

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

PROBABILISTIC MACHINE LEARNING APPROACHES TO MEDICAL CLASSIFICATION PROBLEMS

Promotors:
Prof. dr. ir. S. Van Huffel
Prof. dr. ir. J.A.K. Suykens

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Chuan LU

January 2005



KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

PROBABILISTIC MACHINE LEARNING APPROACHES TO MEDICAL CLASSIFICATION PROBLEMS

Jury:
Prof. dr. ir. L. Froyen, voorzitter
Prof. dr. ir. S. Van Huffel, promotor
Prof. dr. ir. J.A.K. Suykens, promotor
Prof. dr. D. Timmerman
Prof. dr. ir. J. Vandewalle
Prof. dr. J. Beirlant
Prof. dr. P.J.G. Lisboa
Prof. dr. ir. Y. Moreau

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Chuan LU

U.D.C. 681.3*I2 January 2005

© Katholieke Universiteit Leuven – Faculteit Toegepaste Wetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, 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 or any other means without written permission from the publisher.

D/2005/7515/8

ISBN 90-5682-573-9

Acknowledgement

This thesis will not have been possible without the help and support of many people.

First of all, I would like to express my gratitude to Prof. Sabine Van Huffel, who guided me throughout these five years ever since she accepted me as her master's thesis student, who introduced me into the wonderland of biomedical application research, and who has always been there, listening to our ideas, progress or difficulties, and providing her support. Her knowledge, advices, her confidence in me and the work have inspired me and encouraged me to get through this long journey towards my PhD.

My thanks go to Prof. Johan Suykens for his supervision, his encouragement, and his valuable comments that often led me to recheck my work more carefully. I have also benefited a lot from his insights and big pictures in the area of machine learning and nonlinear modelling.

I would like to thank Prof. Dirk Timmerman for his kind support and provision of medical expertise. I am grateful to Prof. Joos Vandewalle for his constant encouragement, his interest and valuable advices in my work. Many thanks also go to Prof. Jan Beirlant, for his time, support, and his wonderful lectures in applied statistical models. I would also like to thank the jury members: Prof. Paulo Lisboa, Prof. Yves Moreau and the chairman Prof. Ludo Froyen.

I should not forget the collaboration and help I received from other colleagues these years. I am thankful to Tony Van Gestel for his ideas, thoughts and our fruitful collaboration. I appreciate the collaboration with George Condous (St George's Hospital Medical School, London), who provided medical data and raised many questions that statisticians and machine learning people are keen to solve. I am grateful to Prof. Giansalvo Cirrincione, whose ideas and enthusiasm in neural networks have been enlightening in the early stage of my research.

I am grateful to Lieveke Ameye, Andy Devos, Lukas, Jean-Michel Papy, Qizheng Sheng and Yu Wang for all those interesting conversations and their friendship. I should thank Kristiaan Pelckmans, Jos De Brabanter, Peter Antal,

Ivan Markovsky, and many others, who have helped me in one way or another. Thanks go to all the people of the BioMed group for building up a friendly working environment here. I sincerely appreciate the efforts of Ida Tassens, Ilse Pardon and Bart Motmans, for their help with the administrative matters.

My past six years in Leuven have been so enjoyable thanks to my friends around, Leuven's international but cozy atmosphere, and its charming sport center as well.

Finally, I thank my family for their continuous loving support and help. Many thanks go to my sister Yang, for sharing with me her enthusiasm and thoughts in AI, and those inspirational discussions. I also appreciate Pieter, for his kindness and help in need. I am in debt so much to my parents, for their endless love and encouragement. I can not thank enough my husband Shigang for his love, understanding and tolerance throughout the duration of my research studies.

Abstract

Recent advances in technologies and computations have facilitated the collection of biomedical data. This also leads to the development of computer-based tools that are able to handle and analyze the various types of medical data in order to assist the decision making process. Probabilistic machine learning approaches are shown to be suitable for such complex tasks.

This thesis examines the use of intelligent computational methods in the development of predictive models for medical classification based on patient data. In particular the Bayesian kernel-based models including Bayesian least squares support vector machines (LS-SVMs) and relevance vector machines (RVMs) have been our focus. Other linear and nonlinear classification techniques including logistic regression (LR), linear discriminant analysis (LDA), and multilayer perceptrons (MLPs) have been compared. Within the Bayesian framework, (hyper)parameters tuning, model parameter estimation, sparseness approximation, model comparison, variable selection, and model prediction can be conducted in a unifying way. We propose two variable selection schemes within Bayesian frameworks. One scheme is the forward selection by maximizing model evidence of Bayesian LS-SVMs for both linear and nonlinear classification problems. The other one integrates the bagging strategy with sequential linear sparse Bayesian learning algorithms, and is suitable for data sets with high dimensionality and relatively small sample size. Furthermore, it is demonstrated that an interpretation is possible in terms of additive kernel-based classifiers. It is also useful to visualize the LS-SVM classifiers by means of kernel principal component regression and nonlinear biplots.

The methodologies have been successfully applied to several real world medical diagnosis problems. Our major application is the predictive modelling for preoperative ovarian tumor classification, using the data from the K.U. Leuven pilot project and the international multi-center IOTA project. The other case studies include brain tumor classification using magnetic resonance spectroscopy (MRS) spectra, and benchmarking problems for leukemia and colon cancer diagnosis based on microarray gene expression data.

Contents

Acknowledgement i

Abstract iii

Contents v

Glossary xi

1 Introduction 1

1.1 Overview of machine learning 1

1.2 Clinical decision support systems 4

1.3 Medical classification problems 5

1.4 Problem challenges 6

1.5 Probabilistic machine learning approaches to classification . . . 8

1.5.1 Bayesian decision theory 8

1.5.2 Probabilistic classifiers 9

1.5.3 Learning and parameter estimation 11

1.6 Goal of the thesis 12

1.7 Structure of the thesis 12

1.8 Contributions and related work 14

2 Supervised learning **17**

2.1 A generic supervised learning model and regularization 17

2.2 Conventional linear classifiers 19

2.2.1 Linear discriminant analysis 19

2.2.2 Logistic regression 22

2.3 Artificial neural networks 23

2.3.1 Multilayer perceptrons 23

2.3.2 Radial basis function neural networks 24

2.4 Support vector machines for classification 26

2.4.1 Statistical learning theory 26

2.4.2 Kernel induced feature spaces 29

2.4.3 Support vector classifiers 30

2.4.4 Least squares support vector classifiers 33

2.4.5 Generalized additive kernel-based models 35

2.4.6 Kernel principal component regression and LS-SVMs . . 36

2.5 Further topics 38

2.5.1 Variable selection 38

2.5.2 Multiclass classification 40

2.5.3 Ensemble methods for predictive modelling 44

2.6 Model evaluation 45

2.6.1 Performance measure 45

2.6.2 Validation strategies 48

2.6.3 Validation of medical diagnostic models 50

2.7 Summary 50

3 Bayesian frameworks for blackbox models **53**

3.1 Bayesian methods 54

3.1.1 Principle of Bayesian learning 54
3.1.2 Bayesian inferences for a generic probabilistic model . . 56
3.2 Bayesian multilayer perceptrons 58
3.3 Sparse Bayesian learning models 59
3.3.1 Introduction 59
3.3.2 An algorithm for sequential sparse Bayesian learning . . 61
3.3.3 Relevance vector machine classifiers 63
3.3.4 A toy example of using RVMs for classification 63
3.4 Bayesian least squares support vector machines 65
3.4.1 Probabilistic inferences in LS-SVM within the evidence framework 66
3.4.2 Design of the LS-SVM classifier in a Bayesian evidence framework 72
3.4.3 Toy examples on Bayesian LS-SVM classifiers 74
3.5 Summary 81

4 Linear and nonlinear models for preoperative classification of ovarian tumors 83

4.1 Introduction 84
4.2 Methods 86
4.3 Preliminary investigation - on the K.U. Leuven pilot project . . 87
4.3.1 Data 87
4.3.2 Experimental settings 88
4.3.3 Selecting predictive input variables 90
4.3.4 Model fitting and prediction 91
4.3.5 Model evaluation 92
4.4 Extensive study - on the multi-center IOTA project 101
4.4.1 The IOTA project 101

4.4.2 Data analysis 102
4.4.3 Experimental settings 104
4.4.4 Prospective evaluation of the models built in the K.U. Leuven pilot project 106
4.4.5 Model development using the IOTA data set 115
4.5 Discussion 127
4.6 Conclusions: towards robust ovarian cancer diagnostic models . 133

5 Bagging linear sparse Bayesian learning models for variable selection in cancer diagnosis 135

5.1 Introduction 136
5.2 Methods 137
5.2.1 Bagging strategy 138
5.2.2 Compared methods 139
5.3 Experiments 140
5.3.1 Experimental settings 140
5.3.2 Binary cancer classification based on microarray data . 141
5.3.3 Classification of brain tumors based on MRS data . . . 144
5.3.4 Biological relevance of the selected variables 144
5.4 Discussion 150
5.5 Conclusions 151

6 Conclusions and future research 153

6.1 Summary 153
6.2 Future research 155

A Medical background of the diseases 159

B Variable descriptions for the IOTA data set 161

Bibliography **167**

Publication list **179**

Biography **183**

Glossary

This section lists symbols and acronyms that occur frequently in this thesis.

Remark. The notation used in this thesis allows one to distinguish among scalars, vectors and matrices. Characters without boldface represent scalar values. Boldface lower case characters are used for vectors. Boldface capitals represent matrices.

Mathematical Notation

$\mathbf{x}$	vector $\mathbf{x} = [x_1, x_2, \cdots, x_D]^T$
$\mathbf{A}$	matrix $\mathbf{A}$
$\mathbf{A}^T$	transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of matrix $\mathbf{A}$
$\det \mathbf{A}$	determinant of matrix $\mathbf{A}$
$\mathbf{x}_m$	mth element of vector $\mathbf{x}$
$\mathbf{A}_{nm}$	element on the nth row and mth column of matrix $\mathbf{A}$
$\mathbb{N}$	the set of natural numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^D$	set of real D-dimensional vectors
$\hat{w}$	estimate of w
$\lvert x \rvert$	absolute value of x
$\lVert \mathbf{x} \rVert$	2-norm of vector $\mathbf{x}$, $\lVert \mathbf{x} \rVert = \sqrt{(\sum_{d=1}^{D} x_d^2)}$
$\mathrm{diag}(\mathbf{x})$	a matrix of $\mathbb{R}^{D \times D}$ with vector $\mathbf{x}$ on the diagonal, all other elements being 0
$\mathrm{Trace}(\mathbf{A})$	the trace $\sum_i \mathbf{A}_{ii}$ of matrix $\mathbf{A}$
$\log(x)$	the natural logarithm of x
$\mathrm{sign}(x)$	the sign of x
$\langle \mathbf{x}, \mathbf{z} \rangle$	the inner product $\mathbf{x}^T\mathbf{z}$ of the vectors $\mathbf{x}$ and $\mathbf{z}$
$k(\cdot, \cdot)$	kernel function

$\frac{\partial f(\mathbf{x})}{\partial x_i}$	the partial derivative of $f(\mathbf{x})$ w.r.t. the ith component of $\mathbf{x}$
$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$	gradient of $f(\mathbf{x})$ $(= [\frac{\partial f(\mathbf{x})}{\partial x_1}, \cdots, \frac{\partial f(\mathbf{x})}{\partial x_D}]^T)$
$P(x)$	probability of an event x
$P(x\|y)$	conditional probability measure in x given y
$p(x)$	probability density function in x
$p(x\|y)$	conditional probability density function in x given y
$\text{logit}(\pi)$	the logarithm of the odds $\pi/(1-\pi)$ of the probability π
$a \ll b$	a is much smaller than b
$a \approx b$	a is approximately equal to b

Fixed Symbols

$\mathcal{D}$	a data set
N	the number of data points in a data set
D	the dimensionality of a data set
α	Lagrange multiplier
λ	regularization constant
C	regularization constant for support vector machines and least squares support vector machines
$\mathcal{F}$	feature space
γ	effective number of parameters
ϑ	a hyperparameter for the prior of a model parameter in the Bayesian framework, e.g. the inverse variance of a prior Gaussian distribution
σ^2	variance
$\mathbf{\Sigma}$	covariance matrix
$\mathcal{N}(\mu, \sigma^2)$	Univariate Gaussian distribution with mean μ and variance σ^2
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$	multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$
$\mathbf{K}$	kernel or Gram matrix
$\mathbf{H}$	Hessian matrix

Acronyms and Abbreviations

ANN	Artificial Neural Network
ARD	Automatic Relevance Determination
AUC	Area Under the ROC curve
BN	Bayesian Belief Network
CDSS	Clinical Decision Support System
CV	Cross-Validation
EM	Expectation Maximization
FDA	Fisher Discriminant Analysis
GRNN	Generalized Regression Neural Network
IOTA	International Ovarian Tumor Analysis
INTERPRET	International Network for Pattern Recognition of Tumors using Magnetic Resonance (IST-1999-10310)
IRLS	Iteratively Re-weighted Least Squares
kPCA	Kernel Principal Component Analysis
kPCR	Kernel Principal Component Regression
KKT	Karush-Kuhn-Tucker
LDA	Linear Discriminant Analysis
Lin	Linear
LR	Logistic Regression
LS	Least squares
LS-SVM	Least Squares Support Vector Machine
MAP	Maximum Posterior
MCMC	Markov Chain Monte Carlo
MLP	Multilayer Perceptron
MRS	Magnetic Resonance Spectroscopy
RKHS	Reproducing Kernel Hilbert Space
OR	Odds Ratio
PCA	Principal Component Analysis
PCR	Principal Component Regression
QP	Quadratic Programming
RBF	Radial Basis Function
RBF-NN	Radial Basis Function Neural Network
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
RVM	Relevance Vector Machine
RFE	Recursive Feature Elimination
RMI	Risk of Malignancy Index
SBL	Sparse Bayesian Learning
SRM	Structural Risk Minimization
SVM	Support Vector Machine
VC	Vapnik-Chervonenkis

Chapter 1

Introduction

Recent advances in technologies and computations have facilitated the collection of biomedical data that can be used for medical decision support. The collected data come from various sources and systems, ranging from patient clinical data, diagnostic images, biological data, to patient genetic profile and population genetic information. This world of information also leads to the development of computer-based tools that are able to handle and analyze the various types of medical data in order to assist the decision making process.

In this introductory chapter, we briefly describe the problem of medical classification for clinical decision making, point out the challenges of development of machine learning systems for clinical decision support. A brief introduction to machine learning, in particular the probabilistic machine learning approaches to classification, is given, outlining some basic concepts in probabilistic modelling. Then the goal of this thesis is stated, followed by a chapter-by-chapter overview. At the end of this chapter, we describe briefly some related work, and outline the major outcome and contributions from this thesis.

1.1 Overview of machine learning

> *"The last five years have seen an explosion in machine learning research. This explosion has many causes: first, separate research communities in symbolic machine learning, computational learning theory, neural networks, statistics, and pattern recognition have discovered one another and begun to work together".*
> *- Thomas G. Dietterich (1997)* [23]

Machine learning, as an area within the field of Artificial Intelligence (AI), refers to application of learning algorithms for the autonomous acquisition and integration of knowledge, resulting in a system capable to learn from experience, analytical observation, and other means. Machine learning techniques have been employed by many disciplines to automate complex decision making and problem solving tasks.

The majority of these tasks are concerned with pattern recognition problems. Pattern recognition is the act of taking in raw data and taking an action based on the "category" or the pattern [24]. In machine learning, the solution of pattern recognition problems lies within the field of supervised learning. The task is e.g. to learn (induce) the relationship between the dependent attributes (input) and designated attribute (output) from a set of examples, i.e. generalize information collected from given data to unseen data. For example, given a set of training data consisting of the measurements of certain features (or variables) from examples of two categories (e.g. patients with malignant tumors and those with benign tumors), a learning system is required to determine the combination of features that is sufficient to distinguish one category (malignant) from the other (benign). Specifically, the system constructs a function $f : \mathbf{x} \rightarrow y$, that maps the given inputs $\mathbf{x}_i \in \mathbb{R}^D, i = 1, 2, \ldots, N$ to outputs y_i, called labels or target values. Each vector $\mathbf{x}_i$ consists of a number of input features which describe the particular case. The target values y_i in such a classification problem denote a class membership for each example case (e.g. malignant vs. benign) and thus $y_i \in \{0, 1\}$ or $y_i \in \{-1, 1\}$.

A number of approaches can be used to solve concept representation problems like this:

- Conventional statistical learning algorithms. The field of statistical pattern recognition is responsible for a large contribution to understanding such problems. Methods such as linear discriminant analysis, generalized linear regression such as logistic regression, and nearest neighbor classification are widely used [24][40].
- Artificial neural networks (ANNs). ANNs are networks of units, called neurons, that exchange information in the form of numerical values with each other via synaptic interconnections, inspired by the biological neural networks of the human brain. They become very powerful and flexible approaches to function approximation [65]. Neural networks can be categorized to three main types: feedforward networks, recurrent networks (feedback networks) and self-organizing networks [42]. In this thesis, ANNs are mainly refer to the feedforward networks such as multilayer perceptrons and radial basis function neural networks, which have been widely used to develop diagnostic models.
- Kernel-based learning algorithms. In recent years, a number of powerful

kernel-based learning algorithms, e.g. support vector machines (SVMs), least squares support vector machines (LS-SVMs) and kernel Fisher discriminant (KFD) analysis have been proposed for both classification and regression [113, 95, 19]. In general, these kernel-based algorithms fall into the category of ANN approaches as well, although they have received more attentions in the recent years. The general idea is to map non-linearly a D-dimensional input space into a high dimensional feature space. A linear classifier (separating hyperplane) is constructed in this high dimensional space to classify the data. The use of the kernel trick allows to construct the classifier in the dual (kernel) space without explicitly knowing the feature space.

- Decision trees. Decision tree algorithms recursively partition the data and give rise to a tree-like structure [65]. The decisions are usually simple attribute tests, using one attribute at a time to discriminate the data. New data can be classified by following the conditions at the nodes down into a leaf. Decision trees have been used extensively in work from both machine learning and statistics.

- Learning sets of rules. Sets of if-then rules can be obtained by converting decision trees. Alternatively, rules can be produced by sequentially searching in the hypothesis space. Inductive logic programming is a collection of machine learning algorithms using first order logic, and the input data can be represented in predicates [65].

- Bayesian networks. This is a statistical and graphical modeling approach, also called Bayesian belief networks [65][24]. Naïve Bayes is the simplest Bayesian network, it is based upon direct application of Bayes Theorem and works under the assumption that the attributes are statistically independent from each other. It is used as a classifier on attribute-value data.

Different approaches have their own particular characteristic benefits and difficulties. For example, decision trees, rule based systems and Bayesian networks are easy to understand for human beings, normally referred to as white-box models, however, they are not good at dealing with continuous variables. ANNs and kernel-based models have powerful learning abilities and can usually achieve a better accuracy in classification tasks than the former white-box models; however, the interpretation of the model is not easy, and usually referred to as blackbox models. In this thesis, we are concentrating on *the blackbox model approaches such as ANNs and kernel-based models, because a good performance is the key for an intelligent system to be accepted for clinical practice, and further study could done in order to open the blackbox models.*

Generalization capability is crucial for the learning systems. Overfitting occurs when the algorithms fit too closely the particular cases in the training data,

including some noise, and thus fails to generalize to unseen cases. This problem can be tackled by restricting the flexibility (capacity) of the models. However, underfitting might occur when the capacity of the model is restricted more than required. Controlling the capacity or complexity of the models usually involves the fine tuning of a number of model-dependent (hyper)parameters.

Because of the ability to identify patterns or trends in data, machine learning methods have been successfully applied in areas such as electrical consumption prediction, industrial process control, risk management, fault detection, speech recognition, text categorization, handwriting character recognition, etc. In this thesis, we applied the machine learning methods in medicine and in particular for diagnosis of diseases. The following section attempts to give a brief description of the role of machine learning in clinical decision support systems.

1.2 Clinical decision support systems

> *Computer based diagnostic decision support systems will play an increasingly important role in health care. Health care will change profoundly with the introduction of clinical diagnostic decision support systems, preferably integrated with electronic patient data and together with online computer communication.*
> – *Y. Reisman, 1996* [81].

Computer based decision support systems are considered to be important in many fields of medicine. For example, human beings, even human experts, have limitations when diagnosing diseases, mainly because this process is subjective, and largely depends on the experiences of the assessor and her/his interpretation on the patient signals. Since 1970s, the application of computer systems based upon the technologies of artificial intelligence (AI) in the medical domain has been gaining popularity [66][17][62][7]. There exist already computer programs that could assist clinicians in the process of diagnosis, and they are now in routine use in many clinical situations [16]. AI techniques can also be applied to create systems able to assist with therapy planning, information seeking, and the alert generation, patient monitors, etc. The use of "-omics" technologies such as microarrays in human genome analysis, has urged the use of computer techniques, and reframes many classes of clinical decision [67]. Such computerized intelligent systems using explicit knowledge to generate patient specific advice or interpretation are typically described as clinical decision support systems (CDSS).

The most common type of CDSS in routine clinical use are expert systems, also known as knowledge-based systems. They contain clinical knowledge typically represented in sets of rules, usually about a very specifically defined task, and are able to reason with data from individual patients to come up with reasoned

conclusions. Machine learning is on the other hand focusing on development of systems that can learn from experience and discover new knowledge. Learning techniques include neural networks and a large variety of other methods as well. For example, some systems are able to learn decision trees from examples taken from data, which can be used to help in diagnosis.

Healthcare has formed a rich test-bed for machine learning experiments in the past, allowing scientists to develop complex and powerful learning systems. While there has been much practical use of expert systems in routine clinical settings, at present machine learning systems still seem to be used in a more experimental way. There are, however, many situations in which they can make a significant contribution.

One direction is to use machine learning systems to model some of the human actions or thinking processes and recognizing diseases from a variety of input sources (e.g. cardiograms, CAT/MRI/ultrasound scans, etc.). One can use machine learning methods to develop the knowledge bases (expressed in rules or in decision trees) used by expert systems. A classical example of this type of system is KARDIO, which was developed to interpret ECGs [13]. Alternatively, quantitative models can be built for diagnosis based upon various types clinical information and different tests performed upon the patient, e.g. neural networks have been applied to breast cancer diagnosis [45]. In the poorly understood areas of healthcare, people usually describe the machine learning system as "data mining" and "knowledge discovery" systems [17][72]. If used in a research setting, these models can serve as initial hypotheses that can drive further experimentation.

