochling, J. Measures of Symptoms and Disease Status in Ankylosing Spondylitis: Ankylosing Spondylitis Disease Activity Score (ASDAS), Ankylosing Spondylitis Quality of Life Scale (ASQoL), Bath Ankylosing Spondylitis Disease Activity Index (BASDAI), Bath Ankylosing Spondylitis Functional Index (BASFI), Bath Ankylosing Spondylitis Global Score (BAS-G), Bath Ankylosing Spondylitis Metrology Index (BASMI), Dougados Functional Index (DFI), and Health Assessment Questionnaire for the Spondylarthropathies (HAQ-S). *Arthritis Care & Research 63*, S11 (2011).

[186] Zou, H., and Hastie, T. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology 67*, 2 (2005), 301–320.

Curriculum

Lieven J. L. Billiet was born on March 13, 1989 in Wilrijk (Antwerp). He received a B.Sc. degree in Engineering Science with a major in Computer Science and a minor in Electrical Engineering with distinction from the University of Leuven in 2010. In 2010-2011, he studied abroad in Stockholm, Sweden as part of the Erasmus programme. In 2012, he obtained a M.Sc. degree in Computer Science with a focus on Human-Computer Interaction with great distinction from the University of Leuven. His dissertation on sign recognition with the Kinect was supervised by professor L. De Raedt. In 2014 he received a second M.Sc. degree, in Biomedical Engineering, a dual degree granted with great distinction by the Universities of Leuven and Louvain-la-Neuve. His dissertation on single-trial classification of EEG responses was supervised by professor S. Van Huffel. Since 2014, he has been active as a doctoral researcher at the Biomed research group, part of STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Department of Electrical Engineering (ESAT), Leuven under the supervision of professor Van Huffel and dr. V. Van Belle.

During his research, he has been part of the KU Leuven SPARKLE interdisciplinary research project on remote patient monitoring with body area networks led by professor R. Westhovens. It involved a collaboration between ESAT-STADIUS, the Division of Rheumatology of the University Hospital, the faculty of Rehabilitation Science (FaBeR), the research group on Microelectronics and Sensors (ESAT-MICAS) and the eMedia Lab at the faculty of Engineering Technology (Group T).

His current research interests include the use of wearables for activity recognition and assessment, machine learning and optimization, interpretable decision support systems and their application on biomedical problems such as monitoring of axSpA patients.

List of publications

Articles in internationally reviewed academic journals

1. **Billiet L.**, Swinnen T., Westhovens R., De Vlam K., Van Huffel S. (2016). Accelerometry-based Activity Recognition and Assessment in Rheumatic and Musculoskeletal Diseases. *Sensors*, 16 (12), 2151, pp. 1-19.

2. **Billiet L.**, Van Huffel S., Van Belle V. (2018). Interval coded scoring: a toolbox for interpretable scoring systems. *PeerJ Computer Science*, 4, e150, pp. 1-28.

3. **Billiet L.**, Swinnen T., De Vlam K., Westhovens R., Van Huffel S. (2018). Recognition of Physical Activities from a Single Arm-worn Accelerometer: a Multiway Approach. *Informatics*, 5 (2), 20, pp. 1-19.

Papers at international scientific conferences and symposia, published in full in proceedings

4. **Billiet L.**, Oramas Mogrovejo J., Hoffmann M., Meert W., Antanas L. (2013). Rule-based Hand Posture Recognition using Qualitative Finger Configurations Acquired with the Kinect. *Proc 2nd International Conference on Pattern Recognition Applications and Methods* (ICPRAM13). Barcelona, Spain, Feb 2013 (pp. 1-4).

5. **Billiet L.**, Hunyadi B., Matic V., Van Huffel S., Verleysen M., De Vos M. (2015). Single Trial Classification in Mobile BCI - A Multiway Kernel Approach. *Proc 8th International Conference on Bio-inspired Systems and Signal Processing* (BIOSIGNALS15). Lisbon, Portugal, Jan 2015 (pp. 5-11).

6. **Billiet L.**, Swinnen T., Westhovens R., De Vlam K., Van Huffel S. (2016). Activity Recognition for Physical Therapy - Fusing Signal Processing Features and Movement Patterns. Proc 3rd International Workshop on Sensor-based Activity Recognition and Interaction (iWOAR16). Rostock, Germany, Jun 2016 (pp. 5:1-5:6).

7. **Billiet L.**, Van Huffel S., Van Belle V. (2016). Interval Coded Scoring Index with Interaction Effects: a Sensitivity Study. *Proc 5th International Conference on Pattern Recognition Applications and Methods* (ICPRAM16). Rome, Italy, Feb 2016 (pp. 33-40).

8. **Billiet L.**, Swinnen T., Westhovens R., De Vlam K., Van Huffel S. (2016). SPARKLE@Home : Bringing Health Technology to Daily Life. *Proc 2nd Conference on Engineering4Society* (E4S16). Leuven, Belgium, Sep 2016 (pp. 33-39).

9. **Billiet L.**, Van Huffel S., Van Belle V. (2017). Interval Coded Scoring Extensions for Larger Problems. *Proc 22nd IEEE symposium on Computers and Communications, 2nd ICTS4eHealth workshop* (ISCC17). Heraklion, Greece, Jul 2017 (pp. 198-203).

Meeting abstracts, presented at national or international scientific conferences and symposia

10. {**Billiet, L.**, Swinnen, T.}[1], Milosevic, M., Dankaerts, W., Van Huffel, S., Westhovens, R., De Vlam, K. (2015). Further development of the instrumented Bath Ankylosing Spondylitis Functional Index (iBASFI) in Axial Spondyloarthritis : The Added Value of Complex Accelerometry-derived Movement Features for Activity Capacity Assessment. *Proc 4th International Conference on Ambulatory Monitoring of Physical Activity and and Movement* (ICAMPAM15). Limerick, Ireland, Jun 2015, p. 31.

11. **Billiet L.**, Van Huffel S. (2016). Activity Recognition for Physical Therapy: Moving to the Home Environment. *Belgian National Day on Biomedical Engineering* (BNDBME16). Brussels, Belgium, Nov 2016.

12. **Billiet L.**, Van Huffel S. (2017). Interval Coded Scoring: Towards Data-derived Medical Scoring Systems. *6th Dutch Biomedical Engineering Conference* (BME17). Egmond aan Zee, The Netherlands, Jan 2017.

[1] Equal contribution



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
STADIUS CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING AND DATA ANALYTICS
Kasteelpark Arenberg 10 box 2446
B-3001 Leuven
lieven.billiet@kuleuven.be
http://esat.kuleuven.be/stadius