We would like to emphasize here that, no decision support system could or should replace medical judgment, as it is unlikely that any system could cope with the complexity of patient interactions with medical care. The purpose of these systems is to assist clinicians, not replace them. The challenges for implementing a competent decision support system are definitely not easy to overcome, however with further advancement in technology and human knowledge, an intelligent clinical decision support tool may indeed not be too far away.

1.3 Medical classification problems

Clinical decision making (such as treatment planning) for individual case is a complex process, which involves various aspects such as multiple diseases and disease causes, cost, physical and psychological situation of a patient, etc. Because of the difficulty (or infeasibility) in developing large models from the limited data while taking into account so many aspects, most intelligent systems in the medical context have focused on tightly constrained diagnostic problems,

which is called "classification problems". For example, a problem that has received considerable attention is whether a specific tumor of a patient is benign or malignant. Another example is to predict the outcome of an early pregnancy of unknown location, which is either a normal pregnancy, a failing one or a life threatening ectopic pregnancy [18]. In this thesis, we only deal with the medical classification problem, which is the essential part in clinical decision making.

Now we give a more complete description of classification. Classification has two distinct meanings. We may be given a set of observations with the aim of establishing the existence of classes or clusters in the data. Or we may know for certain that there are a certain number of classes, and the aim is to establish a rule whereby we can classify a new observation into one of the existing classes. The former type is known as unsupervised learning (or clustering), the latter as supervised learning. In this thesis, we only deal with the restricted classification, i.e. the supervised learning.

1.4 Problem challenges

The development of an intelligent clinical decision support system is a very demanding task. Challenges come from the stage of data collection, e.g. on how to integrate heterogenous patient information ranging from patient history to physical exams, from drug prescriptions to lab results. It is also a challenge to implement a system that fits in the way clinicians reason, think and work. In addition, there are difficulties associated with the medical field, such as policy and financial constraints. However, these are the topics outside the scope of this thesis and our major research interest. Beyond that, we are still facing a couple of challenges related to the predictive modeling and knowledge extraction.

No simple yes or no answers. Most decision-support systems use probabilistic methods as their underlying approach for solving medical diagnosis problems, such as Bayes' theorem and decision analysis. However one of the constraints on these models is the conditional variable. For example the Bayesian model assumes conditional independence of symptoms (i.e. a patient has one disease at a time), but this is rarely the case for most of medical diagnosis problems. There are other systems that use a rule-based approach. Their primary constraint is that the rule set usually makes rigid and binary decisions. However, in medical diagnosis, there are really no simple yes and no answers, therefore not applicable in complex situations.

Uncertainty. "The basic problem of diagnosis is to take a set of effects and determine the cause. Actually the problem is better characterized as looking for an ordered list of possible causes, since some degree of uncertainty

is a nearly constant feature of medicine." [51] The human body is extremely complex, only partially understood. The underlying mechanism of the disease is only partially known. Additionally, subjective effects, noise and errors are also unavoidable in most diagnostic tests. There is often no way to be certain of a diagnosis, even the gold standard results involve uncertainty. Probabilistic models are suitable for reasoning under uncertainty, and thus are preferably used in medical decision support systems.

Validation of the models. An intelligent clinical decision support system (CDSS) is a term which covers many types of intelligent systems that can be applied in the medical field. Clinicians see them as black boxes and the safety critical nature of the domain requires that they should be thoroughly evaluated before entry into clinical practice [91]. First of all, the underlying model must perform well. This is the issue of evaluation and standards of performance. However, we have learned that evaluation of decision support system is difficult and the standards for doing so are only slowly emerging. For example, cross-validation with randomization is popular in machine learning community, however a prospective test is desired in clinical practice. Which criteria should be used in evaluation? Obviously, accuracy, the number of correct classifications is too naive here in the biomedical field since the environment in which we tend to use them is very complex and more aspects should be taken into account.

Curse of dimensionality. Another aspect of the challenge lies in the mathematical approaches to the classification problem, especially in the aspect of knowledge extraction. One difficulty for machine learning approaches to medical classification problems is the curse of dimensionality. The curse of dimensionality is a phenomenon in modelling which says that the number of training data required for a reliable estimate grows exponentially with the number of required parameters be estimated. There are many manifestations of this problem, we have only given an example here. This also motivates the use kernel-based models, in which primal and dual representation could be important in this context.

Curse of data set sparsity. The curse of dataset sparsity [92] occurs more often recently in the post genome age, when the high throughput techniques enable the collection of a large amount of information (such as the gene expression levels) simultaneously in one subject. The data set collected usually has a huge dimensionality (from hundreds to tens of thousands), while contains only a relatively small number of data points (typically around one hundred). This motivates the use of bootstrapping, model ensembles in order to stabilize the modelling process. This also poses challenges to the machine learning researchers for finding suitable algorithms to do analysis and predictive modelling using these data sets.

1.5 Probabilistic machine learning approaches to classification

Probabilistic machine learning approaches use probability theory in the learning procedure (or parameter estimation), in inferences, and in outcome prediction which should be associated with variance or a probability value to indicate the uncertainty in the prediction. We will taste some flavors of the probabilistic framework in this chapter, and will reveal more experience and details with respect to its computational side and its advantages in biomedical applications in the next four chapters.

1.5.1 Bayesian decision theory

A fundamental statistical approach to the problem of pattern recognition or classification is Bayesian decision theory. This approach is based on quantifying the tradeoffs between various classification decisions using probability and the costs accompanied with such decisions [24]. It plays a role when there is some a priori information about what we are trying to classify. It assumes that all the relevant probability values are known in the decision or classification problem. Let us take the same example described in Section 1.1. The distributions of examples in each class are known, i.e., we know the class-conditional density functions: $p(\mathbf{x}|y=-1)$ and $p(\mathbf{x}|y=1)$. And we might also know the prior probability for a case belonging to one specific category $P(y=-1)$ and $P(y=1)$, which can also be estimated using the number of occurrences of one category among all examples. The question is: how to decide to which class a new data point should belong.

Let us first examine the Bayes formula:

$$P(y=\pm 1|\mathbf{x}) = \frac{p(\mathbf{x}|y=\pm 1)P(y=\pm 1)}{p(\mathbf{x})}, \tag{1.1}$$

where $p(\mathbf{x}) = \sum_{y' \in \{-1,+1\}} p(\mathbf{x}|y')P(y')$. And this Bayes formula can also be expressed informally by

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}. \tag{1.2}$$

The Bayes formula shows that by observing the value of $\mathbf{x}$ we can convert the *prior* probability $P(y=\pm 1)$ into a *posterior* probability $P(y=\pm 1|\mathbf{x})$. We call $p(\mathbf{x}|y=\pm 1)$ the *likelihood* of $y=\pm 1$ with respect to $\mathbf{x}$. The *evidence* factor $p(\mathbf{x})$ can be viewed as a normalization factor to guarantee the posterior probabilities sum to one. It can be justified that the optimal decision minimizes

the misclassification rate and should be:

$$\text{Decide } y = +1, \text{ if } P(y = 1|\mathbf{x}) > P(y = -1|\mathbf{x}); \text{ otherwise decide } y = -1. \quad (1.3)$$

Notice that the evidence $p(\mathbf{x})$ is just a normalization factor to ensure that $P(y = 1|\mathbf{x}) + P(y = -1|\mathbf{x}) = 1$. Thus after eliminating $p(\mathbf{x})$, we obtain the equivalent decision rule as follows:

$$\begin{aligned} &\text{Decide } y = +1, \text{ if } p(\mathbf{x}|y = 1)P(y = 1) > p(\mathbf{x}|y = -1)P(y = -1); \\ &\text{otherwise decide } y = -1. \end{aligned} \quad (1.4)$$

Furthermore, we can also make a decision rule which minimizes risk (loss) rather than merely the probability of error if the loss function is given. The loss function states how costly each decision (action) is, and is used to convert a probability determination into a decision. Let c_-^+ and c_+^- denote the cost of misclassifying a case (and taking a certain action accordingly) from class '-1' and '$+1$' respectively. And c_-^- and c_+^+ refer to the costs of correct classification for class '-1' and '$+1$', respectively, which are usually ignored by giving a value of zero though. The minimum-risk decision rule is thus to decide $y = +1$, if

$$(c_+^- - c_+^+)P(y = 1|\mathbf{x}) > (c_-^+ - c_-^-)P(y = -1|\mathbf{x}). \quad (1.5)$$

Equivalently, the minimum-risk decision rule can also be to decide $y = +1$, if

$$(c_+^- - c_+^+)p(\mathbf{x}|y = 1)P(y = 1) > (c_-^+ - c_-^-)p(\mathbf{x}|y = -1)P(y = -1). \quad (1.6)$$

1.5.2 Probabilistic classifiers

There are in general two approaches to classification [83][73]. Generative (or informative) approaches learn models of the joint probability $p(\mathbf{x}, y)$, of the inputs $\mathbf{x}$ and the label y, and make their predictions by using Bayes' rule to calculate $P(y|\mathbf{x})$. Discriminative classifiers model the posterior class probabilities $P(y|\mathbf{x})$ directly, or learn a direct map from inputs $\mathbf{x}$ to the class labels. One of the compelling reasons for using discriminative rather than generative classifiers, has been articulated by Vapnik [113]: "one should solve the (classification) problem directly and never solve a more general problem as an intermediate step (such as modelling $p(\mathbf{x}|y)$)." There exist also some hybrid classifiers which combine both the generative and discriminative approaches .

Generative approach to classification

In a two-class classification problem, we are given a training data set $\mathcal{D} = \{\mathbf{x}_i, y_i\}, i = 1, \ldots, N$, where $\mathbf{x}_i \in \mathbb{R}^D$, is the input and $y_i = \pm 1$ is the output

label. In generative paradigm, the classification problem is effectively reduced to that of modelling the class-conditional probability distribution $p(\mathbf{x}|y = \pm 1)$ for the two classes, and then Bayes' rule (Equation 1.1) is employed to obtained the posterior probabilities for the final decision. Typical methods using generative approaches to classification include naïve Bayes, linear discriminant analysis.

Very often a certain model is chosen for the class densities, for example a Gaussian model, $p_\theta(\mathbf{x}|y = \pm 1) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\pm, \boldsymbol{\Sigma})$, with $\theta = \{\boldsymbol{\mu}_+, \boldsymbol{\mu}_-, \pi_+, \pi_-, \boldsymbol{\Sigma}\}$ corresponding to the mean ($\boldsymbol{\mu}_\pm$), prior class probability ($\pi_\pm$) for class '+1' and class '−1', and the covariance matrix for the input $\mathbf{x}$ ($\boldsymbol{\Sigma}$), respectively.

It is sometimes convenient to write a discriminant function, which is expressed in the form of a logarithm of likelihood ratio:

$$g(\mathbf{x}) = \log \frac{P(y = 1|\mathbf{x})}{P(y = -1|\mathbf{x})}. \tag{1.7}$$

The decision rule is then to decide class + if $g(\mathbf{x}) > 0$, otherwise decide class −. Therefore, the discriminant function for the above gaussian density model is:

$$\begin{aligned} g(\mathbf{x}) &= (\log \frac{\pi_+}{\pi_-} - \frac{1}{2}(\boldsymbol{\mu}_+ + \boldsymbol{\mu}_-)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)) + (\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)\boldsymbol{\Sigma}^{-1}\mathbf{x} \quad (1.8) \\ &= b + \mathbf{w}^T\mathbf{x} \quad (1.9) \end{aligned}$$

However, it has always been difficult to model $p(\mathbf{x}|y = \pm 1)$ accurately. The advantages of a generative approach is that if it is more efficient if the model assumption is correct, since it makes use of the marginal distribution $p(\mathbf{x})$. On the other hand, it could be biased if the model assumption is incorrect.

Discriminative approach to classification

The discriminative approach models the class posteriors and the discriminants directly. This approach is more flexible in the sense that it makes fewer assumptions about the classes and is generally more robust against outliers and noise in the data. However they provide less insight into the structure of the data space and it is difficult to handle data containing missing values. Logistic regression is a well known example of the discriminative approach and is widely used in medical research. Decision trees are another kind of discriminative classifier. Recently, attentions has shifted to neural network type classifiers, and support vector machines (SVM).

In some of these classifiers, the estimation of the posterior probabilities is unnecessary. The classifier simply returns the label y by applying discrimination

functions on the input $\mathbf{x}$. However, in medical diagnosis, decisions are usually taken by combining the output of the model with other source of information. In addition, some of the cases that are ambiguous have to be rejected for further examination. In this context, it is important to provide not only an indication of the most plausible class, but also a faithful description of the plausibility (very often in form of probability) of various hypotheses regarding the classes under consideration. Therefore we will always try to derive the class probabilities from the latent output of the discriminant.

There exist also approaches which combine both the generative and discriminant approaches. Take a simple example, least squares SVM classifiers are originally discriminative classifiers which generate a one-dimensional latent output u rather than posterior class probabilities. By performing generative classification (sort of normal discriminant analysis) on the transformed data $\mathcal{D}_u = \{u_i, y_i\}_{i=1}^N$, we could finally obtain the class posterior probabilities.

1.5.3 Learning and parameter estimation

There are some other general issues which are independent of the types of models in machine learning. One issue is parameter estimation. There are mainly two approaches: maximum likelihood and Bayesian parameter estimation [24].

The problem of parameter estimation is a classical one in statistics, and it can be approached in several ways. We will consider two common and reasonable procedures, namely, maximum likelihood estimation and Bayesian estimation. Although the results obtained with these two procedures are frequently nearly identical, the approaches are conceptually quite different. Maximum likelihood and several other methods view the parameters as quantities, of which the values are fixed but unknown. The best estimate of their value is defined to be the one that maximizes the probability of obtaining the samples actually observed. In contrast, Bayesian methods view the parameters as random variables having some known prior distributions. Observation of the samples converts this to a posterior density, thereby revising our opinion about the true value of the parameters.

In the Bayesian case, we shall see that a typical effect of observing additional samples is to sharpen the a posteriori density function, causing it to peak near the true values of the parameters. This phenomenon is known as Bayesian learning.

Linear discriminant analysis and logistic regression are two classical probabilistic models for classification. Some advanced models in machine learning including artificial neural networks, kernel based models such as SVMs and LS-SVMs, usually have a more complex structure and thus a stronger learning capability. By incorporating them with Bayesian framework, both the learn-

ing procedure and the predictive output of the models become probabilistic. Therefore, we could make use of such advanced models within the Bayesian framework and try to achieve the desired property of a model for medical diagnosis, i.e. to take the uncertainty in parameter estimation into account, and to provide a probability for a class label.

1.6 Goal of the thesis

This work was basically aimed as an application based research. During the course of investigation, it became clear that a more extensive theoretical investigation was needed on a number of existing learning algorithms and how to adapt them for obtaining a reliable and stable diagnostic system. This leads us to the investigation of a series of techniques that help in understanding, controlling and evaluating the behavior of a learning system.

The overall aim of this study described in this dissertation is to present a comparative study between the classical linear and nonlinear classifiers such as logistic regression and artificial neural networks, and the more recently emerging kernel-based classifiers particularly within a Bayesian framework. Meanwhile, the ultimate goal of this work is to develop predictive computer models for clinical decision support on several medical diagnosis problems. Many of our efforts have been focused on models discriminating between malignant and benign ovarian tumors based on the clinical patient data. On the other hand, we considered also how to extend the existing learning algorithms for classification based on other sources of data, including genomic and metabolic data, which are typically characterized by a very high dimensionality and a relatively small sample size.

1.7 Structure of the thesis

The rest of the thesis is organized in the following way.

Chapter 2 aims at giving an overview of the supervised learning approaches for classification. We start with an introduction of a generic model structure and the use of regularization for parameter estimation. Then two conventional linear classification techniques are introduced, including linear discriminant analysis and logistic regression. Moving to some more advanced classification techniques, we first review artificial neural networks, particularly multilayer perceptrons (MLPs) and radial basis function neural networks (RBF-NNs), then the kernel-based models which have received much more attention in this thesis. Support vector machines (SVMs), and least squares support vector

machines (LS-SVMs) have been described, and their links to the generalized additive models have been pointed out. At the end of this chapter, we also cover some important topics in modelling including variable selection, multi-class classification strategies, ensemble methods and model validation.

Chapter 3 is dedicated to Bayesian learning methods for blackbox models, particularly MLPs, models linear in parameters such as RBF-NNs, and LS-SVMs. After a brief overview of the Bayesian learning and its application on MLPs, we focus on Bayesian kernel-based classifiers. We investigated the sparse Bayesian learning methods for linear parametric modelling using fixed basis functions as inputs, where the sparseness is obtained via automatic relevance determination. Particularly, when taking the kernel basis functions, we obtain relevance vector machines which in general use very few data points in form of kernel-based models; when taking the input variables as basis functions, we reach a sparse logistic regression (or linear regression in case of a regression problem) model with an automatic variable selection mechanism.

The LS-SVM enjoys the primal-dual interpretation, the use of the least squares cost function during training, and the intrinsical use of regression paradigm, and thus allows an analytical expression of the posterior for the (hyper)parameters and the predictions within a Bayesian framework. We propose also a procedure of iteratively pruning the training data so as to impose sparseness to the Bayesian LS-SVM classifiers. This leads to the training procedure as a mimic of the margin maximization as in standard SVM classifiers, and can usually improve the generalization performance of the LS-SVM classifiers. A variable selection scheme is also proposed based upon the evidence (posterior probability) of the Bayesian LS-SVM.

Chapter 4 presents the application of the aforementioned probabilistic classification techniques on classification of ovarian tumors. We first describe the initial study on this problem of diagnosis of ovarian cancer, using the data collected from a single clinical center at K.U. Leuven. Analysis encompasses data exploration, variable selection, model building and validation. We present the results from the Receiver operating characteristic (ROC) analysis. Carried with the experiences and some initial conclusions from the first study, we performed an extensive study on modelling for ovarian tumor classification using the data set from the IOTA project (an international multi-center study on the ovarian tumors). We present a prospective validation of our models built from the previous study on the K.U. Leuven pilot project. And then the development and validation of the series probabilistic models using the IOTA data set were reported and discussed.

Chapter 5 concerns variable selection and modelling for cancer diagnosis using data sets with very high dimensionality (with hundreds and thousands of variables), and relatively small sample size. The sequential sparse Bayesian learning algorithm applied on the linear logit models was taken as the basic

variable selection algorithm, and predictive models were built using probabilistic kernel-based classifiers. A bagging strategy has been proposed both for variable selection, model building and prediction. We tested the methods on two benchmark binary classification problems for cancer diagnosis based on microarray data, and one multi-class brain tumor classification problem using Magnetic Resonance Spectroscopy (MRS) data. It is demonstrated that the stability as well as the performance of the model has been significantly improved by use of bagging.

Chapter 6 summarizes the thesis and suggests the topics for further research.

1.8 Contributions and related work

Contributions

The outcome of the thesis is twofold. With respect to machine learning, it is the development, evaluation and implementation of probabilistic machine learning algorithms to clinical and genomic data sets for predictive modeling. With respect to real life applications, it is the diagnostic models ready for test in the clinical practice and the interpretation of them.

- Based on previous work [94][95] [96][110][111] (by Suykens, Van Gestel, et al.) on Bayesian least Squares support vector machines (LS-SVMs), we further proposed a simple pruning procedure to impose sparsity to the LS-SVM models. We demonstrated that the sparse Bayesian LS-SVMs are able to improve the generalization performance of the model. We examined and studied the properties of the variances associated with sparse LS-SVM modelling. This is described in Chapter 3. Some of the results have been published in [59].

- We designed and implemented the Bayesian learning systems for LS-SVM and relevance vector machines, in which the parameters for the kernels or basis functions can be tuned automatically. The least human interaction is involved in the hyperparameter tuning and parameter estimation process. This is explained in Chapter 3 and has been previously reported in [54].

- We pointed out in Chapter 2 the relationship between the kernel based models and the generalized additive models by use of additive univariate kernels. To enhance the interpretability of the kernel-based models, additive kernels, i.e. component wise kernels, can be employed, allowing us to visualize and interpret the outcome of the models. This also provides a tool to analyze the impact of each variable to the final outcome.

- We demonstrated that the decisions of LS-SVM classifiers can be roughly visualized by means of kernel principal regression and nonlinear biplots. We have illustrate this visualization methods in an early publication [56], which are explained in Chapter 2 with an toy example.

- We proposed a stable forward stepwise feature (input variable) selection scheme using Bayesian LS-SVM classifier by maximizing the marginal likelihood, which is proved to be of great value in feature selection and marker identification for our real life medical classification problems. This method is described in Chapter 3, and has been published in [59, 57, 56].

- We implemented an efficient incremental feature selection approach on high dimensional data sets, such as on gene expression data. A bagging strategy was proposed both for variable selection and classification, which can stabilize and improve the performance of the single classifiers. A detail report on this study is given in Chapter 5, and was previously presented in [54].

- We have created and evaluated a series of computational models and help to find out and understand the risk factors for the diseases. Our most important applications are the prediction of malignancy of ovarian tumors, using the data collected at University Hospitals Leuven and a large multicenter data set from the International Ovarian Tumor Analysis (IOTA) project. Chapter 4 presents the results of this application in detail. Parts of the results have already been published in [53, 55, 59, 58, 57, 56]. The other case studies include the brain tumor classification using magnetic resonance Spectra from the INTERPRET project [31][22], and some benchmarking problems such as leukemia and colon cancer diagnosis based on microarray gene expression data. These results are shown in Chapter 5, and have been reported in [54].

Related work

The work in this thesis has been based upon previous work and it is the result of the collaboration of a couple of units and people. Parts of the material presented in this thesis has been previously published in [53, 55, 59, 58, 57, 56, 54]. Sabine Van Huffel and Johan Suykens, as my supervisors, have provided continuous guidance, inspirations and critical scientific comments throughout the work.

Background introduction on machine learning and supervised learning techniques are given in Chapter 1 and 2. The idea of using additive kernels for generalized additive kernel models, described in Chapter 2, is related to the discussion with Aleks Jakulin and the componentwise LS-SVMs introduced in [77] as well. Visualization of LS-SVMs via kPCR and nonlinear biplots are

the result of the collaborative work with Tony Van Gestel [56]. Most materials in Chapter 3 have been published in [59][57], the development of Bayesian framework for LS-SVM classifier including variable selection and sparseness approximation has been done in collaboration of Tony Van Gestel. The implementation of sequential sparse Bayesian learning algorithms was benefited from the helpful comments from Mike Tipping.

The initial investigation of the ovarian tumor data set reported in Chapter 4 has been published in [59, 58, 57]. Exploratory data analysis of the ovarian tumor data from the K.U. Leuven pilot project was done in collaboration with Jos De Brabanter [53]. And the statistical analysis and preprocessing of the IOTA data set prior to model development was primarily done by Lieveke Ameye [102]. Dirk Timmerman has provided us the expert knowledge and guidance on the ovarian cancer diagnosis problem, on selection of the variables and interpretation of the models from a clinical point of view. Contents of Chapter 5 has been previously reported in [54], the application of bagging strategy for variable selection and model building in brain tumor classification is the joint work mainly with Andy Devos who has preprocessed the data set and helped to interpret the selected variables.

Chapter 2

Supervised learning

In this chapter, methods of supervised learning, especially for classification have been reviewed. Many important topics in supervised learning are covered in this chapter, including model structures, parameter estimation, model complexity control, variable selection, model ensemble, strategy for mulitclass classification and model validation. The contents are limited to the methods that have been investigated in this thesis work. We start with the introduction of conventional linear classifiers including linear discriminant analysis and logistic regression, followed by a quick review of feed-forward artificial neural networks. Then some theory and algorithms for kernel-based classifiers are described, with focusing on support vector classifiers, and least-squares support vector machine classifiers.

2.1 A generic supervised learning model and regularization

Supervised learning infers a functional relation $y \Leftrightarrow f(\mathbf{x})$ from a training data set composed of N independent, identically distributed (i.i.d.) samples $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$. Considering scalar-valued target function only, in the presence of additive noise, a generic regression model can be written as [61, 106]

$$y_n = f(\mathbf{x}_n) + \epsilon_n, \tag{2.1}$$

where $x_n \in \mathbb{R}^D$, $y_n \in \mathbb{R}$, and ϵ_n is i.i.d. random variable. And ϵ_n is assumed to follow a Gaussian distribution with mean zero and variance σ^2.

Typically, the predictive functions $f(\mathbf{x})$ can be defined in the input space $\mathbf{x}$:

$$f(\mathbf{x};\mathbf{w}) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}). \tag{2.2}$$

where the output is a linearly weighted sum of M generally fixed basis functions for input $\mathbf{x}$, $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_M(\mathbf{x})]^T$. The weights in the model $\mathbf{w}$ are usually estimated by minimizing the regularized cost function:

$$\mathcal{J}(\mathbf{w}) = E_D(\mathbf{w}) + \lambda E_W(\mathbf{w}) \tag{2.3}$$

The error term E_D is usually a sum of squared errors

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left[y_n - f(\mathbf{x}_n;\mathbf{w}) \right]^2 . \tag{2.4}$$

While the weight penalty term is usually the sum of squared weights:

$$E_W(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}. \tag{2.5}$$

The regularizer λ determines the amount of regularization, i.e., a trade-off between the model fitness and the smoothness.

Let $\mathbf{y} = (y_1, \cdots, y_N)^T$, and $\boldsymbol{\Phi}$ as the $N \times M$ design matrix with element $\boldsymbol{\Phi}_{nm} = \phi_m(\mathbf{x}_n)$. Then the regularized (penalized) least squares estimate, i.e., the so-called ridge regression (RR) estimate for the weights is given by

$$\mathbf{w}_{\mathrm{RR}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \mathbf{I})^{-1} \boldsymbol{\Phi}^T \mathbf{y}. \tag{2.6}$$

In this probabilistic model , the conditional distribution of the target variable given the input variable and the model parameters is again a Gaussian. Thus the likelihood of the data set, under the assumption of i.i.d. data, can be written as (note that $\mathbf{x}$ will be omitted from the conditioning list for notational simplicity):

$$p(\mathbf{y}|\mathbf{w},\sigma^2) = \prod_{n=1}^{N} p(y_n|\mathbf{w},\sigma^2) = \prod_{n=1}^{N} (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left[-\frac{(y_n - f(\mathbf{x}_n;\mathbf{w}))^2}{2\sigma^2} \right]. \tag{2.7}$$

One can easily verify that, the maximum likelihood estimate for $\mathbf{w}$ actually tends to minimize the negative logarithm of the likelihood $-\log p(\mathbf{y}|\mathbf{w},\sigma^2)$, which is identical to the least squares solution.

Instead of the regularization, we can also control the complexity of the model via a prior distribution which normally expresses our preference to a smooth model. It is common to assume $p(\mathbf{w}|\vartheta) \propto \exp(-\vartheta E_W))$ with ϑ as a hyperparameter in a probabilistic framework. As a specific example, we might choose a Gaussian distribution for $p(\mathbf{w}|\vartheta)$ of the form

$$p(\mathbf{w}|\vartheta) = \left(\frac{\vartheta}{2\pi}\right)^{M/2} \exp\left(-\frac{\vartheta}{2}\mathbf{w}^T\mathbf{w}\right). \tag{2.8}$$

In the MAP (maximum posterior) approach, the parameters are sought to maximize the the posterior probability $p(\mathbf{w}|\mathbf{y}, \vartheta, \sigma^2) \propto p(\mathbf{y}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\vartheta)$, or to minimize the negative log-posterior cost function $-\log p(\mathbf{y}|\mathbf{w}, \sigma^2) - \log p(\mathbf{w}|\vartheta)$. Using (3.9) and (2.8), we see that maximizing the posterior probability is equivalent to minimizing $\frac{1}{2\sigma^2}E_D + \frac{\vartheta}{2}E_W$. Therefore, the regularized least squares approach can be viewed as a special case of the MAP technique, and the regularizer $\lambda = \sigma^2\vartheta$.

For binary classification models which apply logistic sigmoid link function $g(a) = 1/(1 + e^{-a})$ to $f(\mathbf{x})$, the computation of the likelihood is based on the Bernoulli distribution:

$$p(\mathbf{y}|\mathbf{w}) = \prod_{n=1}^{N} g(f(\mathbf{x}_n; \mathbf{w}))^{y_n}[1 - g(f(\mathbf{x}_n; \mathbf{w}))]^{1-y_n}, \tag{2.9}$$

where the target $y_n \in \{0, 1\}$, and no hyperparameter σ^2 is involved. The priors for the weights are kept the same as for the regression case. Thus the cost function to be minimized turns out to be $-\log p(\mathbf{y}|\mathbf{w})) + \frac{\vartheta}{2}E_W$. To find the maximum likelihood or MAP estimates based on the logistic likelihood function, the iteratively reweighted least squares (IRLS) algorithm is often used [71, 106, 9].

2.2 Conventional linear classifiers

2.2.1 Linear discriminant analysis

A widely used way to represent pattern classifiers is in terms of a set of discriminant functions $g_i(\mathbf{x}), i = 1, \ldots, K$ (K is the number of classes, and C_i corresponds to the class label for the ith class). A classifier assigns an input vector $\mathbf{x}$ to C_i if $g_i(\mathbf{x}) > g_j(\mathbf{x}), \forall i \neq j$.

Linear discriminant for normal densities

Let us consider binary discriminant functions defined in terms of the natural logarithm of likelihood:

$$g_{\pm}(\mathbf{x}) = \log p(\mathbf{x}|y = \pm 1) + \log P(y = \pm 1) \tag{2.10}$$

One can evaluate this discriminant function directly if the class conditional densities are multivariate normal, i.e. if $p(\mathbf{x}|y = \pm 1) \sim \mathcal{N}(\boldsymbol{\mu}_{\pm}, \boldsymbol{\Sigma}_{\pm})$. In a simplified case when the covariance matrices for both classes are identical, i.e. $\boldsymbol{\Sigma}_+ = \boldsymbol{\Sigma}_- = \boldsymbol{\Sigma}$, the discriminant function turns out to be

$$g_{\pm}(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{\pm})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{\pm}) + \log P(y = \pm 1). \tag{2.11}$$

If the prior probabilities $P(y = +1) = P(y = -1)$, then the $\log P(y = \pm 1)$ term can be ignored. This leads to an optimal decision rule based on the squared Mahalanobis distance $(\mathbf{x} - \boldsymbol{\mu}_{\pm})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{\pm})$ from the $\mathbf{x}$ to each of the mean vectors, and $\mathbf{x}$ is assigned to the class of the nearest mean.

Another representation for such a binary discriminant for normal densities has already been mentioned in (1.8). A boundary between class + and − has the equation

$$\mathbf{w}^T \mathbf{x} + b = 0, \tag{2.12}$$

where
$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) \tag{2.13}$$

and
$$b = \log \tfrac{P(y=+1)}{P(y=-1)} - \tfrac{1}{2}(\boldsymbol{\mu}_+ + \boldsymbol{\mu}_-)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-). \tag{2.14}$$

Fisher's linear discriminant

Fisher's linear discriminant (Fisher, 1936) [24] is a classical discriminant analysis. Suppose that we have a set of N D-dimensional samples $\mathbf{x}$, N_- data in the subset of class "−" $\mathcal{D}_-$ and N_+ in the subset of class "+" $\mathcal{D}_+$. If we form a linear combination of the components of $\mathbf{x}$, we obtain the scalar dot product

$$y = \mathbf{w}^T \mathbf{x} \tag{2.15}$$

The goal is to find the best directions defined by $\mathbf{w}$, which enable accurate classification. Geometrically, if $\|\mathbf{w}\| = 1$, each y_i is the projection of $\mathbf{x}_i$ onto a line in the direction of $\mathbf{w}$.

The Fisher linear discriminant seeks for a linear function $\mathbf{w}^T \mathbf{x}$ maximizing the ratio between the between-class scatter and within-class scatter:

$$\max_{\mathbf{w}} \mathcal{J}(\mathbf{w}) = \frac{|\tilde{m}_+ - \tilde{m}_-|^2}{\tilde{s}_+^2 + \tilde{s}_-^2}, \tag{2.16}$$

where $\tilde{m}_\pm$ denote the sample mean of the projected one-dimensional data $(\mathbf{w}^T\mathbf{x})|_{\mathbf{x}\in\mathcal{D}_\pm}$ for class "+" and "−", while $\tilde{s}_\pm$ denote the scatter for the projected sample for class "+" and "−", respectively.

In order to express $\mathcal{J}(\cdot)$ in terms of $\mathbf{w}$, we define the following mean vectors and scatter matrices. Let $\mathbf{m}_+$ and $\mathbf{m}_-$ denote the D-dimensional sample mean given by $\mathbf{m}_\pm = \frac{1}{N_\pm}\sum_{\mathbf{x}\in\mathcal{D}_\pm}\mathbf{x}$. And define the scatter matrices $S_\pm$, S_W and S_B by

$$\mathbf{S}_\pm = \sum_{\mathbf{x}\in\mathcal{D}_\pm}(\mathbf{x}-\mathbf{m}_\pm)(\mathbf{x}-\mathbf{m}_\pm)^T, \tag{2.17}$$

the within-class scatter matrix

$$\mathbf{S}_\mathrm{W} = \mathbf{S}_+ + \mathbf{S}_-, \tag{2.18}$$

and the between-class scatter matrix

$$\mathbf{S}_\mathrm{B} = (\mathbf{m}_+ - \mathbf{m}_-)(\mathbf{m}_+ - \mathbf{m}_-)^T. \tag{2.19}$$

The criterion function $\mathcal{J}(\cdot)$ can then be written in terms of $\mathbf{S}_\mathrm{B}$ and $\mathbf{S}_\mathrm{W}$ by

$$\mathcal{J}(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_\mathrm{B}\mathbf{w}}{\mathbf{w}^T\mathbf{S}_\mathrm{W}\mathbf{w}}. \tag{2.20}$$

This is the well known generalized Rayleigh quotient. One can then obtain a solution for the $\mathbf{w}$ that maximize $\mathcal{J}(\cdot)$:

$$\mathbf{w} = \mathbf{S}_\mathrm{W}^{-1}(\mathbf{m}_+ - \mathbf{m}_-). \tag{2.21}$$

Let us assume that the conditional densities $p(\mathbf{x}|y=\pm1)$ are multivariate normal with equal covariance matrices $\mathbf{\Sigma}$. As we can see from (1.8) and (2.12), if $\boldsymbol{\mu}_i$ and $\mathbf{\Sigma}$ are estimated by use of sample means and covariance, the vector $\mathbf{w}$ in the discriminant function for the Gaussian density model (1.8) is just the same direction obtained by maximizing the Fisher's criterion $\mathcal{J}(\cdot)$. To do classification based on the Fisher's linear discriminant, one just needs to set up a threshold on the projected data $\mathbf{w}^T\mathbf{x}$. And the prior class probabilities can be estimated by $P(y=\pm1) = N_\pm/N$ with N_+ and N_- the number of training data in the class + and class −, respectively. More generally, one can fit the projected data using two univariate Gaussian densities, in which case one can calculate the bias term according to (2.14), or obtain the posterior probabilities via Bayes rule.

Note that, a linear discriminant function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ can also be obtained from the least squares solution (or the maximum likelihood estimate under a certain condition) to a linear regression problem. It has been shown that the

resulting discriminant is actually equivalent [1] to the solution for Fisher's linear discriminant provided that the target values for y are appropriately coded as $y = N/N_+$ for class $+$, and $y = -N/N_-$ for class $-$ [24][63].

2.2.2 Logistic regression

Logistic regression (LR) is a traditional statistical tool for classification and has gained its popularity in medical data analysis. LR models can output posterior class probabilities and the model parameters are easy to interpret. In case of a binary classification problem, the response variable y has only two possible outcomes: positive or presence of diseases, denoted by 1 and negative or absence of diseases denoted by 0. The observation can be expressed in a input data vector $\mathbf{x} = (x_1, x_2, ..., x_M)$, with M the number of input variables. Logistic regression tries to model the conditional class probabilities given the input $\mathbf{x}$. Let $P(y = 1|\mathbf{x}) = \pi(\mathbf{x})$ and $P(y = 0|\mathbf{x}) = 1 - P(y = 1|\mathbf{x}) = 1 - \pi(\mathbf{x})$. The model can be expressed as:

$$P(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{w}^T\mathbf{x} + b))} \tag{2.22}$$

where $\mathbf{w} = (w_1, w_2, ..., w_M)^T$ and b are the weight vector and intercept of the decision hyperplane respectively.

This model defines a curvilinear relation between the mean of the response variable and the explanatory variables, which is equivalent to

$$\text{odds}(y = 1) = \frac{P(y = 1|\mathbf{x})}{P(y = 0|\mathbf{x})} = \frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})} = \exp(-(\mathbf{w}^T\mathbf{x} + b)). \tag{2.23}$$

The model is basically a linear one, as the relationship between the input variables and the predicted value, subject to the logit transformation (i.e. taking the natural logarithm of the odds), is linear. The model can also be written as a linear relation between the logit of the mean for the response variable and the input variables, which is equivalent to:

$$\text{logit}(\pi(\mathbf{x})) = \log(\text{odds}) = \log(\frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + \mathrm{b},$$

The usual linear regression is in theory less suitable for predicting a binary outcome than the logistic model, even though for P between 0.2 and 0.8 the two types of models do not differ much. One reason is validity, since linear regression can predict dependent-variable values above 1 or below 0. Whereas the major advantage of the logistic model lies in its weaker assumptions in contrast to

[1] except that the weights are in a different scale

linear regression models, that assume residuals being normally distributed and with constant variance.

The relative influence (importance) of individual input variables is usually expressed in terms of the odds ratio (OR). The general formula is

$$\mathrm{OR} = \exp(-w_i(x_i - x_i')). \tag{2.24}$$

The interpretation of the logistic regression model can thus be done by means of odds ratio. In case of binary input the OR reduces to e^{w_i}. For nominal or continuous variables, the formula for OR represents the change in the odds if x_i increases by one unit while holding all other input values constant.

Prior class probabilities can be incorporated by adjusting the bias term b. An iteratively re-weighted least squares (IRLS) algorithm such as Newton-Raphson method can be employed to perform maximum likelihood estimation for LR model fitting [71]. However, such algorithms potentially suffer from the existence of many local minima.

2.3 Artificial neural networks

Artificial neural networks (ANNs) are networks of units called neurons that exchange information in the form of numerical values with each other via synaptic interconnections. In this thesis, we consider only feed-forward neural networks, which are known as universal approximators, to create a nonlinear mapping between a set of input variables and the output variables. There are two major types of feed-forward neural networks, they are multilayer perceptrons (MLPs) and radial basis function neural networks (RBF-NN). Many successful applications in pattern recognition using these two types of networks have been reported [10].

2.3.1 Multilayer perceptrons

A multilayer perceptron (MLP) creates a nonlinear input-output mapping defined by the layers of summation and elementary nonlinear mappings [10]. We concentrate here to one hidden layer MLPs with hyperpolic tangent activation function ($\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$) for the hidden neurons, and with logistic sigmoid function as the output activation function. Other typical activation functions such as a threshold function can be chosen as well. The MLP model of our concern can be expressed as:

$$f(\mathbf{x}, \mathbf{w}) = g\left(\sum_{j=0}^{M} w_j^{(2)} \tanh\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right)\right), \tag{2.25}$$

where $\mathbf{w}$ includes the weights and the bias parameter, and indices i and j correspond to input and hidden units, respectively. The logistic function is chosen as the output activation function here, since its output value ranges from 0 to 1, and can be interpreted as probability [10].

This multilayer perceptrons (MLP) can be seen as a generalization of logistic regression analysis. Or, the logistic regression model can be seen as a special MLP that has no hidden layer, and takes a logistic sigmoidal function as its output activation function.

The training of the feed-forward neural network is often done by the back-propagation procedure or using more advanced numerical optimization (such as Levenberg-Marquardt, quasi-Newton, conjugate gradient etc.), until the discrepancy between the target output and actual response is minimized [10][42]. Traditionally the complexity of the model can be controlled by weight decay, where the regularized cost function should be minimized.

The well known local minimum problem for MLP learning can be tackled by retraining using different reinitializations of the weights. In order to avoid overfitting, one also needs to select an appropriate network architecture, e.g. the number of hidden neurons, types of activation functions, and the regularization hyperparameters.

2.3.2 Radial basis function neural networks

Radial basis function (RBF) neural networks are also feedforward, but have only one hidden layer. For the architecture shown in Figure 2.1, we can write the RBF-NN mapping from input vector $\mathbf{x}$ to output y in the following form:

$$y(\mathbf{x}) = \sum_{j=0}^{M} w_j \phi_j(\mathbf{x}) \tag{2.26}$$

where $\phi_j(\mathbf{x}) = \exp(-\frac{\|\mathbf{x}-\boldsymbol{\mu}_j\|^2}{2r^2})$ for the case of Gaussian basis functions, and $\mathbf{x}$ is the D-dimensional input vector with elements x_i, and $\boldsymbol{\mu}_j$ is the vector determining the center of basis function $\boldsymbol{\phi}_j$ and has elements μ_{ji}. ϕ_0 is an extra basis function whose activation is set to 1. M denotes the number of hidden neurons. Its output activation function hence is a linear transform of the activations of the hidden neurons (or M basis functions).

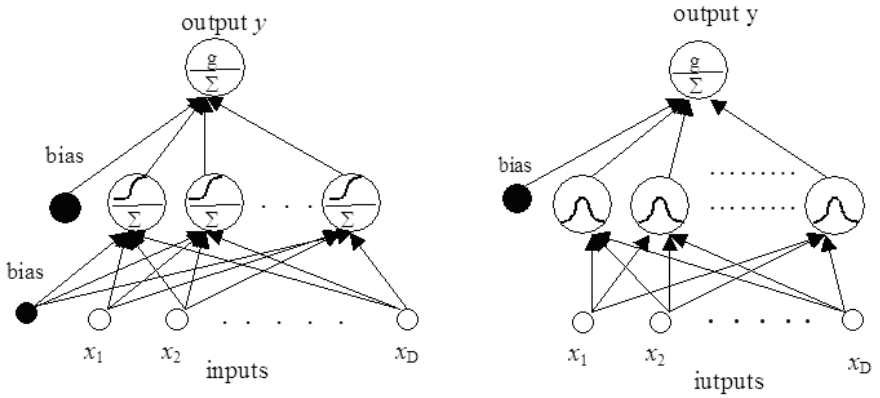


Figure 2.1: A schematic illustration of the architecture of a multilayer perceptron (left) and a radial basis function neural network (right).

The roles of the first and second layers of weights are different in radial basis function networks. This often leads to a two-stage training procedure for training RBF-NNs. In the first stage the input data set $\{\mathbf{x}_n\}_{n=1,\ldots,N}$ alone is used to determine the parameters of the basis function (e.g. $\boldsymbol{\mu}_j$ and r_j for the Gaussian basis function). In the second phase, the basis functions are kept fixed, while the second layer weights are found by supervised learning.

One special paradigm of radial basis function (RBF) neural networks is the generalized regression neural networks (GRNNs), often used for function approximations. [2] It's another term for Nadaraya-Watson kernel regression, and has the following form for the function mapping:

$$y(\mathbf{x}) = \frac{\sum_n y_n \exp(-\|\mathbf{x} - \mathbf{x_n}\|^2/2r^2)}{\sum_n \exp(-\|\mathbf{x} - \mathbf{x_n}\|^2/2r^2)}. \tag{2.27}$$

GRNNs share a special property, namely that they do not require iterative training; the hidden-to-output weights are just the target values y_n of the training set, so the output $y(\mathbf{x})$, is simply a weighted average of the target values y_n of training cases $\mathbf{x}_n$ close to the given input case $\mathbf{x}$. It can be viewed as a normalized RBF neural network in which there is a hidden unit centered at every training case. These RBF units are called "kernels" and are usually probability density functions such as the Gaussian basis function. The only weights that need to be learned are the widths of the RBF units r. These widths (often a single width is used) are called "smoothing parameters" or "bandwidths" and are usually chosen by cross-validation. GRNN is a universal approximator for smooth functions, so it should be able to solve any smooth

[2] When performing classification, the class label can be assigned by thresholding over the latent output of the GRNN model.

function-approximation problem given enough data. The main drawback of GRNNs is that, like kernel methods in general, GRNNs cannot ignore irrelevant inputs without major modifications to the basic algorithm.

2.4 Support vector machines for classification

In the early 1990s, Vapnik and his coworkers invented a computationally powerful class of supervised learning networks, called support vector machines (SVMs) for solving pattern recognition [11]. The new algorithm design is firmly grounded in the framework of statistical learning theory developed by Vapnik, Chervonenkis and others [113], in which VC dimension (Vapnik and Chervonenkis, 1971) provides a measure for the capacity of a neural network to learn from a set of samples. The basic idea of Vapnik's theory is closely related to regularization, nevertheless capacity control is employed for model selection.

2.4.1 Statistical learning theory

Generally, for a given learning task, with a given finite amount of training data, the best generalization performance will be achieved if the right balance is reached between the accuracy attained on the particular training set, and the ability of the machine to learn any training set without error, that is, its *capacity*.

Empirical risk minimization

Let us take again the binary classification problem as example, the task for the supervised learning is to estimate a function $f : \mathbb{R}^D \to \{-1, +1\}$, using input output training data generated i.i.d. according to an unknown probability distribution $P(\mathbf{x}, y), (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \in \mathbb{R}^D \times \mathcal{Y}, \mathcal{Y} \in \{-1, +1\}$, such that f will correctly classify unseen examples $(\mathbf{x}, y)$. An example is then: assign to the class $+1$ if $f(\mathbf{x}) \geq 0$ and to the class -1 otherwise, assuming the test examples are generated from the same distribution as the training data. The best estimated function f can be obtained by minimizing the expected error (risk)

$$R[f] = \int l(f(\mathbf{x}, y) dP(\mathbf{x}, y) \tag{2.28}$$

where l denotes a chosen loss function, e.g. a $0/1$ loss function defined as $l(f(\mathbf{x}, y) = \Theta(-yf(\mathbf{x}))$, with $\Theta(z) = 0$ for $z < 0$ and $\Theta(z) = 1$ otherwise.

However, the underlying distribution $P(\mathbf{x}, y)$ is usually unknown. This necessitates to estimate the function f from the candidate function class $\mathbf{F}$ by

minimizing the approximation of the risk (2.28), i.e. the *empirical risk*:

$$R_{\text{emp}}[f] = \frac{1}{N}\sum_{i=1}^{N} l(f(\mathbf{x}_i), y_i) \tag{2.29}$$

Under certain conditions one can ensure that the empirical risk will converge toward the expected risk asymptotically (as $N \to \infty$). However, for small sample sizes, large deviations are possible and overfitting might occur. One way to avoid overfitting is to restrict the complexity of the candidate function class $\mathbf{F}$, e.g. by using regularization. This intuition can be described in the form of Occam's razor, as choosing the "simple" function that explains most of the data is preferable to a complex one. This is also the problem of model selection.

Structural risk minimization

A specific way of controlling the complexity of a function is given by Vapnik-Chevonenkis (VC) theory and the structural risk minimization (SRM) principle. The complexity can be captured by the VC dimension h of the function class $\mathbf{F}$ that the estimate f is chosen from. The VC dimension is defined as the largest number of (training) points that can be separated for all possible labeling using functions of the given class.

VC theory shows the set of functions $\mathbf{F}$ where f is chosen from must be restricted to one which has a capacity that is suitable for the amount of the available training data. To do so, VC theory introduces bounds on the expected risk that depends on both the empirical risk and the capacity of $\mathbf{F}$. The minimization of these bounds leads to the *structural risk minimization* principle. Based on this principle, for all the functions in $\mathbf{F}$ and $N > h$ with a probability of at least $1 - \eta$ ($0 \leq \eta \leq 1$), the following bound holds:

$$R[f] \leq R_{\text{emp}}[f] + \sqrt{\frac{h(\log(2N/h)) + 1 - \log(\eta/4)}{N}}. \tag{2.30}$$

The second term of this bound is called the VC *confidence* term, which is a monotonically increasing function of h for every value of N.

In order to choose a function class $\mathbf{F}_i$, one can construct a structure, i.e. a nested family of function classes $\mathbf{F}_1 \subset \ldots \subset \mathbf{F}_k$ with nondecreasing VC dimensions. Let $f_1, \ldots, f_k$ be the solutions of the empirical risk minimization (2.29) in the function class $\mathbf{F}_i$. SRM chooses the function class $\mathbf{F}_i$ and the function f_i, such that an upper bound e.g. (2.30) on the generalization error is minimized.

Note that there are some other tighter upper bounds as well. However, they all suffer from similar problems in practice, such as the difficulties in computing

the VC dimension. Nevertheless, bounds offer helpful theoretical insights into the nature of learning problems.

Margins and VC dimension

Now let us assume that the training sample is separable by a hyperplane (see Fig. 2.2), i.e., we choose functions of the form

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b. \tag{2.31}$$

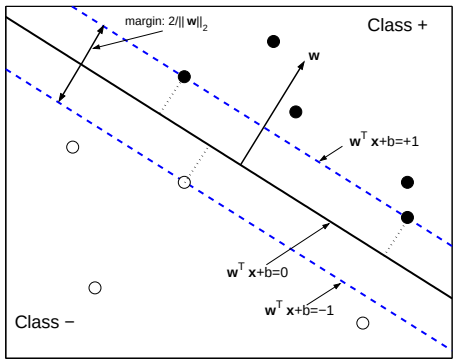


Figure 2.2: Illustration of a linear classifier and its margin. A linear classifier is defined by a hyperplane's normal vector $\mathbf{w}$ and an bias b, i.e. the decision boundary is $\{\mathbf{x}|\mathbf{w}^T\mathbf{x} + b = 0\}$ (solid line). The margin of the linear classifier in this case is the distance between the dashed lines.

It has been shown that for the class of hyperplanes the VC dimension itself can be bounded in terms of another quantity, the margin. The margin is defined as the minimal distance of a sample to the decision surface. The margin in turn can be measured by the length of the weight vector in (2.31): as we assumed that the training sample is separable we can rescale and such that the points closest to the hyperplane satisfy $|\mathbf{w}^T\mathbf{x}| = 1$. Now consider two samples $\mathbf{x}_1$ and $\mathbf{x}_2$ from different classes with $\mathbf{w}^T\mathbf{x}_1 + b = 1$ and $\mathbf{w}^T\mathbf{x}_2 + b = -1$, respectively. Then the margin is given by the distance of these two points, measured perpendicular to the hyperplane, i.e., $2/\|\mathbf{w}\|_2$. Vapnik has shown that hyperplanes satisfying $\|\mathbf{w}\|_2 < \Lambda$ have a VC-dimension h which is upper bounded by

$$h \leq \min([\Lambda^2 R^2], N) + 1, \tag{2.32}$$

where $[\cdot]$ denotes the integer part and R is the radius of the smallest ball around the data (the points $\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_N)$ in the high dimensional feature space, see Section 2.4.2) [113]. This inequality links the VC dimension of the class of separating hyperplanes to the margin or the length of the weight vector. This insight has been implemented in support vector machines, which maximize the

margin of a classifier. Note that, although the choice of linear functions at this moment seems very limiting, a very rich set of nonlinear decision functions is possible via linear models by use of the kernel trick.

2.4.2 Kernel induced feature spaces

The basic idea of algorithms in kernel induced feature spaces is the following. The data $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$ is to be mapped to a potentially higher (even infinite) dimensional feature space $\mathcal{F}$ via a mapping:

$$\begin{aligned} \varphi(\cdot) : \mathbb{R}^D &\rightarrow \mathbb{R}^{D_\mathcal{F}} \\ \mathbf{x} &\mapsto \varphi(\mathbf{x}). \end{aligned}$$

For a given learning problem, one can use the same algorithms in $\mathcal{F}$ instead of $\mathbb{R}^D$, i.e. working on the transformed examples

$$(\varphi(\mathbf{x}_1), y_1), \ldots, (\varphi(\mathbf{x}_N), y_N) \in \mathcal{F} \times \mathcal{Y}.$$

Given this mapped representation, a simple linear classifier in $\mathcal{F}$ can be constructed. This is also implicitly done for MLPs and RBF-NNs. And the variability, richness or nonlinearity of the decision hyperplanes is then introduced by the mapping $\varphi(\cdot)$.

Although it is difficult to know the mapping $\varphi(\cdot)$ directly, for certain feature spaces and corresponding mappings there is a trick for computing scalar products in feature spaces using kernel functions. The computation of a scalar product between two feature space vectors $\mathbf{x}$ and $\mathbf{z}$, can be readily reformulated in terms of a kernel function k:

$$\langle \varphi(\mathbf{x}), \varphi(\mathbf{z}) \rangle = \varphi(\mathbf{x})^T \varphi(\mathbf{z}) := k(\mathbf{x}, \mathbf{z}). \tag{2.33}$$

If $k : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$ is a continuous kernel of a positive integral operator on a Hilbert space $L_2(\mathcal{C})$ on a compact set $\mathcal{C} \subset \mathbb{R}^D$, i.e.,

$$\forall f \in L_2(\mathcal{C}) : \int_{\mathcal{C}^\epsilon} k(\mathbf{x}, \mathbf{z}) f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z} \geq 0 \tag{2.34}$$

then there exists a space $\mathcal{F}$ and a mapping $\varphi(\cdot) : \mathbb{R}^D \rightarrow \mathcal{F}$ such that $k(\mathbf{x}, \mathbf{z}) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{z}) \rangle$ [113]. This can be seen directly from Mercer's theorem [64] saying that any kernel of a positive integral operator can be expanded in its eigenfunctions $\psi_j (\lambda_j > 0, D_\mathcal{F} \leq \infty)$:

$$k(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{D_\mathcal{F}} \lambda_j \psi_j(\mathbf{x}) \psi_j(\mathbf{z}). \tag{2.35}$$

In this case, one can find a possible realization of the feature space in the form of

$$\varphi(\mathbf{x}) = (\sqrt{\lambda_1}\psi_1(\mathbf{x}), \sqrt{\lambda_2}\psi_2(\mathbf{x}), \ldots). \tag{2.36}$$

Note furthermore that using a particular support vector kernel corresponds to an implicit choice of the feature space. Table 2.4.2 lists some of the most widely used kernel functions.

Table 2.1: Common kernel functions: linear kernel, polynomial (degree $d \in \mathbf{N}, \theta \in \mathbb{R}$), RBF (bandwidth $r \in \mathbb{R}_+$, multilayer perceptron ($\kappa, \theta \in \mathbb{R}$). These kernels are known to fulfill Mercer's condition except for MLP kernels. Not all the choice of κ and θ will lead to a MLP kernel satisfying Mercer's condition.

Linear	$k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T\mathbf{z}$
Polynomial	$(\mathbf{x}^T\mathbf{z} + \theta)^d$
RBF	$k(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{r^2}\right)$
MLP	$\tanh(\kappa\mathbf{x}^T\mathbf{z} + \theta)$

2.4.3 Support vector classifiers

Separable classification

In its simplest, linear form, a SVM is a hyperplane $y = \text{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + \mathrm{b})$ that separates the positive from the negative examples with maximum margin. For nonlinear support vector classification, the kernel trick described in the last section is used. A hyperplane in the feature space $\mathcal{F}$ usually $y = \text{sign}(\mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}) + \mathrm{b})$ is to be sought.

According to the structural risk minimization principle, the bound (2.30) for expected risk is to be minimized. In case of linearly separable problems, the empirical risk can be kept zero, while minimizing the complexity term, which is a monotonically increasing function of the VC dimension h. Thanks to the upper bound $h \leq \|\mathbf{w}\|^2R^2 + 1$, we can minimize the complexity term by minimizing $\|\mathbf{w}\|^2$, or equivalently, maximizing the margin. Finding this optimal hyperplane (without training error) thus involves solving the following optimization problem:

$$\min_{\mathbf{w},b} \tfrac{1}{2}\|\mathbf{w}\|^2, \tag{2.37}$$

$$\text{subject to } y_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b) \geq 1, \ , i = 1, \ldots, N. \tag{2.38}$$

We can avoid the explicit usage of $\mathbf{w}$ by forming the dual optimization problem. Introducing Lagrange multipliers $\alpha_i \geq 0, i = 1, \ldots, N$, one for each of the

constraints in (2.38), we get:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^T \varphi(\mathbf{x}_i) + b) - 1). \tag{2.39}$$

The task is to minimize (2.39) with respect to $\mathbf{w}, b$ and to maximize it with respect to α_i. At the optimal point, we have the following saddle point equations:

$$\frac{\partial L}{\partial b} = 0 \text{ and } \frac{\partial L}{\partial \mathbf{w}} = 0$$

i.e.

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \varphi(\mathbf{x}_i). \tag{2.40}$$

Note that according to the Karush-Kuhn-Tucker (KKT) conditions, at the saddle point for each Lagrange multiplier α_i, the product of the multiplier with its corresponding constraint vanishes, as shown by

$$\alpha_i (y_i(\mathbf{w}^T \mathbf{x} + b) - 1) = 0 \text{ for } i = 1, \ldots, N. \tag{2.41}$$

Therefore, only those multipliers exactly satisfying equation (2.41) can assume nonzero values.

By substituting (2.40) into (2.39) and replacing $\langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle$ with kernel functions $k(\mathbf{x}_i, \mathbf{x}_j)$, we reach the dual quadratic optimization problem:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & \alpha_i \geq 0, i = 1, \ldots, N. \\ & \sum_{i=1}^{N} \alpha_i y_i = 0. \end{aligned}$$

Thus, by solving the dual optimization problem, one obtains the coefficients $\alpha_i, i = 1, \ldots, N$. This leads to the nonlinear decision function

$$\begin{aligned} f(\mathbf{x}) &= \text{sign}(\sum_{i=1}^{N} y_i \alpha_i \varphi(\mathbf{x})^T \varphi(\mathbf{x}_i) + b) \\ &= \text{sign}(\sum_{i=1}^{N} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b). \end{aligned}$$

Non-separable classification

For noisy data, we might face overfitting effects if the empirical error is zero. To find a good tradeoff between the empirical risk and the complexity term in (2.30), one can introduce slack variables to relax the hard margin constraints

$$y_i(\mathbf{w}^T\varphi(\mathbf{x})+b) \geq 1-\xi_i, \quad \xi_i \geq 0, \quad i=1,\ldots,N, \tag{2.42}$$

additionally allowing for some errors in the training set. The SVM solution can be found by 1) keeping the upper bound on the VC dimension small and 2) by minimizing an upper bound $\Sigma_{i=1}^{N}\xi_i$ on the empirical risk, i.e. the number of training errors. Thus one minimizes

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \tag{2.43}$$

where the regularization constant $C>0$ determines the tradeoff between the empirical error and the complexity term. This leads to the dual problem:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j) \tag{2.44}$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C, \quad i=1,\ldots,N. \tag{2.45}$$

$$\sum_{i=1}^{N}\alpha_i y_i = 0. \tag{2.46}$$

By introducing the slack-variables ξ_i, one gets the *box* constraints that limit the size of the Lagrange multipliers: $\alpha_i \leq C, i=1,\ldots,N$.

Also from the KKT conditions, at the saddle point, we have the following:

$$\begin{aligned} \alpha_i = 0, &\quad \Rightarrow \quad y_i f(\mathbf{x}_i) \geq 1 \text{ and } \xi_i = 0 \\ 0 < \alpha_i = C, &\quad \Rightarrow \quad y_i f(\mathbf{x}_i) = 1 \text{ and } \xi_i = 0 \\ \alpha_i = C, &\quad \Rightarrow \quad y_i f(\mathbf{x}_i) \leq 1 \text{ and } \xi_i \geq 0. \end{aligned} \tag{2.47}$$

They reveal one of the most important properties of SVMs: the solution is sparse in $\boldsymbol{\alpha}$, i.e. many patterns are outside the margin area and the optimal α's are zero. Specifically, based on the KKT conditions, only those training pattern $\mathbf{x}_i$, which is either on the margin or inside the margin, could have a nonzero corresponding α_i. Such training data points $\mathbf{x}_i$ with nonzero α_i are thus called support vectors (SVs).

The optimum bias b can be determined by exploiting the fact that for all support vectors (SVs) $\mathbf{x}_i$ with $0 < \alpha_i < C$, the slack variable ξ_i is zero. Thus for any support vector $\mathbf{x}_i$ with $i \in I := \{i : 0 < \alpha_i < C\}$ holds $y_i(b + \Sigma_{j=1}^{N} y_j \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)) = 1$. Thus averaging over these patterns we obtain a numerically stable solution for the bias term b:

$$b = \frac{1}{|I|} \sum_{i \in I} (y_i - \sum_{j=1}^{N} y_j \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)). \tag{2.48}$$

Some properties of the SVM classifier and its solutions are the following:

- *Choice of kernel functions*: the kernel functions should satisfy the Mercer condition.
- *Global and unique solution*: The solution to the convex QP problem is global and unique provided a positive definite kernel function is chosen. For a positive semidefinite kernel, the solution to the QP problem is global but not necessarily unique.
- *Sparseness*: In the dual space the SVM classifier takes the form: $y(\mathbf{x}) = \text{sign}(\sum_{i \in I} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b)$, where only the support vectors $\mathbf{x}_i$ with non-zero α_i contribute to the output of SVM classifier.
- *Geometric meaning of support vectors*: The support vectors are those data points that lie closest to the decision and are therefore the most difficult to classify.

2.4.4 Least squares support vector classifiers

Standard SVM classifiers solve classification problems by means of convex quadratic programming (QP), and achieve sparseness in the solution as a result of the QP problem. Suykens [94][95] proposed the idea of modifying Vapnik's SVM formulation by adding a least squared term to the cost function and replacing inequality by equality constraints, thereby transforming the problem to solving a set of linear equations in the dual space instead of the quadratic programming as for the standard SVM case. This approach significantly reduces the costs in complexity and computation time for solving the problems for moderate data set sizes. Moreover, LS-SVM is closely related to Gaussian processes and kernel Fisher discriminant analysis [95][111].

The LS-SVM classifier $y(\mathbf{x}) = \text{sign}\left[\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) + b\right]$ in the primal space is inferred from the mapped data $[(\varphi(\mathbf{x}_1), y_1), \ldots, (\varphi(\mathbf{x}_N), y_N)]$ in the feature space $\mathcal{F}$ with binary targets $y_i \in \{-1, +1\}$ via the following optimization problem.

$$\min_{\mathbf{w},b,\mathbf{e}} \mathcal{J}(\mathbf{e},\mathbf{w},b) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} e_i^2, \tag{2.49}$$

$$\text{subject to } y_i[\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_i) + b] = 1 - e_i, \quad i = 1,\ldots,N \tag{2.50}$$

In the LS-SVM formulation, instead of inequality constraints, equality constraints have been used in (2.50), where the value 1 at the right hand side is considered as a target value instead of a threshold value. In this sense, the LS-SVM classifier is intrinsically following a regression approach taking a target value of $\{-1,+1\}$. There are variable e_i functions similarly as the slack variable ξ_i in the SVM formulation. Moreover, a squared loss function is taken for the empirical risk function. This modification has simplified the computation problems, as explained next.

The direct computation in the primal space is in general difficult, and $\mathbf{w}$ in the feature space might be very high and potentially infinite. The optimization problem in the dual space is to be derived. The Lagrangian for the problem is

$$\mathcal{L}(\mathbf{w},b,e;\alpha) = \mathcal{J}(\mathbf{w},e) - \sum_{i=1}^{N}\alpha_i(y_i(\mathbf{w}^T\varphi(\mathbf{x}) + b) - 1 + e_i) \tag{2.51}$$

where the α_i values are the Lagrangian multipliers, which can be positive or negative now due to the equality constraints. The conditions for optimality yield

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 0 \quad &\rightarrow \quad \mathbf{w} = \Sigma_{i=1}^{N}\alpha_i y_i\varphi(\mathbf{x}_i),\\
\frac{\partial\mathcal{L}}{\partial b} = 0 \quad &\rightarrow \quad \Sigma_{i=1}^{N}\alpha_i y_i = 0,\\
\frac{\partial\mathcal{L}}{\partial e_i} = 0 \quad &\rightarrow \quad \alpha_i = Ce_i\\
\frac{\partial\mathcal{L}}{\partial\alpha_i} = 0 \quad &\rightarrow \quad y_i(\mathbf{w}^T\varphi(\mathbf{x}_i + b) - 1 + e_i = 0
\end{aligned} \tag{2.52}$$

for $i = 1,\ldots,N$.

This optimization problem can be transformed and solved through a linear system in the dual space [94][95]:

$$\left[\begin{array}{c|c} 0 & \mathbf{y}^T \\ \hline \mathbf{y} & \boldsymbol{\Omega} + C^{-1}\mathbf{I} \end{array}\right]\left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{1} \end{array}\right] \tag{2.53}$$

with $\mathbf{y} = [y_1,\cdots,y_N]^T$, $\mathbf{e} = [e_1,\cdots,e_N]^T$, $\boldsymbol{\alpha} = [\alpha_1,\cdots,\alpha_N]^T$, and $\mathbf{1} = [1,\cdots,1]^T$.

Mercer's theorem is applied to the matrix $\mathbf{\Omega}$ with $\mathbf{\Omega}_{ij} = y_i y_j \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x}_j) = y_i y_j \, k(\mathbf{x}_i, \mathbf{x}_j)$, where $k(\cdot, \cdot)$ is a chosen positive definite kernel that satisfies Mercer condition [64].

The LS-SVM classifier is then constructed in the dual space as:

$$y(x) = \text{sign}\left[\sum_{i=1}^{N} \alpha_i \, y_i \, k(\mathbf{x}, \mathbf{x}_i) + b\right].$$

In order to achieve a high performance of LS-SVM models, one still need to tune the regularization and possible kernel parameters.

In brief, the above conventional modelling techniques often apply error minimization based maximum likelihood or MAP approaches for supervised learning. In order to avoid overfitting, one needs also to control the model complexity, thus to choose the appropriate model structure and hyperparameters such as regularization parameters and possibly the kernel parameters. This is usually done via cross validation (CV). However, we should be aware that CV gives noisy estimates (with large variance) for the model performance, and it easily becomes computationally prohibitive with the increase of the number of (hyper)parameters to be determined.

2.4.5 Generalized additive kernel-based models

In general, together with artificial neural networks, support vector classifiers and least squares support vector classifiers, are viewed as blackbox models due to the difficulty in interpretation of the behavior and decision of these models. The interpretation problem also highlights an important feature of the linear model that has made it so popular for statistical inference: the linear model is additive in the predictor effects. Once we have fitted the linear model we can examine the predictor effects separately, in the absence of interactions.

Additive models retain some of the interpretability of the linear model by assuming additivity of effects [39]. An additive model is defined by

$$y = \sum_{j=1}^{D} f_j(x_j) + b + \epsilon \tag{2.54}$$

where as before the errors ϵ are independent of the x_j, $E(\epsilon) = 0$ and $\text{var}(\epsilon) = \sigma^2$. The f_j are arbitrary univariate functions, one for each input variable.

Generalized additive models make the same extension from additive models as for generalized linear regression models, i.e. to apply a link function such as the logistic link function to the additive models. The generalized additive model

in the form of logistic regression replace the linear predictor with an additive one:

$$\log(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}) = \sum_{j=1}^{D} f_j(x_j) + b, \tag{2.55}$$

where $\pi(\mathbf{x}) = P(y = 1|\mathbf{x})$.

There are algorithms designed specifically for the generalized additive models such as backfitting [39]. However, we would not go into that direction in this thesis. Our goal of applying generalized additive models is to improve the interpretation in kernel-based classifiers. Additive kernel-based models can be obtained by using additive kernels, which assume an additive effect of each variable to the evaluation of the kernel functions. We define an additive kernel function as the sum of univariate kernel functions for all the variables.

$$k(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{D} k_j(x_j, z_j) \tag{2.56}$$

where $k_j(x_j, z_j)$ is a univariate kernel function, and usually only fixed kernel functions are used, i.e. $k_j(x_j, z_j) = k(x_j, z_j), j = 1, \ldots, D$. All the kernel functions as defined previously (cf. Table 2.4.2) have their own additive version by simply summing up the univariate kernel functions for all the variables, though this leads to a different model. For example, the additive RBF kernel can be written as:

$$k(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{D} \exp\left[-\frac{(x_j - z_j)^2}{r^2}\right]. \tag{2.57}$$

And a linear kernel is itself an additive kernel, because $\mathbf{x}^T\mathbf{z} = \sum_{j=1}^{D} x_j z_j$.

Therefore, just by using additive kernels, one can obtain an additive kernel-based model without modifying the learning algorithms. The additive kernel-based model can be expressed by

$$f(\mathbf{x}) = \sum_{j=1}^{D}\sum_{i=1}^{N} \alpha_i y_i k_j(x_j, x_{ij}) + b \tag{2.58}$$

The advantages of using additive kernel-based models are the fact that the effect of the individual variables can be examined and thus the model behavior can somehow be better understood and interpreted. Some examples of using additive kernel-based classifiers will be given in Chapter 3 and 4.

2.4.6 Kernel principal component regression and LS-SVMs

It is sometimes helpful to check the decisions of the support vector classifiers directly in the feature space $\mathcal{F}$, so as to get some insights to the decisions of the kernel based models. As mentioned previously, a possible realization of the kernel features transformed from the input $\mathbf{x}$ could be in the form of equation 2.36, i.e. $\varphi(\mathbf{x}) = (\sqrt{\lambda_1}\psi_1(\mathbf{x}), \sqrt{\lambda_2}\psi_2(\mathbf{x}), \ldots)$, where λ_k and $\psi_k(\mathbf{x})$ correspond to the kth eigenvalue and eigenfunction of the kernel satisfying Mercer's condition, respectively. Extraction of the kernel features could also be derived indirectly via the kernel principal component analysis (kPCA), i.e. performing principal component analysis on the feature space. A link between SVMs and kPCA has been established in [97], showing that the PCA problem can be interpreted as a one-class modeling problem with a target value equal to zero around which one maximize the variance, and thus can be solved by using the LS-SVM formulations.

Principal component regression performs regression in projection of the data points onto the principal components. Here, we attempt to link kernel principal component regression with the decision boundaries of LS-SVMs.

The idea is to first transform the $\varphi(\mathbf{x})$ in the kernel induced feature space into a score vector $\mathbf{z} = \mathbf{V}^T(\varphi(\mathbf{x}) - \hat{\mathbf{m}}_\varphi)$, where $\hat{\mathbf{m}}_\varphi = \frac{1}{N}\sum_{i=1}^{N}\varphi(\mathbf{x}_i)$, $\mathbf{V}$ are the projection vectors $[\mathbf{v}_1, \cdots, \mathbf{v}_N]$, $\mathbf{v}_k$ corresponding to the kth principal component (PC) found by applying PCA to the covariance matrix of the centered vectors $[\varphi(\mathbf{x}_1) - \hat{\mathbf{m}}_\varphi, ..., \varphi(\mathbf{x}_N) - \hat{\mathbf{m}}_\varphi]$. Since the feature space is typically unknown, z scores will be computed in the dual space using the kernel trick [95]. Defining an $N \times N$ matrix $K_{ij} := \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$, one computes an eigenvalue problem for the expansion coefficient α_i, that is now only dependent on the kernel function

$$\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}, \tag{2.59}$$

with $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_N)^T$. The solutions in feature space $\mathcal{F}$ should be normalized, i.e. $\mathbf{V}\mathbf{V}^T = \mathbf{I}_N$. Also as in standard PCA algorithm, the data in $\mathcal{F}$ should be centered. This can be done via substituting the kernel matrix $\mathbf{K}$ with the centered one: $\mathbf{K}_{\mathrm{c}} = \mathbf{K} - \mathbf{1}_{\mathrm{c}}\mathbf{K} - \mathbf{K}\mathbf{1}_{\mathrm{c}} + \mathbf{1}_{\mathrm{c}}\mathbf{K}\mathbf{1}_{\mathrm{c}}$, where $(\mathbf{1}_{\mathrm{c}})_{ij} = 1/N$. For details see [85]. For extracting features of a new pattern $\mathbf{x}$ with kPCA, one simply projects the mapped pattern $\varphi(\mathbf{x})$ on to $\mathbf{v}_k$ to get the z score on the kth principal component:

$$\begin{aligned} z_k(\mathbf{x}) &= \langle \mathbf{v}_k, \varphi(\mathbf{x})\rangle \\ &= \sum_{i=1}^{M}\alpha_i^{(k)}\langle\varphi(\mathbf{x}_i), \varphi(\mathbf{x})\rangle \\ &= \sum_{i=1}^{M}\alpha_i^{(k)}k(\mathbf{x}_i, \mathbf{x}), \end{aligned} \tag{2.60}$$

with $\boldsymbol{\alpha}^{(k)}$ the eigenvector for the kth principal component computed via (2.59).

Kernel principal component regression (kPCR) then performs regression on the new projected data $\{\mathbf{z}(\mathbf{x}_i), y_i)\}_{i=1}^N$, fitting the model: $\tilde{y}(\mathbf{x}) = \sum_{k=1}^{N_{\text{PC}}} \tilde{w}_k z_k(\mathbf{x}) + \tilde{b}$, where N_{PC} is the chosen number of first principal components (associated with the largest eigenvalues) to be used in the model. If we take all nonzero z_k scores (with nonzero eigenvalues) into account, no information of the original data set will be lost, and we can estimate the parameter $\tilde{\mathbf{w}}$ by using the regularized least squares solution (or ridge regression solution) in (2.8).

Now we check the relevance of the kPCR model with LS-SVM classifier, which intrinsically performs ridge regression in feature space $\mathcal{F}$ for target $y = \pm 1$

$$\begin{aligned} \tilde{y}(\mathbf{x}) &= \mathbf{w}^T \varphi(\mathbf{x}) + b \\ &= \mathbf{w}^T \mathbf{V}\mathbf{V}^T (\varphi(\mathbf{x}) - \hat{\mathbf{m}}_\varphi) + b + \mathbf{w}^T \hat{\mathbf{m}}_\varphi \\ &= \tilde{\mathbf{w}}^T \mathbf{z} + \tilde{b} \end{aligned} \tag{2.61}$$

with $\mathbf{V}\mathbf{V}^T = \mathbf{I}_N$, the regression coefficient vector $\tilde{\mathbf{w}} = \mathbf{V}^T\mathbf{w}$ and bias term $\tilde{b} = b + \mathbf{w}^T \hat{\mathbf{m}}_\varphi$. If we use the same amount of regularization as for the LS-SVM classifier, i.e. let the regularization parameter in (2.6) $\lambda = 1/C$, then the output $\tilde{y}(\mathbf{x})$ of the two estimation functions will be equivalent. Based on this relation, the data points and the separating hyperplanes in the feature space $\mathbf{w}^T\varphi(\mathbf{x}) + b = 0$ can be visualized by projecting them on different pairs of PCs in the feature space. In supervised classification problems, PCs with large correlation to the target should be selected first for visualization. Some toy examples of using kPCA and kPCR can be seen in [112][95]. An example of visualizing the LS-SVM classifiers by means of kPCR is given in the next chapter.

2.5 Further topics

2.5.1 Variable selection

In this section we will give a very brief review of the existing variable selection methods. A more detailed treatment of the variable selection problem is given in Chapter 4 and 5 for the medical classification problems at hand.

Variable (feature) selection methods have received much attention in the classification literature, where two kinds of methods have generally been studied - filter methods and wrapper methods [48]. A wrapper method makes use of the algorithm that will be used to build the final classifier, while a filter method is independent of the classifier. A wrapper method searches in the space of the

variable subset, and compares the models by using e.g. the performance measure of the trained classifiers via cross-validation, information criterion (such as AIC), or some other fitness measure of the models. A filter method attempts to find a predictive subset of variables by ranking the simple statistics computed from the empirical distribution (such as correlation between the input variable and class labels, and information gain).

In traditional statistics, information criteria based on a chi-square test are often used as measures for model selection. Variable selection can be performed using these criteria and via a heuristic search such as forward, backward and stepwise selection [71]. In forward selection, the variable with the largest score chi-square statistic from the remaining variables is entered into the model if it meets the specified significance level for entry. Backward elimination progressively removes the least significant variable from the model, until all the variables in the model satisfy the critical staying significance level. In stepwise selection variables are entered into and removed from the model in such a way that each forward selection step may be followed by one or more backward elimination steps. The stepwise selection process terminates if no further variable can be added to the model or if the variable just entered into the model is the only variable removed in the subsequent backward elimination. Stepwise logistic regression which selects variables under the guidance of the partial F-statistics, has been employed as a conventional variable selection technique in this thesis.

Forward, backward and stepwise variable selection procedure provides an efficient way to find a subset of variables for model development. These procedures are however unstable, i.e., slight changes in the learning data set can cause changes in the selected subset. Multi-collinearity problem can also cause problems and should be taken seriously into consideration the selection process and model interpretation. In order to stabilize the selection process, we can apply resampling methods such as bagging, as will be introduced later both in this chapter and in Chapter 5.

The third approach to variable selection are the so-called embedded methods, usually achieved via regularization or shrinkage methods such as the LASSO (Least Absolute Shrinkage and Selection Operator) [99][40]. While the aforementioned subset selection methods search in the combinatory space of the variable subset, regularization or shrinkage methods trim the hypothesis space by constraining the magnitude of the parameters. For example, the LASSO fits a linear model $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}+b$, by solving a constrained optimization problem:

$$\min_{\mathbf{w},b} \sum_{i=1}^{N}(y - \mathbf{w}^T\mathbf{x})^2 + b, \tag{2.62}$$

$$\text{subject to} \qquad \sum_{d=1}^{D} |w_d| \leq s, \tag{2.63}$$

where s is a tuning hyperparameter. When taking a small value of $s \geq 0$, the solutions are shrunken versions of the least squares estimates, and some of the weights w_d are zero. The computation of a LASSO solution is a quadratic

programming problem, other more efficient algorithms exist as well. Several generalizations of the lasso-type estimator for performing variable selection with more complex models such as additive models, neural networks or kernel machines are also available [36].

2.5.2 Multiclass classification

So far, our discussion has been focused on binary classification, where the class labels can have only two values. Many real-world problems, however, have more than two classes, for example the tumor types usually are not restricted to benign and malignant tumors, they could be subclassified into more types, or tumors with different grades. The multiclass classification refers to assigning each of the observations into one of $K(K > 2)$ classes.

In designing machine learning algorithms, it is often easier first to devise algorithms for distinguishing between only two classes. Some machine learning algorithms, such as Fisher's linear discriminant and logistic regression can be naturally extended to handle the multiclass case. For other algorithms, such as support vector machines algorithm, a direct extension to the multiclass case may be problematic. Typically, in such cases, the multiclass problem is reduced to multiple binary classification problems that can be solved separately.

Overall, there is probably no multiclass approach that generally outperforms the others. For practical reasons, the choice of approach will depend on constraints at hand. Relevant factors include the required accuracy, the time available for development and training, and the nature of the classification problem (e.g. for a problem with many classes, it would not be wise to use one versus one (1vs1) approach.

Recall that we are interested in classifiers which can provide class probability estimate, hence in this dissertation we utilize the pairwise methods which can generate multiclass probability. Prior to introducing pairwise coupling algorithm for computing class probability for multiclass classification, we first review briefly the commonly used strategies for multiclass classification problems. [86]

Multiclass objective functions

From a theoretical point of view, a most elegant multiclass algorithm would be to modify the objective function in such a way that it simultaneously allows the computation of a multiclass classifier. In terms of accuracy, the results obtained with this approach are comparable to those obtained with the widely used one versus the rest (1vsR) approach. Unfortunately, the optimization problem forces it to deal with all training data at the same time. In the other approaches, the individual binary classifiers usually have much smaller training

sets, with beneficial effects on the training time.

Error-correcting output coding

Error-correcting output coding (ECOC) have been proposed to enhance generalization ability of classifiers [23]. The idea of the error-correcting output coding (ECOC) method has also been applied to classifiers such as SVMs, LS-SVMs logistic regression and decision-tree algorithms [1][95]. One can generate a large number L of further binary problems by splitting the original set of classes into two subsets. If the set of binary classifiers $f^1, \cdots, f^L$ is designed properly, the binary responses will completely determine the class of a test case. Each class corresponds to a unique vector in $\{\pm 1\}^L$. For K classes, we thus get a so-called decoding matrix $M \in \{\pm 1\}^{K \times L}$. There are situations where the binary responses are inconsistent with each other, when for instance, the problem is noisy or the training sample is too small to estimate the binary classifiers reliably. To deal with these cases, it was proposed that one can design a set of binary problems properly, and the closest match between the vector of responses and the rows of the coding matrix M is determined using for example the Hamming distance (the number of entries where the two vectors differs; essentially, the L_∞ distance).

One versus the rest

To get K-class classifiers, it is common to construct a set of binary classifiers $f^1, \cdots, f^K$, each trained to separate one class from the rest, and combine them by doing the multiclass classification according to the maximal output before applying the sign function (i.e. the so-called latent output). The latent output can also be used for rejecting decisions.

The main shortcoming of 1vsR, sometimes called the winner-takes-all approach, is that it is somewhat heuristic. The binary classifiers used are obtained by training on different binary classification problems, and thus it is unclear whether their latent outputs (before thresholding) are on comparable scales. Although there are already some efforts on transforming the latent output to class probability, nevertheless, the probability is still conditional. On the contrary, one needs the joint class probabilities for all the classes in order to make it comparable. This can be a problem, since situations often arise where several binary classifiers assign the pattern to their respective class (or where none does); in this case, one class must be chosen by comparing the real-valued outputs. In addition, binary 1vsR classifiers are also dealing with the rather asymmetric (or more unbalanced) problems. In this aspect, the one versus one approach is advantageous since the problems are more symmetric compared with 1vsR.

One versus one - pairwise classification

For pairwise one versus one (1vs1) multiclass classification, there exist also several ways to combine the binary classifiers. For K classes, this results in $K(K-1)/2$ binary classifiers. This number is usually larger than the number of 1vsR classifiers. Although this suggests large training times, the individual problems that we need to train on are significantly smaller, and if the training algorithm scales superlinearly with the training set size, it is actually possible to save time. One is the voting scheme which will simply assign to the test case the class with the maximum votes among the pairwise 1vs1 classifiers. That is, the test case is assigned to the class that has the highest number of votes in all the pairwise comparisons. Another one is to embed the binary classifiers into a directed acyclic graph. Each classification run then corresponds to a directed transversal of that graph, and classification can be much faster [79].

Pairwise coupling for computing probability outcome of multiclass classification

The aforementioned methods are all strategies that could be applied to any types of classifier. However, in problems of medical classification, the probability output is very important for the final decision making. Hence we now focus on the methods which can provide the joint probability for each class. There are a couple of variants on the pairwise coupling methods, here we only concentrate on the version of Hastie [41].

It has been pointed out by Friedman [30] that the rule of majority voting for combining the pairwise classification is equivalent to Bayes rule when the class posterior probabilities p_i (at the test point) are known:

$$\text{argmax}_i[p_i] = \text{argmax}_i[\sum_{j \neq i} \text{I}(\frac{p_i}{p_i + p_j} > \frac{p_j}{p_i + p_j})], \tag{2.64}$$

where $\text{I}(\cdot)$ is an indicator function. Hastie proposed to improve the majority voting scheme by combining the pairwise class probability estimates into a joint probability estimate for all K classes [41], i.e. by coupling pairwise one versus one (1vs1) classification results. The problem is then transformed to the following. Given a set of events $A_1, A_2, \cdots, A_K$, some classifiers provide the pairwise probabilities $r_{ij} = P(A_i|A_i \text{or} A_j)$. Let us assume a model $P(A_i|A_i \text{or} A_j) = p_i/(p_i + p_j)$ with unknown parameters $p_i, i = 1, \cdots, K$, which forms the basis for the Bradley-Terry model for paired comparisons [12] [41]. Then the model is fitted by minimizing a Kullback-Leibler distance criterion to find the best approximation $\hat{r}_{ij} = \hat{p}_i/(\hat{p}_i + \hat{p}_j)$ to a given set of r_{ij}'s. This is carried out at each input $\mathbf{x}$, and use the estimated probabilities to predict the class membership at $\mathbf{x}$.

The algorithm of coupling the pairwise probabilities is as follows.

Let the class probabilities at input vector $\mathbf{x}$ be $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x})^T, \cdots, p_K(\mathbf{x}))$. In this section, the argument $\mathbf{x}$ is dropped since the calculations are done at each $\mathbf{x}$ separately.

We assume that for each $i \neq j$, there are n_{ij} observations in the training set and from these we have estimated conditional probabilities $r_{ij} = P(i|i \texttt{ or } j)$.

The model is

$$\mu_{ij} = \frac{p_i}{p_i + p_j} \tag{2.65}$$

or equivalently

$$\log \mu_{ij} = \log(p_i) - \log(p_i + p_j), \tag{2.66}$$

a log-linear model.

One needs to find estimates of p_i so that the $\hat{\mu}_{ij}$'s are close to the r_{ij}'s. There are $K-1$ independent parameters but $K(K-1)/2$ equations, so it is not possible in general to find $\hat{p}_i$'s so that $\hat{\mu}_{ij} = r_{ij}$ for all i, j. So we need to minimize the cost function which is the average (weighted) Kullback-Leibler distance between r_{ij} and μ_{ij}:

$$\ell(\mathbf{p}) = \sum_{i<j} n_{ij}[r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}}] \tag{2.67}$$

with $\mathbf{p} = (p_1, \cdots, p_K)^T$, and we find $\mathbf{p}$ to minimize this function.

The following iterative procedure is used to compute the $\hat{p}_i$'s:

1. Start with some guess for the $\hat{p}_i$, and corresponding $\hat{\mu}_{ij}$.
2. Repeat $(i = 1, 2, \cdots, K, 1, \cdots)$ until convergence:

$$\hat{p}_i \leftarrow \hat{p}_i \frac{\sum_{j \neq i} n_{ij} r_{ij}}{\sum_{j \neq i} n_{ij} \mu_{ij}} \tag{2.68}$$

   renormalize the $\hat{p}_i$, and recompute the $\hat{\mu}_{ij}$.
3. $\hat{\mathbf{p}} \leftarrow \hat{\mathbf{p}} / \sum \hat{p}_i$.

The procedure converges since $\ell(\hat{\mathbf{p}})$ is bounded above by zero and increases at each step [41].

The resulting classification rule is to assign the given input vector $\mathbf{x}$ to class i if $\hat{p}_i(\mathbf{x}) > \hat{p}_j(\mathbf{x}), \forall i \neq j$.

2.5.3 Ensemble methods for predictive modelling

The predictive performance of a model can sometimes be not nearly as strong on unseen data as that obtained on the training data. This phenomenon is often described as overfitting, where the model is too specialized to the training data. Overfitting may arise due to the existence of high variance in the data. Empirical results indicate that the variance can be greatly reduced by inducing multiple models (an ensemble) from the same data. The classification of an unseen case is then determined by for example a weighted combination of the classifications assigned by the multiple models.

Techniques such as bagging, stacking, and boosting, have been developed to utilize ensembles of models in a predictive context [24]. These methods either combine models developed using different techniques on the same data, or combine models developed using the same technique but on different subsets of the data. In this thesis, we experimented on combining models built with the same algorithms, but based on different bootstrap samples as well as different input variables selected based on the different bootstrap samples.

In the case of ensemble of probabilistic classifiers for data with high dimensionality with a relatively small sample size, the key step in inducing multiple models is in the *sampling* of the training data and selection of the input variables on different samples. In one simple approach, called *bagging* [26], a sample of size N is taken with replacement from the original set of N examples. Some examples will be repeated in the sample, others may not occur. Thus, it is possible to generate many samples and induce classifiers from each sample.

An alternative method of sampling, *boosting* usually performs better than bagging. Instead of sampling all cases randomly, so that each data point has a $1/N$ chance of being drawn from the sample, an incremental approach is used to bias the selection. The objective is to increase the odds of selecting examples that have been erroneously classified by the models that have been induced in previous iterations. The concept of boosting, i.e. adaptive resampling, applies to many learning methods. However, resampling is particularly advantageous when used in conjunction with some unstable variable selection algorithms, such as sparse Bayesian logit models and decision trees. There are two key reasons for this: sparse Bayesian learning logit model and decision tree algorithms are relatively efficient in high dimensions, and they tend to have a bigger component of model variance than other methods like kernel-based models or neural networks.

We have explored applications of bootstrapping to variable selection and modeling building in predictive modeling for cancer diagnosis, and have observed significant improvements in the model performance.

2.6 Model evaluation

An intelligent predictive model needs careful and reliable validation, especially for its generalization performance.

2.6.1 Performance measure

Accuracy is the most commonly used performance measure for a machine learning classification system, it is defined as the rate of correct classification, under the assumptions of equal misclassification costs and constant class distribution in the target environment. The use of the overall classification accuracy as an evaluation metric is adequate provided the class distribution among examples is constant and relatively balanced. Moreover, this measure assumes equal error costs, i.e. in case of binary classification, a false positive error is equally significant as a false negative error. Unfortunately, both assumptions are rarely satisfied in real life problems. All tests could have an error rate and on occasion will either fail to identify an abnormality, or identify an abnormality which is not present. Sensitivity and specificity are just two measures that can quantify these two situations.

Assume a dichotomic classifier generates a numerical output $y(\mathbf{x})$, given input $\mathbf{x}$. The binary results of positive or negative values can be obtained by defining the normal or abnormal range of results. For example, the ultimate decision is taken by comparing the output $y(\mathbf{x})$ with a certain cutoff value. If the output is beyond a threshold, the test case can be classified into the positive class, otherwise, into a negative class. See Figure 2.3 for a graphical illustration of a binary classification using a diagnostic test. The dotted and solid curves represent the distribution of the test output value $y(\mathbf{x})$ for negative and positive class respectively.

It is then common to describe the classification performance in four categories: False Positive (FP) if the system labels a negative case as positive; False Negative (FN) if the system labels a positive case as negative; True Positive (TP) and True Negative if the system correctly predicts the label. In the following, *TP, TN, FP, FN* are used to denote the number of cases of true positive, true negative, false positive and false negative cases. A contingency table (or confusion matrix) can be used to summarize the performance of a learning machine (here "+" stands for positive, and "−" for negative).

A tabular format, called contingency table is illustrated in Table 2.2.

Traditional accuracy measures of a diagnostic test include sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV). The sensitivity or true positive rate of a learning machine is defined as the

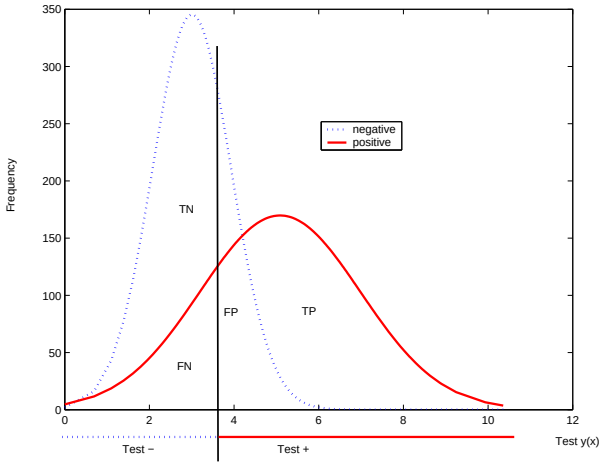


Figure 2.3: An illustration of classification using a diagnostic test.

	True	
Test result	−	+
−	TN	FN
+	FP	TP

Table 2.2: Contingency table used for evaluation of the performance of a binary classifier, or a diagnostic test in clinical medicine.

proportion of the positive cases that are predicted to be positive:

$$\text{sensitivity} = \frac{\#\text{ positive correctly classified}}{\#\text{ total positive}} = \frac{TP}{TP+FN}. \tag{2.69}$$

The specificity or true negative rate is defined as the proportion of the negative cases that are predicted to be negative:

$$\text{specificity} = \frac{\#\text{ negative correctly classified}}{\#\text{ total negative}} = \frac{TN}{TN+FP}. \tag{2.70}$$

The overall accuracy is then the ratio between the number of correctly classified instances and the number of test cases:

$$\text{accuracy} = \frac{\#\text{ instances correctly classified}}{\#\text{ total instances}} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{2.71}$$

The positive predictive value (PPV) is the fraction of all cases with a positive test result that are true positive. It is given by $TP/(TP+FP)$. Conversely, the

negative predictive value (NPV) is the fraction of all cases with a negative test result that are actually negative. It is given by $TN/(TN+FN)$. Both the PPV and the NPV are strongly influenced by the disease prevalence in the population. Although it is commonly stated that both the sensitivity and specificity are not affected by population characteristics due to the fact that they depend on only cases with disease or without the disease. In fact, however, empirical sensitivity and specificities are commonly found to vary between populations, because of variability in other factors which affect the test performance.

In the diagnostic test disease case, we wish to quantify the probability of a given patient having the disease, based on whether or not they had a positive or negative test result, relative to the chance of the patient having the disease before the test result is known. Thus the posttest probabilities are measured relative to pretest probabilities for all patients in the population.

To enhance the interpretation of the test results of our models, we would also use the measure of likelihood ratios. Likelihood ratios tell us how much we should shift our suspicion for a particular test result. Because tests can be positive or negative, there are at least two likelihood ratios for each test. The positive likelihood ratio (LR+) tells us how much to increase the probability of disease if the test is positive, while the negative likelihood ratio (LR−) tells us how much to decrease it if the test is negative. The formula for calculating the positive likelihood ratio is:

$$\text{LR+} = \frac{\text{probability of a positive case having a positive test result}}{\text{probability of a negative case having a positive test result}} \tag{2.72}$$

And the negative likelihood ratio is expressed in the formula:

$$\text{LR+} = \frac{\text{probability of a positive case having a negative test result}}{\text{probability of a negative case having a negative test result}} \tag{2.73}$$

We can also compute the LR+ and LR− in terms of sensitivity and specificity: LR+ = sensitivity / (1−specificity) and LR− = (1−sensitivity) / specificity.

Unlike predictive values, likelihood ratios are independent on the prevalence of the disease, thus they are more helpful and informative for clinicians in evaluating and deciding whether or not to use the diagnostic model.

The Receiver Operating Characteristics (ROC) curve analysis is a classical method in the area of signal detection theory, and has been extended for use in statistics, medical diagnosis and visualization in medicine [98][38][20], and in machine learning as an alternative method for comparing learning systems [80], or generally evaluating the performance of a classifier.

The ROC curve is used for visualizing the performance of a classifier, where the sensitivity (true positive rate) is plotted against 1−specificity (false positive rate). In this way, classifiers can be compared by a point in a plane. For

a classifier obtained by thresholding a real valued function or depending on a real parameter, the ROC curve is constructed by plotting the sensitivity versus the false positive rate, for varying cutoff (i.e., threshold) values. The perfect diagnostic test should have 100% sensitivity and 0 false positive (or 100% specificity) reaching the upper left corner of the graph. The lower left point (i.e. 0% sensitivity and 100% specificity) represents the strategy of always alarming. A diagonal ROC curve (dashed line in figure 2.4) indicates a diagnostic test which does not produce any useful differentiation between different classes and it is equivalent to the randomly guessing the class. Informally, one point in ROC space is better than another if it is higher and closer to the upper left corner.

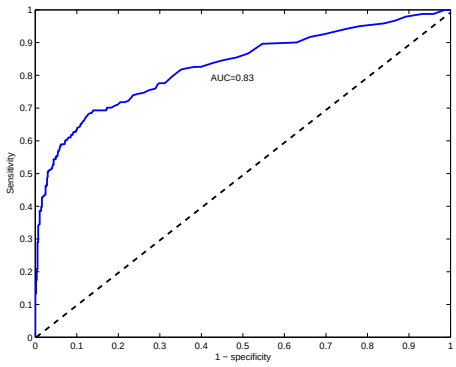


Figure 2.4: An example of the ROC curve for a diagnostic test.

The area under the ROC curve (AUC) is a one-value measure of the accuracy of a test. It can be statistically interpreted as the probability of the classifier to correctly classify positive cases and negative cases. That is, let $\mathbf{x}_+$ and $\mathbf{x}_-$ denote the random sample from the positive and negative class, respectively, then AUC $= P(y(\mathbf{x}_-) < y(\mathbf{x}_+))$. The higher the AUC on the test data, the better the classifier or the test. In this thesis work, the area under the ROC curves was obtained by a non-parametric method based on the Wilcoxon statistic, using the trapezoidal rule, to approximate the area [38]. This method also gives a standard error that can be used for comparing two different ROC curves [20].

2.6.2 Validation strategies

Data splitting and cross-validation

To evaluate a model we would like to know the generalization rate of a classifier on a given problem. This way we could see if the classifier performs well enough to be useful; and we can compare its performance with that of competing design.

Cross-validation (CV) is a method of estimating generalization performance

based on resampling, and the obtained results are often used for model comparison.

The holdout method is often used for cross-validation, it partitions the data into two mutually exclusive subsets, namely a training set and a validation set. One can also repeat the holdout method K times. Each time a different partition is chosen. Then the estimated accuracy estimate such as AUC is derived by averaging over all the runs. However, in medical practice, the holdout method makes inefficient use of the data set, which is usually smaller than desired.

For example, one third of the data set is not used for training the classifier. K-fold cross-validation is a variant of cross-validation. The data set is randomly divided into $K(K > 1)$ mutually exclusive subsets (K folds) of approximately equal size. The model is trained on all the subsets except for one, and the validation accuracy measure is obtained by testing it on the subset left out. This procedure is repeated K times, each time using a different subset for validation. The performance of the model is assessed by averaging the accuracy measures under validation over the K estimates. Repeating the K-fold cross-validation for multiple runs can provide a better statistical estimate. The cross-validation estimate is a random number that depends on the division of the data set. We hope that the estimates have low bias and low variance.

Leave-one-out (LOO) cross-validation is a special K-fold cross-validation, in which the number of folds equals the number of available data. This method is almost unbiased, but has high variance, leading to unreliable estimates. When choosing the number of folds, we would like to tradeoff bias for low variance.

Cross-validation maybe used to estimate the generalization performance of a given model, or for model selection by choosing the model that has the best generalization performance. Hence cross-validation could be used for choosing the structure (number of hidden neurons) of a multilayer perceptron, or a subset of input variables, or to determine the kernel parameters and regularization parameters for support vector machines.

Bootstrapping

Another approach is bootstrapping, which is useful when the data set is very small. A "bootstrap" data set is one created by randomly selecting M points from the training set $\mathcal{D}$ with replacement [26]. In bootstrap estimation, this selection process is independently repeated B times to yield B bootstrap data sets, which are treated as independent sets. There are several ways to use bootstrap method to problem of accuracy estimation. One of the commonly used approaches is to fit the model on each bootstrap sample, and the performance is measured both on the bootstrap sample and on the original sample. The average difference between the two performance measures forms an estimate of the optimism. The bias corrected performance measure was then calculated by

subtracting the optimism from the measure of the model built on the original data. In practice, the high computational complexity of bootstrap estimation of classifier accuracy is rarely worth possible improvements in that estimate [24].

2.6.3 Validation of medical diagnostic models

For validation of a diagnostic or prognostic model in clinical practice, and from the study design point of view, there are in general a hierarchy of increasingly stringent validation strategies:

1. internal - procedures restricted to a single data set;
2. temporal - evaluation on a second data set from the same center(s);
3. external - evaluation on data from one or more other centers.

Most of the learning machine validation strategies fall into the first category by means of the aforementioned data splitting or cross-validation, or bootstrapping in a single data set. Although cross-validation is widely recommended, we still need to be aware of the way of splitting which might influence the validation results. Randomly splitting must lead to data set that are the same other than for chance variation and is thus a weak procedure. Furthermore, estimates of predictive accuracy from data-splitting procedures, though unbiased, tend to be imprecise [26]. A tougher test is to split the data in a non-random way, which might reflect the real situation when applied in clinical practice. For example, we might take groups of patients seen in different time periods. The second type of validation temporal validation is no different in principle from the data splitting according to time period. However, it is at least a prospective evaluation, independent of the original data and model fitting process. External validation is also desirable since it evaluates a model on new data collected from an appropriate patient population in a different center, and the goal of validation here is actually to demonstrate satisfactory performance for patients from a different population than the original on. Some design issues such as sample selection and sample size are important here, however, these are out of the scope of this thesis.

2.7 Summary

In this chapter, we have described some major supervised learning algorithms for classification. We briefly outline the conventional linear classifiers including Fisher's linear discriminant and logistic regression. The advanced machine learning algorithms for classification such as MLPs, RBF neural networks,

SVMs, and LS-SVMs have been reviewed. The problem of parameter estimation and model selection have been raised concerning the generalization ability of the predictive models. Least squares solutions, maximum likelihood and maximum a posterior estimate are linked in the context of regularization. Support vector classifiers are supported by the structural risk minimization principle. The use of the kernel trick has made it possible to use the linear algorithms to obtain in the end a much richer (even nonlinear) decision surface. Compared with feed-forward neural networks such as MLPs and RBF-NNs, LS-SVM learning algorithm is advantageous, which has a global and unique solution to the optimization problem, and uses less nuisance hyperparameters in determination of the model structure.

It is also pointed out that an additive kernel-based classifier retains the interpretability of additive models, while it can still keep some nonlinear effects in the models. Moreover, a kernel-based model with linear kernels can be represented in the way of linear additive models, and thus can be understood in the same way as normal linear classifiers. LS-SVMs have also been linked with kernel PCR for the sake of model visualization and interpretation.

One important issue in classification is the selection of the input variables, which is desired not only for interpretation of the classifier but also the generalization performance. There are in general three approaches for variable selection: filter, wrapper, and embedded approach. The approaches to multiclass classification problems have been discussed, with emphasis on the pairwise coupling methods. Finally, an overview on the strategies and performance measures for validation of predictive models have been given.

So far, we have reviewed a series of modelling techniques, however, to achieve a high level of performance for the models, we still need to: 1) choose the appropriate basis functions, related to e.g. the number of hidden neurons in a neural network, the kernel type and kernel parameters in kernel-based models, the input variables to be used in the models; and 2) find the optimal hyperparameters, such as the regularization parameters. These issues all concern model selection, and will be addressed in the next chapter by exploiting Bayesian learning methods.

Chapter 3

Bayesian frameworks for blackbox models

This chapter describes how to use Bayesian learning techniques for various linear and nonlinear models, particularly for artificial neural networks and least squares support vector machines. After an overview of the Bayesian learning principle and the Bayesian framework for neural networks pioneered by MacKay, we describe in detail the sparse Bayesian learning algorithms for regression and classification, which employ an automatic relevance determination (ARD) method to obtain sparse solutions. When the basis functions are taken as the original variables, the sparse solution entails the automatic selection of the input variables. When the basis functions are kernels as for SVMs, the resulting sparse models are referred to as relevance vector machines (RVMs). Then we show how to integrate a Bayesian framework with the LS-SVM for automatic tuning of the hyperparameters and choice of the kernel parameters, and how to transform the output of the standard LS-SVM to the posterior class probabilities in a probabilistic framework. A pruning scheme has also been proposed to impose sparseness to the LS-SVM solutions, and show how the sparseness upon LS-SVM can help to improve the classification performance. The variances associated with the predicted output within the Bayesian framework for various models have also been examined.

3.1 Bayesian methods

3.1.1 Principle of Bayesian learning

Recently, Bayesian methods have become a viable alternative to maximum likelihood or maximum a posteriori (MAP) approaches [10, 61, 69, 95, 106]. The main advantages of Bayesian methods are:

- Automatic control of the model complexity
- Possibility to use prior information and hierarchical models for the hyperparameters
- Predictive distributions for outputs, from which the uncertainty of the prediction can be judged.

The key principle of the Bayesian approach is to define the probability distributions over all quantities within the model, and to update the distributions given data using Bayes' rule; and then construct posterior probability distributions for the parameters and hyperparameters in the models. Predictions can be made based on the posterior (marginal) distributions over all the parameters.

Evidence, Bayesian Occam's razor and prediction

As discussed in Section 2.1, given a model with unknown parameter vector $\boldsymbol{\theta}$, the maximum likelihood parameter estimate is the value $\hat{\boldsymbol{\theta}}$ that maximize the probability of the training data, that is, $p(\mathcal{D}|\hat{\boldsymbol{\theta}})$. The direct generalization of these techniques, leads to the MAP method, which finds the value that maximize the posterior probability of the parameters $p(\boldsymbol{\theta}|\mathcal{D})$ under a probabilistic framework. These generalized methods are sometimes called type II maximum likelihood methods, with the goal to choose the model that best explains the training data.

Let $\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_L$ represent a series of candidate hypothesizes or models and let $\mathcal{D}$ represent the training data. The posterior probability of any given model is given by Bayes rule:

$$P(\mathcal{H}_i|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{H}_i)P(\mathcal{H}_i)}{p(\mathcal{D})} \propto P(\mathcal{D}|\mathcal{H}_i)P(\mathcal{H}_i), \quad i = 1, \cdots, L \tag{3.1}$$

where the normalizing factor $p(\mathcal{D})$ is rarely needed. The data dependent term $P(\mathcal{D}|\mathcal{H}_i)$ is the *evidence* for $\mathcal{H}_i$; the factor $P(\mathcal{H}_i)$ is our subjective prior over the hypothesis space. In practice, equal priors are normally assigned to different hypothesis, $P(\mathcal{H}_i)$ are thus often neglected in the computation. In the evidence

framework, or type II maximum likelihood approaches, we find the maximum likelihood parameters for each of the candidate models, calculating the resulting *evidence*, and then select the model with the largest evidence.

Bayesian model comparison uses the full information over priors when computing posterior probabilities in (3.1). In particular, the evidence for a hypothesis $\mathcal{H}_i$ is obtained via integration, or marginalization over the parameters:

$$P(\mathcal{D}|\mathcal{H}_i) = \int p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{H}_i) p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{H}_i) d\boldsymbol{\theta}, \tag{3.2}$$

where $\boldsymbol{\theta}$ corresponds to the parameters in the candidate model. It is common for the posterior $P(\boldsymbol{\theta}|\mathcal{D}, \mathcal{H}_i)$ to be peaked at $\hat{\boldsymbol{\theta}}$, and thus the integral for the evidence can often be approximated as [61][24]

$$\begin{array}{rcccl} P(\mathcal{D}|\mathcal{H}_i) \simeq & \underbrace{P(\mathcal{D}|\hat{\boldsymbol{\theta}}, \mathcal{H}_i)} & \times & \underbrace{p(\hat{\boldsymbol{\theta}}|\mathcal{H}_i)\Delta\boldsymbol{\theta}}. & \\ \text{Evidence} \simeq & \text{best-fit likelihood} & \times & \text{Occam factor} & \end{array} \tag{3.3}$$

Before the data arrive, some broad range of values can be taken for $\boldsymbol{\theta}$, and is denoted by $\Delta^0\boldsymbol{\theta} \in \mathbb{R}$. The range of possible values for $\boldsymbol{\theta}$ changes to $\Delta\boldsymbol{\theta} \in \mathbb{R}$ after the data arrive. Suppose for simplicity that the prior $p(\hat{\boldsymbol{\theta}}|\mathcal{H}_i)$ is uniform on the range of $\Delta^0\boldsymbol{\theta}$. The Occam factor in (3.3) is then $\Delta\boldsymbol{\theta}/\Delta^0\boldsymbol{\theta}$, i.e., the ratio of the volume that can account for data $\mathcal{D}$ ($\Delta\boldsymbol{\theta}$) and the prior volume accessible to model independent of $\mathcal{D}$ ($\Delta^0\boldsymbol{\theta}$), and it can be written as

$$\text{Occam factor} = p(\hat{\boldsymbol{\theta}}|\mathcal{H}_i)\Delta\boldsymbol{\theta} = \frac{\Delta\boldsymbol{\theta}}{\Delta^0\boldsymbol{\theta}}. \tag{3.4}$$

The Occam factor has magnitude less than 1, and is thus a measure of complexity of the model. The full integral of (3.2) is not analytically or numerically available. However, if $\boldsymbol{\theta}$ is M-dimensional and its posterior can be assumed to be a Gaussian, then the Occam factor can be calculated directly, yielding

$$P(\mathcal{D}|\mathcal{H}_i) \simeq P(\mathcal{D}|\hat{\boldsymbol{\theta}}, \mathcal{H}_i) p(\hat{\boldsymbol{\theta}}|\mathcal{H}_i) (2\pi)^{M/2} (\det \mathbf{H})^{-1/2}, \tag{3.5}$$

where,

$$\mathbf{H} = \frac{\partial^2 \log p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{H}_i)}{\partial \boldsymbol{\theta}^2} \tag{3.6}$$

is a Hessian matrix. Note that this Gaussian approximation is based on the assumption that the evidence distribution arises from a large number of independent uncorrelated processes, and does not rely on the distribution of the data itself.

The predictive distribution is also an integral over the parameters $\boldsymbol{\theta}$:

$$p(y_*|\mathcal{D},\mathcal{H}_i) = \int p(y_*|\boldsymbol{\theta},\mathcal{D},\mathcal{H}_i)p(\boldsymbol{\theta}|\mathcal{D},\mathcal{H}_j)d\boldsymbol{\theta} \tag{3.7}$$

or

$$p(y_*|\mathcal{D},\mathcal{H}_i) = \int p(y_*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D},\mathcal{H}_i)d\boldsymbol{\theta}. \tag{3.8}$$

3.1.2 Bayesian inferences for a generic probabilistic model

Given a training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, suppose we need to train a model as defined in (2.1) and (2.2), $f(\mathbf{x};\mathbf{w}) = \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})$, and the additive noise on target value has variance σ^2. The fixed basis functions $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_M(\mathbf{x})]^T$ are used here, $\boldsymbol{\Phi}$ denotes the $N\times M$ design matrix with element $\boldsymbol{\Phi}_{nm} = \phi_m(\mathbf{x}_n)$. One can define the error term E_D as the sum of squared errors, and the weight penalty term E_W as the sum of squared weights. Under the assumption of i.i.d. data, the likelihood of the data set can be written as:[1]

$$p(\mathbf{y}|\mathbf{w},\sigma^2) = \prod_{n=1}^N p(y_n|\mathbf{w},\sigma^2) = \prod_{n=1}^N (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left[-\frac{(y_n - f(\mathbf{x}_n;\mathbf{w}))^2}{2\sigma^2}\right]. \tag{3.9}$$

Further assume the prior distribution over weights, which is $p(\mathbf{w}|\vartheta) \propto \exp(-\vartheta E_W)$ with ϑ the inverse variance of the priori Gaussian distribution for w. Given the likelihood and the prior distribution, we can compute the posterior distribution via Bayes' rule:

$$p(\mathbf{w}|\mathbf{y},\vartheta,\sigma^2) = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing factor}} = \frac{p(\mathbf{y}|\mathbf{w},\sigma^2)p(\mathbf{w}|\vartheta)}{p(\mathbf{y}|\vartheta,\sigma^2)}. \tag{3.10}$$

Here, the posterior follows also the Gaussian distribution: $p(\mathbf{w}|\mathbf{y},\vartheta,\sigma^2) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ with

$$\boldsymbol{\mu} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \sigma^2\vartheta\mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{y} \tag{3.11}$$

$$\boldsymbol{\Sigma} = \sigma^2(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \sigma^2\vartheta\mathbf{I})^{-1}. \tag{3.12}$$

The mean of the posterior distribution for the weights $\boldsymbol{\mu}$ is indeed obtained via the MAP method, which results into the same solution as the regularized least squares one with $\lambda = \sigma^2\vartheta$. However, the key difference is that in the Bayesian approach the prediction is made by integrating over the distribution of the model parameters $\mathbf{w}$, rather than based on a specific estimate for $\mathbf{w}$. This is the same as taking the average prediction of all the models weighted

[1] Here the likelihood is written as $p(\mathbf{y}|\mathbf{w})$, but not $p(\mathcal{D}|\mathbf{w})$. This is because we are modelling the probability of y with the condition of a given $\mathbf{x}$ and the model. It should be $p(\mathbf{y}|\mathbf{x},\mathbf{w},\sigma^2)$ then. But $\mathbf{x}$ will be omitted from the conditioning list for notational simplicity.

by their posterior probability, which typically results in improved predictive capability. Specifically, to predict the new output y_* given the new input $\mathbf{x}_*$, a full Bayesian predictive distribution is obtained by integrating (marginalizing) the predictions of the model with respect to the posterior distribution of the model parameters

$$p(y_*|\mathbf{x}_*, \mathbf{y}) = \int p(y_*|\mathbf{x}_*, \mathbf{w}, \sigma^2) p(\mathbf{w}, \vartheta, \sigma^2|\mathbf{y}) d\mathbf{w} d\vartheta d\sigma^2. \tag{3.13}$$

However, the integrations for getting $p(\mathbf{w}, \vartheta, \sigma^2|\mathbf{y})$ or $p(y_*|\mathbf{x}_*, \mathbf{y})$ are nearly always analytically intractable. This necessitates the use of approximation methods for marginalization, such as type II maximum likelihood, Laplace (Gaussian) approximation, variational methods, and Markov Chain Monte Carlo (MCMC) sampling techniques.

In our work, type II maximum likelihood approach, i.e. the evidence framework is used. Suppose that the point estimate is representative of the posterior, we can thus replace the integration over the hyperparameters by point estimates involving their most probable posterior values. Further using $p(\mathbf{w}, \vartheta, \sigma^2|\mathbf{y}) \equiv p(\mathbf{w}|\mathbf{y}, \vartheta, \sigma^2) p(\vartheta, \sigma^2|\mathbf{y})$ and (3.13), we see that the predictive distribution can be approximated by a distribution at the "most probable" values for the hyperparameters ϑ_{MP} and σ^2_{MP}. Under the assumption of flat, noninformative priors over $\log \vartheta$ and $\log \sigma^2$, we have $p(\vartheta, \sigma^2|\mathbf{y}) \propto p(\mathbf{y}|\vartheta, \sigma^2)$, then ϑ_{MP} and σ^2_{MP} can be found by maximizing

$$\begin{aligned} p(\mathbf{y}|\vartheta, \sigma^2) &= \int p(\mathbf{y}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\vartheta) d\mathbf{w} \\ &= (2\pi)^{-\frac{N}{2}} |\sigma^2 \mathbf{I} + \vartheta^{-1} \mathbf{\Phi} \mathbf{\Phi}^T|^{-\frac{1}{2}} \exp\left[-\frac{1}{2} \mathbf{y}^T (\sigma^2 \mathbf{I} + \vartheta^{-1} \mathbf{\Phi} \mathbf{\Phi}^T)^{-1} \mathbf{y} \right]. \end{aligned} \tag{3.14}$$

This is indeed a Gaussian distribution over the N-dimensional data vector $\mathbf{y}$. To determine ϑ_{MP} and σ^2_{MP}, some gradient based optimization methods are often applied. Some relations in the optimality condition can be exploited. As shown in [60], the variance of the distribution for w_m can be estimated by $1/\vartheta_{\mathrm{MP}} = \|\mathbf{w}_{\mathrm{MP}}\|^2/\gamma$, where γ is the effective number of parameters: $\gamma = M - \vartheta \mathrm{Trace} \mathbf{\Sigma}$. The quantity of γ is between 0 and the total number of parameters M. Similarly, in a regression problem with a Gaussian noise model, the variance of the target variable can be estimated by

Having found ϑ_{MP} and σ^2_{MP}, our approximation to the predictive distribution is then a Gaussian:

$$p(y_*|x_*, \mathbf{y}) \quad = \quad \int p(y_*|x_*, \mathbf{w}, \sigma^2_{\mathrm{MP}}) p(\mathbf{w}|\mathbf{y}, \vartheta_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}}) d\mathbf{w} = \mathcal{N}(\mu_*, \sigma^2_*)$$

with $\boldsymbol{\mu}_* = f(\mathbf{x}; \boldsymbol{\mu})$, and $\sigma^2_* = \sigma^2_{\mathrm{MP}} + \boldsymbol{\phi}(\mathbf{x}_*)^T \mathbf{\Sigma} \boldsymbol{\phi}(\mathbf{x}_*)$. Intuitively, the mean predictor $\boldsymbol{\mu}_*$ is the model evaluated at the posterior mean weights, and the

predictive variance σ_*^2 is the sum of the variances associated with both the noise process and the uncertainty in the weight estimates.

Bayesian model comparison

The model selection here involves mainly the choice of basis functions, which is represented by a hypothesis $\mathcal{H}_i$ with prior probability $p(\mathcal{H}_i)$. To be more general, the model hypothesis might describe the selection of input variables, the structure for a neural network, the kernel type and parameters such as the bandwidth r of an RBF kernel for SVM or RVM models, etc. Different models can be compared by examining their posterior $p(\mathcal{H}_i|\mathbf{y}) = p(\mathbf{y}|\mathcal{H}_i)p(\mathcal{H}_i)/p(\mathbf{y})$. Assuming a uniform prior $p(\mathcal{H}_i)$ over all models, the models can be ranked by the marginal likelihood, or model evidence $p(\mathbf{y}|\mathcal{H}_i)$, which can be evaluated using a Gaussian approximation. In this study, the model hypothesis that maximizes the model evidence is chosen from a set of candidates.

Statistical interpretation is also available for the comparison between two models in the Bayesian framework. Bayes factor B_{10} for model $\mathcal{H}_1$ against $\mathcal{H}_0$ from data is defined as $B_{10} = p(\mathbf{y}|\mathcal{H}_1)/p(\mathbf{y}|\mathcal{H}_0)$. Under the assumption of equal model priors, the Bayes factor can be seen as a measure of the evidence given by the data in favor of a model compared to a competing one. When the Bayes factor is greater than 1, the data favor $\mathcal{H}_1$ over $\mathcal{H}_0$; otherwise, the reverse is true. The rules of thumb for interpreting $2\log B_{10}$ include: the evidence for $\mathcal{H}_1$ is very weak if $0 \leq 2\log B_{10} \leq 2.2$, and the evidence for $\mathcal{H}_1$ is decisive if $2\log B_{10} > 10$, etc, as shown in [46].

Next, we discuss some implementation issues in integrating Bayesian learning with various black-box modelling techniques. This is a review of the work from several researchers. MacKay's evidence framework has provided a unified theoretical treatment of learning in feedforward neural networks [60][10]. Relevance vector machines, a special case of sparse Bayesian modelling with kernel basis, by means of automatic relevance determination within the evidence framework, has been introduced by Tipping recently and appears to yield very sparse solutions and obtain good empirical performance as well [106][105][9]. The evidence framework has also been applied to LS-SVM by Van Gestel and Suykens, achieving similar test set results as Gaussian processes and SVMs in several benchmark problems [111][95].

3.2 Bayesian multilayer perceptrons

In the evidence framework for MLP learning, the posterior distribution for parameters can be only approximated locally by a Gaussian. This is in contrast with the linear parametric model in Section 3.1.2, where the Gaussian distri-

bution is exact for the posterior of parameters. The approximation in Bayesian MLP modelling takes a Gaussian distribution with the mean being the "most probable" value $\mathbf{w}_{\mathrm{MP}}$, and a covariance matrix: $\mathbf{\Sigma} = \mathbf{H}^{-1}$:

$$p(\mathbf{w}|\mathbf{y}, \vartheta, \sigma^2) \cong p(\mathbf{w}_{\mathrm{MP}}|\mathbf{y}, \vartheta, \sigma^2) \exp\left[-\frac{1}{2}(\mathbf{w} - \mathbf{w}_{\mathrm{MP}})^T H (\mathbf{w} - \mathbf{w}_{\mathrm{MP}})\right], \quad (3.15)$$

where the Hessian $\mathbf{H}$ is evaluated at $\mathbf{w}_{\mathrm{MP}}$, $\mathbf{H} = -\nabla\nabla \log p(\mathbf{w}|\mathbf{y}, \vartheta, \sigma^2)|_{\mathbf{w}_{\mathrm{MP}}}$, and $\mathbf{w}_{\mathrm{MP}}$ is normally obtained by a gradient based optimization approach.

Therefore the predictive distribution becomes $p(\mathbf{y}_*|\mathbf{y}) \approx N(\mu_*, \sigma_*^2)$ with mean $\mu_* = f(\mathbf{x}_*; \mathbf{w}_{\mathrm{MP}})$, and variance $\sigma_*^2 = \sigma_{\mathrm{MP}}^2 + \boldsymbol{\psi}^T \mathbf{H}^{-1} \boldsymbol{\psi}$ with $\boldsymbol{\psi} = \frac{\partial f}{\partial \mathbf{w}}|_{\mathbf{x}_*, \mathbf{w}_{\mathrm{MP}}}$.

For classification problems, the *moderated* output can be calculated for MLP with logistic sigmoid output activation function. The moderated output probability can account for the uncertainty in weights (by using a simple numerical approximation), which is in contrast with the output probability using directly the most probable network output [60].

Also worth noticing is the following. It is a good practice to assign different regularization hyperparameters to the weights and the bias of different layers. For a two layer MLP, for instance, we could use four different values of hyperparameters $\vartheta_i, i = 1, \ldots, 4$ for each layer of weights and bias.

3.3 Sparse Bayesian learning models

3.3.1 Introduction

Sparse Bayesian learning (SBL) is the application of Bayesian automatic relevance determination (ARD) to models linear in their parameters in the form of (2.2), by which the sparse solutions to the regression or classification tasks can be obtained [106, 9]. Although the sparse Bayesian learning methods can be applied to general models with arbitrary sets of basis functions, only two forms of basis functions $\phi_m(\mathbf{z})$ are considered here, namely $\phi_m(\mathbf{z})$=$z_m, m = 1, \cdots, D$, (i.e. $\phi_m(\mathbf{z})$ as the original input variable), or $\phi_m(\mathbf{z})$=$k(\mathbf{z}, \mathbf{x}_m), m = 1, \cdots, N$, where $k(\cdot, \cdot)$ denotes some symmetric kernel function. $\phi_0(\mathbf{z})$ is always set to 1 in order to include an intercept (bias) term in the model. The sparse Bayesian learning model using the first form of basis functions, which are referred to as the linear basis functions. In contrast, the relevance vector machines (RVMs) take the second form, i.e. the kernel representation, for the basis function. For RVMs, kernels can be any symmetric kernel functions, though kernels in LS-SVMs must be positive definite and satisfy Mercer's condition. This is because RVM is a parameterized kernel model while LS-SVMs have primal-dual representations.

In Section 3.1.2, we have depicted the Bayesian inference where the hyperparameter ϑ is set to be the same value for all the parameters. In order to encode the preference of sparsity in parameter estimation, one can retain the traditional Gaussian prior for weights, but use M independent hyperparameters $\boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_M)$ to control the prior distribution for $M = N + 1$ weights. Note that for standard SVMs for regression, sparsity is obtained by Vapnik's ϵ-insensitive loss function. The Gaussian prior is modified to be:

$$p(\mathbf{w}|\boldsymbol{\vartheta}) = \prod_{m=1}^{M} \mathcal{N}(w_m|0, \vartheta_m^{-1}), \tag{3.16}$$

and one can place the previously utilized log-uniform hyperpriors over $\boldsymbol{\vartheta}$, which is given by a special case of the Gamma distribution. It is illustrated in [106] and [9] that the marginal prior $p(w_m) = \int p(w_m|\vartheta_m)p(\vartheta_m)d\vartheta_m \propto 1/|w_m|$. This is equivalent to the use of a penalty function: $E_W = \sum_m \log|w_m|$, which encourages sparsity.

Having defined the hierarchical prior, Bayesian inference then proceeds as previously. In the regression version, the likelihood is still a Gaussian. The posterior distribution over $\mathbf{w}$ is also Gaussian: $p(\mathbf{w}|\mathbf{y}, \boldsymbol{\vartheta}, \sigma^2) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma} = (\sigma^{\mathbf{2}}\boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\Phi} + \mathbf{A})^{-\mathbf{1}}$, $\boldsymbol{\mu} = \sigma^2\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{y}$, $\mathbf{A} = \mathrm{diag}(\vartheta_1, \ldots, \vartheta_M)$. The hyperparameters are sought by maximizing the marginal likelihood (evidence)

$$p(\mathbf{y}|\boldsymbol{\vartheta}, \sigma^2) = (2\pi)^{-N/2}|\mathbf{C}|^{-1/2}\exp(-\frac{1}{2}\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}), \tag{3.17}$$

with $\mathbf{C} = \sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T$. By differentiating $\log p(\mathbf{y}|\boldsymbol{\vartheta}, \sigma^2)$, one can get the re-estimation formulas: $\vartheta_m^{\text{new}} = \gamma_m/\mu_m^2$, and $(\sigma^2)^{\text{new}} = \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2/(N - \sum_{m=1}^{M}\gamma_m)$, where $\gamma_m = 1 - \boldsymbol{\Sigma}_{mm}$ is a measure of effectiveness of parameter w_m.

In sparse Bayesian classification, the logistic sigmoid link function is utilized and the likelihood is given by the Bernoulli distribution as in (2.9). And the Gaussian approximation is used for computing the weight posterior $p(\mathbf{w}|\mathbf{y}, \boldsymbol{\vartheta})$ or the marginal likelihood $p(\mathbf{y}|\boldsymbol{\vartheta})$. In an iteratively reweighted least squares algorithm, for the current value of $\boldsymbol{\vartheta}$, the most probable weights $\widehat{\boldsymbol{\mu}}$ is given by:

$$\widehat{\boldsymbol{\Sigma}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \mathbf{A})^{-1}, \tag{3.18}$$

$$\widehat{\boldsymbol{\mu}} = \widehat{\boldsymbol{\Sigma}}\boldsymbol{\Phi}^T\mathbf{B}\widehat{\boldsymbol{y}}, \tag{3.19}$$

$$\text{and } \widehat{\boldsymbol{y}} = \boldsymbol{\Phi}\widehat{\boldsymbol{\mu}} + \mathbf{B}^{-1}(\mathbf{y} - g(\boldsymbol{\Phi}\widehat{\boldsymbol{\mu}})),$$

with $\mathbf{B} = \mathrm{diag}(\beta_1, \beta_2, \ldots, \beta_n)$, where β_n, the inverse noise variance for ϵ_n is given by $g(f(\mathbf{x}_n; \widehat{\boldsymbol{\mu}}))[1 - g(f(\mathbf{x_n}; \widehat{\boldsymbol{\mu}}))]$.

This optimization process can be performed efficiently using an iterative re-estimation procedure and utilizing the conditions on the maximum of $p(\mathbf{y}|\boldsymbol{\vartheta}, \sigma^2)$ [106][9]. A fast sequential learning algorithm has also been introduced in [105], which enables us to efficiently process data of high dimensionality. We have adapted this algorithm to our applications, which will be detailed in the next subsection.

Practically, it has been observed that the optimum value of many of the ϑ_m will be infinite, implying that the corresponding weight estimate will be zero. The most relevant variables for the classifier can be obtained from the resulting sparse solutions, if the original variables are taken as basis functions in the SBL model. This type of models is referred to as linear SBL models in this thesis.

3.3.2 An algorithm for sequential sparse Bayesian learning

The sequential SBL algorithm [105] starts from a zero basis, adds or deletes a basis function at each iteration step, or updates a hyperparameter ϑ_m until convergence.

For optimization of the hyperparameters $\boldsymbol{\vartheta}$, the objective function uses the logarithm of the marginal likelihood $\mathcal{L}(\boldsymbol{\vartheta}) = \log p(\mathbf{y}|\boldsymbol{\vartheta}, \sigma^2)$. It is shown in [105][9] that we could analyze the properties of $\mathcal{L}(\boldsymbol{\vartheta})$ by decomposing it into the marginal likelihood $\mathcal{L}(\boldsymbol{\vartheta}_{-i})$ with $\boldsymbol{\phi}_i$ (the ith column of $\boldsymbol{\Phi}$) excluded, and the marginal likelihood $\ell(\vartheta_i)$ including only $\boldsymbol{\phi}_i$. That is $\mathcal{L}(\boldsymbol{\vartheta}) = \mathcal{L}(\boldsymbol{\vartheta}_{-i}) + \ell(\vartheta_i)$, where $\ell(\vartheta_i) = \frac{1}{2}[\log \vartheta_i - \log(\vartheta_i + s_i) + \frac{q_i^2}{\vartheta_i + s_i}]$, $s_i = \boldsymbol{\phi}_i^T \mathbf{C}_{-i}^{-1} \boldsymbol{\phi}_i$ and $q_i = \boldsymbol{\phi}_i^T \mathbf{C}_{-i}^{-1} \mathbf{y}$, $\mathbf{C}_{-i}$ is $\mathbf{C}$ with the contribution of $\boldsymbol{\phi}_i$ removed. Since s_i and q_i are independent of ϑ_i, one can obtain a unique maximum of $\mathcal{L}(\boldsymbol{\vartheta})$ with respect to ϑ_i by setting the first derivative of $\ell(\vartheta_i)$ to zero. The optimal values for ϑ_i are:

$$\tilde{\vartheta}_i = \frac{s_i^2}{q_i^2 - s_i}, \text{ if } q_i^2 > s_i, \tag{3.20}$$

$$\tilde{\vartheta}_i = \infty, \text{ if } q_i^2 \leq s_i. \tag{3.21}$$

One convenient way to derive s_i and q_i is to utilize these expressions: $s_i = \frac{\vartheta_i S_i}{\vartheta_i - S_i}$, $q_i = \frac{\vartheta_i Q_i}{\vartheta_i - S_i}$, and when $\vartheta_i = \infty$, $s_i = S_i$ and $q_i = Q_i$. In practice the two quantities S_i and Q_i are computed using the following equations:

$$S_i = \boldsymbol{\phi}_i^T \mathbf{C}^{-1} \boldsymbol{\phi}_i = \boldsymbol{\phi}_i^T \mathbf{B} \boldsymbol{\phi}_i - \boldsymbol{\phi}_i^T \mathbf{B} \boldsymbol{\Phi} \hat{\Sigma} \boldsymbol{\Phi}^T \mathbf{B} \boldsymbol{\phi}_i \tag{3.22}$$

$$Q_i = \boldsymbol{\phi}_i^T \mathbf{C}^{-1} \hat{\mathbf{y}} = \boldsymbol{\phi}_i^T \mathbf{B} \hat{\mathbf{y}} - \boldsymbol{\phi}_i^T \mathbf{B} \boldsymbol{\Phi} \hat{\boldsymbol{\mu}}, \tag{3.23}$$

where $\boldsymbol{\Phi}$, $\boldsymbol{\Sigma}$ and $\hat{\boldsymbol{\mu}}$ contain only the parts corresponding to the basis functions included in the model (with $\vartheta_m < \infty$). In the regression framework, $\mathbf{B} \equiv \sigma^{-2}\mathbf{I}$,

$\hat{\boldsymbol{\Sigma}} \equiv \boldsymbol{\Sigma} = (\sigma^2 \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \mathbf{A})^{-1}$, $\hat{\boldsymbol{\mu}} \equiv \boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{y}$, and $\hat{\mathbf{y}} \equiv \mathbf{y}$. Here $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ correspond to the mean and variance of the posterior distribution over $\mathbf{w}$, which is also Gaussian. In the case of classification utilizing a logistic link function, the weight posterior $p(\mathbf{w}|\mathbf{y}, \boldsymbol{\vartheta})$ is however not analytically available, but can be computed using a Gaussian approximation. For the current value of $\boldsymbol{\vartheta}$, we can estimate the quantities of interest through an iteratively reweighted least squares (IRLS) algorithm such as the Newton-Raphson method. And the expressions in (3.18) can be exploited.

The marginal likelihood maximization algorithm jointly optimizes the weights and the hyperparameters $\{\vartheta_m\}_{m=0}^{M_{\text{all}}}$, with M_{all} the maximum index for the basis functions. In case of linear basis functions, $M_{\text{all}} = d$; in case of linear kernel basis functions $M_{\text{all}} = N$. Define the complete set of possible indices for the basis functions as $\mathcal{I}_{\text{all}}$, containing the integer numbers from 0 to M_{all}. Our modified algorithm for classification utilizing the logistic function is as follows.

1. Initialize the model with only an intercept: $\vartheta_0 < \infty$ (e.g., $\vartheta_0 = (\mathbf{y}^T \mathbf{y}/N)^{-2}$), and $\forall m > 0, \vartheta_m = \infty$. Initialize the index set of the bases in the model $\mathcal{I}_{\text{sel}} \leftarrow \{0\}$.

2. Given the current $\boldsymbol{\vartheta}$ value, estimate $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$ using the IRLS algorithm for the logit model. Note that $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$ are only related to the basis functions included in the current model, initially with only one scalar element. And $\boldsymbol{\Phi}$ starts with only one column vector $\boldsymbol{\phi}_0 = [1, \cdots, 1]_{1\times N}^T$.

3. Randomly select M_{out} bases with the indices of $\mathcal{I}_{\text{out}} \subseteq (\mathcal{I}_{\text{all}} \setminus \mathcal{I}_{\text{sel}})$. Set $\mathcal{I}_{\text{can}} \leftarrow \mathcal{I}_{\text{out}} \cup \mathcal{I}_{\text{sel}}$.

4. For each basis vector in the candidate sets $\boldsymbol{\phi}_m, m \in \mathcal{I}_{\text{can}}$, compute the value of s_m and q_m, find out the optimal action with respect to each ϑ_m, then calculate ∇_m, the corresponding change in marginal likelihood $\mathcal{L}(\boldsymbol{\vartheta})$ after taking that action. The following rules are used:

   - If $q_m^2 > s_m$ and $\vartheta_m < \infty$ (i.e. $\boldsymbol{\phi}_i$ is in the model), estimate $\tilde{\vartheta}_m$ using (3.20), $\nabla_m = \frac{Q_m^2}{S_m + [\tilde{\vartheta}_m^{-1} - \vartheta_m^{-1}]^{-1}} - \log\{1 + S_m[\tilde{\vartheta}_m^{-1} - \vartheta_m^{-1}]\}$.
   - If $q_m^2 > s_m$ and $\vartheta_m = \infty$, add $\boldsymbol{\phi}_m$ to the model, compute $\tilde{\vartheta}_m$ using (3.20), $\nabla_m = \frac{Q_m^2 - S_m}{S_m} + \log \frac{S_m}{Q_m^2}$.
   - If $q_m^2 \leq s_m$ and $\vartheta_m < \infty$, then delete $\boldsymbol{\phi}_m$, setting $\tilde{\vartheta}_m = \infty$, $\nabla_m = \frac{Q_m^2}{S_m - \vartheta_m} - \log(1 - \frac{S_m}{\vartheta_m})$.

5. Select one basis $m* = \arg\max \nabla_m \mathcal{L}$, and take the corresponding action, i.e. $\vartheta_{m*} \leftarrow \tilde{\vartheta}_{m*}$ and update $\boldsymbol{\Phi}$.

6. If convergence is reached then stop, otherwise goto step 2.

The number of bases to be screened for updating ϑ_m is the number of bases in the model plus M_{out}, the predefined number of randomly selected bases from those not used by the model. In our experiments, M_{out} ranged from 1 to 100 depending on the size and/or dimension of the training set. And in this thesis, the optimization procedure is considered to be converged when the maximum value of $|\log(\tilde{\vartheta}_m/\vartheta_m)|_{m\in\mathcal{I}_{\text{can}}}$ in step 4 is lower than 10^{-6}.

However, we should also be aware of the uncertainty involved in the basis function selection, which might result from the existence of multiple solutions, and the sensitivity of the algorithm to small perturbations of experimental conditions. Attempts to tackle this problem are for example bagging and committee machines. We will revisit this problem in Section 5.2.1 in Chapter 5.

3.3.3 Relevance vector machine classifiers

As mentioned previously, the relevance vector machine is a special case of sparse Bayesian learning models, in which the basis functions are given by kernel functions of the same type as for SVMs. The sequential learning algorithm introduced in Section 3.3.2 was again applied to the optimization of RVMs. The predicted probability of being positive for a given input $\mathbf{x}_*$ can be computed by using the logistic function:

$$p(y_* = 1|\mathbf{x}_*, \mathbf{y}, \boldsymbol{\vartheta}) = \frac{1}{1 + e^{-\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_*)}}. \tag{3.24}$$

Prior class probabilities can be incorporated into the model by adjusting the bias term w_0 to $w_0 - \log(N_+/N_-) + \log(P_+/P_-)$, where $N+$ and N_- denote the number of positive and negative cases in the training set, P_+ and P_- the prior class probability for the positive and negative class, respectively.

For classification purposes, one can use a conceptually more correct likelihood function based on the Bernoulli distributions. However, no analytical expression is available for optimization of this likelihood function, instead, an iterative least squares algorithm is normally used to perform this optimizing task. This involves an approximation of the evidence as well. And the algorithm is much slower than the regression version. Therefore, alternatively, one can also utilize the similar approach as used by Bayesian LS-SVM classifiers, which is an regression approach to classification, and much a stable result might be obtained.

3.3.4 A toy example of using RVMs for classification

In this chapter we use the Ripley's synthetic binary classification problem [2] to illustrate how a certain method works. There are two variables and two

[2] The data sets are availabale from http://www.stats.ox.ac.uk/pub/PRNN/.

classes. Each class is in fact an equal mixture of two normal distributions with identical covariance matrix, and the class distributions were chosen to allow a best possible error rate (Bayes error rate) of about 8% [82]. The training set consists of 250 patterns with 125 patterns in each class. The testing set consists of 1000 patterns, with 500 patterns to each class. Note that, the examples on Ripley's synthetic data have all been normalized component-wise to achieve a mean zero and variance one on the training data.

Shown in Figure 3.1 is an example of using RVM classifier on Ripley's synthetic data. The contours in the figure correspond to the posterior probabilities for a point belonging to Class 2. RBF kernels are used and the width of the kernel is selected by maximizing the marginal likelihood of the model via coarse-to-fine searching. To cope with the local minima problem, the RVM modelling procedure, including kernel width searching, was repeated 5 times and the model with the highest model evidence was chosen. Notice that the RVM classifier utilizes only a very small number of data points in the model, the so-called relevance vectors (RVs) include only 4 training data, which can be viewed as a sort of prototypical data. This model achieved an error rate of 9.4% on the 1000 test data when taking 0.5 as the probability cutoff value. And the area under the ROC curve is 0.969 (SE=0.006).

Figure 3.2 illustrates the use of the RVM classifier with additive RBF kernels for

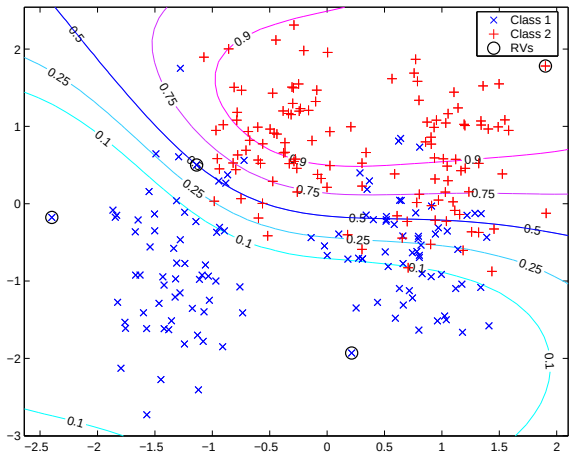


Figure 3.1: Classification of Ripley's synthetic data set using relevance vector classifiers with RBF kernels. The contours for the posterior probabilities for a point belonging to class 2 have been shown. The circled data are the 4 (4.4% of the number of training data) relevance vectors (RV) used in the final model. The classifier has an error rate of 9.4% on the 1000 test data when taking 0.5 as the probability cutoff, and the area under the ROC curve 0.969 (SE=0.006).

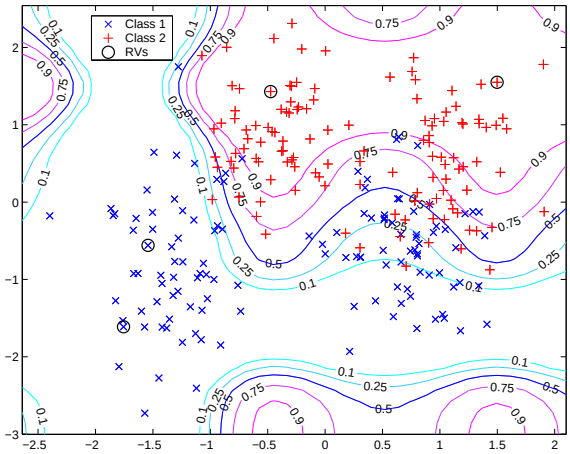


Figure 3.2: Classification of Ripley's synthetic data set using relevance vector classifiers with additive RBF kernels. The contours for the posterior probabilities for a point belonging to class 2 have been shown. The circled data are the 4 (1.6%) relevance vectors (RV) used in the final model. The classifier has an error rate of 9.3% on the 1000 test data when taking 0.5 as the probability cutoff, and the area under the ROC curve is 0.969 (SE=0.006).

classification of Ripley's data. It can be observed that the univariate additive kernels, can be calibrated to obtain roughly the same decision boundary as the multivariate kernels, in the central regions of the training data. The width of the univariate RBF kernel is also determined by maximizing the model evidence via searching within a range of candidates. The error rate is 9.3% on the test set, and the AUC is 0.969, which is approximately the same as for the RVM classifier using multivariate RBF kernels.

3.4 Bayesian least squares support vector machines

Let us recall now some of the contents in Chapter 2. MLPs have become very popular black-box classifiers, however, they suffer from several drawbacks like non-convexity of the underlying optimization problem and difficulties in choosing the best number of hidden units. In Support Vector Machines (SVMs) [113], the learning problem is formulated and represented as a convex Quadratic Programming (QP) problem. In Least Squares Support Vector Machines (LS-SVMs) [94], one uses equality constraints instead of inequality constraints and a sum of squared errors term in order to obtain a linear set of equations in the

dual space.

However, to achieve a high level of performance, some parameters in the LS-SVM model must be tuned. These adjustable hyperparameters include: a regularization parameter, which determines the tradeoff between minimizing the training errors and minimizing the model complexity; and a kernel parameter such as the width of the RBF kernel. One popular way to choose the hyperparameters is cross-validation. Alternatively, one can utilize an upper bound on the generalization error resulting from Vapnik-Chervonenkis (VC) learning theory [113].

In [111], the evidence framework applied to neural networks has been applied to LS-SVMs for classification. Because of the least squares formulation of LS-SVMs, the derivation of analytic expressions on the different levels of inferences is possible. Relating a probabilistic framework to the LS-SVM formulation on the first level of Bayesian inference, the hyperparameters are inferred on the second level. Model comparison is performed on the third level in order to select the kernel parameters.

In the following subsections, we briefly review the use of LS-SVMs in binary classification problems, and how to apply the Bayesian framework to LS-SVM classifiers. For more mathematical details and other applications the interested reader may consult the book [95] and the papers [94, 95, 111, 110]. Then we introduce a LS-SVM input variable selection scheme and sparse approximation procedures for LS-SVM classifiers within the evidence framework.

3.4.1 Probabilistic inferences in LS-SVM within the evidence framework

The LS-SVM classifier $y(\mathbf{x}) = \text{sign}\left[\mathbf{w}^T\varphi(\mathbf{x}) + b\right]$ is inferred from the data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with binary targets $y_i = \pm 1$, by minimizing the following cost function:

$$\min_{\mathbf{w},b,\mathbf{e}} \mathcal{J}_1(\mathbf{w},\mathbf{e}) = \vartheta E_W + \frac{1}{\sigma^2} E_D = \frac{\vartheta}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2\sigma^2}\sum_{i=1}^{N} e_i^2 \qquad (3.25)$$

subject to the equality constraints

$$e_i = 1 - y_i[\mathbf{w}^T\varphi(\mathbf{x}_i) + b], \qquad i = 1, ..., N. \qquad (3.26)$$

The regularization and sum of squared errors term are defined as $E_W = \frac{1}{2}\mathbf{w}^T\mathbf{w}$, and $E_D = \frac{1}{2}\sum_{i=1}^N e_i^2$ respectively. The tradeoff between the training error and regularization is determined by $C = 1/(\vartheta\sigma^2)$.

One defines the Lagrangian

$$\mathcal{L}(\mathbf{w}, b, \mathbf{e}; \boldsymbol{\alpha}) = \mathcal{J}_1 - \sum_{i=1}^{N} \alpha_i \{y_i[\mathbf{w}^T \varphi(x_i) + b] - 1 + e_i\},$$

where α_i are Lagrange multipliers. The Kuhn-Tucker conditions for optimality $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0$, $\frac{\partial \mathcal{L}}{\partial b} = 0$, $\frac{\partial \mathcal{L}}{\partial e_i} = 0$, $\frac{\partial \mathcal{L}}{\partial \alpha_i} = 0$ provide a set of linear equations $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \varphi(\mathbf{x}_i)$, $\sum_{i=1}^{N} \alpha_i y_i = 0$, $\alpha_i = C e_i$, $y_i[\mathbf{w}^T \varphi(\mathbf{x}_i) + b] - 1 + e_i = 0$, for $i = 1, ..., N$, respectively. Elimination of $\mathbf{w}$ and $\mathbf{e}$ gives

$$\left[\begin{array}{c|c} 0 & \mathbf{y}^T \\ \hline \mathbf{y} & \boldsymbol{\Omega} + C^{-1} \boldsymbol{I}_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \boldsymbol{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{1}_v \end{array}\right] \tag{3.27}$$

with $\mathbf{y} = [y_1, \cdots, y_N]^T$, $\alpha = [\alpha_1, \cdots, \alpha_N]^T$, $e = [e_1, \cdots, e_N]^T$, $\mathbf{1}_v = [1, \cdots, 1]^T$, and $\boldsymbol{I}_N$ the $N \times N$ identity matrix. Mercer's theorem is applied to the matrix $\boldsymbol{\Omega}$ with $\Omega_{ij} = y_i y_j \, \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j) = y_i y_j \, k(\mathbf{x}_i, \mathbf{x}_j)$, where $k(\cdot, \cdot)$ is a chosen positive definite kernel that satisfies Mercer condition [64]. The LS-SVM classifier is then constructed in the dual space as:

$$y(\mathbf{x}) = \text{sign}\left[\sum_{i=1}^{N} \alpha_i \, y_i \, k(\mathbf{x}, \mathbf{x}_i) + b\right]. \tag{3.28}$$

It is interesting to notice here that the least squares formulation is related to kernel Fisher Discriminant Analysis. In addition an LS-SVM with linear kernel corresponds to linear Fisher Discriminant Analysis (FDA) with regularization term [111].

Inference of model parameters (level 1)

The parameters $\mathbf{w}$ and bias term b for given values of ϑ, σ^2 are inferred from the data $\mathcal{D}$ at the first level. By applying Bayes' rule, a probabilistic interpretation for (3.25) and (3.26) is obtained:

$$p(\mathbf{w}, b|\mathbf{y}, \vartheta, \sigma^2, \mathcal{H}) = \frac{p(\mathbf{y}|\mathbf{w}, b, \vartheta, \sigma^2, \mathcal{H}) p(\mathbf{w}, b|\vartheta, \sigma^2, \mathcal{H})}{p(\mathbf{y}|\vartheta, \sigma^2, \mathcal{H})}, \tag{3.29}$$

where the model $\mathcal{H}$ corresponds to the kernel function $k(\cdot)$ with different kernel parameters such as the width of an RBF kernel r. The *evidence* $p(\mathbf{y}|\vartheta, \sigma^2, \mathcal{H})$ is a normalizing constant and will be needed in the next level of inference.

The LS-SVM learning process can be given the following probabilistic interpretation. The error function is interpreted as the negative log likelihood for a noise model: $p(\mathbf{y}|\mathbf{w}, b, \sigma^2, \mathcal{H}) \propto \exp(-E_D/\sigma^2)$. Thus the use of the sum of squared errors E_D corresponds to an assumption of Gaussian noise on the target variable defined in (2.1) with variance σ^2.

We assume a separable Gaussian prior on the parameters $\mathbf{w}$, with variance $1/\vartheta$, $p(\mathbf{w}|\vartheta,\mathcal{H}) = (\frac{\vartheta}{2\pi})^{N_\mathcal{F}/2}\exp(-\frac{\vartheta}{2}\mathbf{w}^T\mathbf{w})$, and a Gaussian prior for b with variance $\sigma_b^2 \to \infty$ to approximate a uniform distribution. Thus the regularization term E_W is interpreted in terms of a log prior probability distribution over the parameters $\mathbf{w}$ and b: $p(\mathbf{w},b|\vartheta,\sigma^2,\mathcal{H}) = p(\mathbf{w}|\vartheta,\mathcal{H})p(b|\sigma_b,\mathcal{H}) \propto \exp(-\vartheta E_W)$.

Hence the expression for the first level of inference becomes

$$p(\mathbf{w},b|\mathbf{y},\vartheta,\sigma^2,\mathcal{H}) \propto \exp(-\vartheta E_W)\exp(-E_D/\sigma^2) = \exp(-\mathcal{J}_1(\mathbf{w},b)). \quad (3.30)$$

The maximum a posteriori estimates $\mathbf{w}_{\mathrm{MP}}$ and b_{MP} are then obtained by minimizing the negative logarithm of (3.25), i.e. solving the linear set of equations in (3.27).

Class probabilities for the LS-SVM classifiers (level 1)

Given the posterior probability of the model parameters $\mathbf{w}$ and b we will now integrate over all $\mathbf{w}$ and b values so as to obtain the posterior probability $p(y|\mathbf{x},D,\vartheta,\sigma^2,\mathcal{H})$.

In the evidence framework, we assume that the posterior distribution of $\mathbf{w}$ can be approximated by a single Gaussian at $\mathbf{w}_{\mathrm{MP}}$. We define two error variables corresponding to different classes (indicated by subscripts '+' and '−') as $\mathbf{e}_\pm = \mathbf{w}^T(\varphi(\mathbf{x}) - \hat{\mathbf{m}}_\pm)$, where $\hat{\mathbf{m}}_+$ and $\hat{\mathbf{m}}_-$ are the centers of the positive and negative class respectively. After marginalizing over $\mathbf{w}$ the distribution of $\mathbf{e}_\pm$ will also be Gaussian, centering around mean $\mathbf{m}_{\mathbf{e}_\pm}$ with variance $(\sigma_\pm^2 + \sigma_{\mathbf{e}\pm}^2)$. The expression for the mean is

$$\begin{aligned} \mathbf{m}_{e_\pm} &= \mathbf{w}_{\mathrm{MP}}^T(\varphi(\mathbf{x}) - \hat{\mathbf{m}}_\pm) \\ &= \sum_{i=1}^N \alpha_i y_i k(\mathbf{x},\mathbf{x}_i) - \frac{1}{N_\pm}\sum_{i=1}^N \alpha_i y_i \sum_{j\in \boldsymbol{I}_\pm} k(\mathbf{x}_i,\mathbf{x}_j), \end{aligned} \quad (3.31)$$

where $\boldsymbol{I}_+$ and $\boldsymbol{I}_-$ indicate the sets of indices whose corresponding data points have positive and negative labels respectively. The computation of the variance from the target noise $\sigma_\pm^2$ will be discussed in the next section. While the corresponding expression of the additional variance due to the uncertainty in the parameters $\mathbf{w}$ is

$$\sigma_{\mathbf{e}_\pm}^2 = [\varphi(\mathbf{x}) - \hat{\mathbf{m}}_\pm]^T\mathbf{Q}_{11}[\varphi(\mathbf{x}) - \hat{\mathbf{m}}_\pm], \quad (3.32)$$

where $\mathbf{Q}_{11}$ is the upper left $N_\mathcal{F} \times N_\mathcal{F}$ block of the covariance matrix $\mathbf{Q} = \mathrm{covar}([\mathbf{w};b],[\mathbf{w};b])$, which is related to the Hessian $\mathbf{H}$ of the LS-SVM cost function $\mathcal{J}_1(\mathbf{w},b)$,

$$\mathbf{Q} = \mathbf{H}^{-1} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{\partial^2 J_1}{\partial \mathbf{w}^2} & \frac{\partial^2 J_1}{\partial \mathbf{w}\partial b} \\ \frac{\partial^2 J_1}{\partial b\partial \mathbf{w}} & \frac{\partial^2 J_1}{\partial b^2} \end{bmatrix}^{-1}. \quad (3.33)$$

And the variance will be finally computed in the dual space. Let $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x},\mathbf{x}_1)\cdots k(\mathbf{x},\mathbf{x}_N)]^T$, $\mathbf{K}$ be the $N \times N$ kernel matrix with element $K_{ij} = k(x_i,x_j)$, and a centering matrix $\mathbf{K}_c = (\boldsymbol{I}_N - \frac{1}{N}\mathbf{1}_v\mathbf{1}_v^T)$ with $\mathbf{1}_v = [1,\cdots,1]^T$. Further define $\mathbf{1}_+,\mathbf{1}_- \in \mathbb{R}^N$ as the vector with element zero or one, for $i = 1,...,N$, $\mathbf{1}_{\pm,i} = 1$ if $y_i = \pm 1$, otherwise $\mathbf{1}_{\pm,i} = 0$. By using matrix algebra and applying Mercer's condition, we obtain:

$$\begin{aligned}\sigma^2_{e\pm} &= \frac{1}{\vartheta}k(\mathbf{x},\mathbf{x}) - \frac{2}{\vartheta N_\pm}\sum_{i\in\boldsymbol{I}_\pm} k(\mathbf{x},\mathbf{x}_i) + \frac{1}{\vartheta N_\pm^2}\sum_{i,j\in\boldsymbol{I}_\pm} k(\mathbf{x}_i,\mathbf{x}_j) - \\ &\frac{1}{\vartheta\sigma^2}(\mathbf{k}^T(\mathbf{x}) - \frac{1}{N_\pm}\mathbf{1}_\pm^T\mathbf{K})\mathbf{K}_c(\vartheta\boldsymbol{I}_N + \sigma^{-2}\mathbf{K}_c\mathbf{K}\mathbf{K}_c)^{-1}\mathbf{K}_c(\mathbf{k}(\mathbf{x}) - \frac{1}{N_\pm}\mathbf{K}\mathbf{1}_\pm).\end{aligned} \tag{3.34}$$

Thus the conditional probability densities given an input $\mathbf{x}_*$ can be computed as:

$$p(\mathbf{x}_*|y_* = \pm 1,\mathbf{y},\vartheta,\sigma^2,\sigma^2_\pm,\mathcal{H}) = (2\pi(\sigma^2_\pm + \sigma^2_{\mathbf{e}_\pm}))^{\frac{-1}{2}}\exp(-\frac{\mathbf{m}^2_{\mathbf{e}_\pm}}{2(\sigma^2_\pm + \sigma^2_{\mathbf{e}_\pm})}). \tag{3.35}$$

By applying Bayes' rule the following posterior class probabilities of the LS-SVM classifier are obtained (for notational simplicity, $\vartheta,\sigma^2,\sigma^2_\pm,\mathcal{H}$ are dropped in this expression):

$$p(y_*|\mathbf{x}_*,\mathbf{y}) = \frac{p(y_*)p(\mathbf{x}_*|y_*,\mathbf{y})}{P(y_*=1)p(\mathbf{x}_*|y_*=1,\mathbf{y}) + P(y_*=-1)p(\mathbf{x}_*|y_*=-1,\mathbf{y})}, \tag{3.36}$$

where $p(y)$ corresponds to the prior class probability. The posterior probability could also be used to make minimum risk decisions in case of different error costs. Let c^+_- and c^-_+ denote the cost of misclassifying a case from class '$-$' and '$+$' respectively. One trick to combine the posterior probability with the different error costs is by replacing $p(y)$ in (3.36) with the adjusted class prior:

$$P'(y{=}1) = P(y{=}1)c^-_+/(P(y{=}1)c^-_+ + P(y{=}-1)c^+_-),$$

and

$$P'(y{=}-1) = P(y{=}-1)c^+_-/(P(y{=}1)c^-_+ + P(y{=}-1)c^+_-).$$

Inference of hyperparameters (level 2)

The second level of inference via Bayes' rule is the following:

$$\begin{aligned}p(\vartheta,\sigma^2|\mathbf{y},\mathcal{H}) &= \frac{p(\mathbf{y}|\vartheta,\sigma^2,\mathcal{H})p(\vartheta,\sigma^2|\mathcal{H})}{p(\mathbf{y}|\mathcal{H})} \\ &\propto p(\mathbf{y}|\vartheta,\sigma^2,\mathcal{H}),\end{aligned} \tag{3.37}$$

where a uniform distribution is assumed in ϑ and σ^{-2} for the prior $p(\vartheta, \sigma^2|\mathcal{H}) = p(\vartheta|\mathcal{H})p(\sigma^2|\mathcal{H})$. The probability $p(\mathbf{y}|\vartheta, \sigma^2, \mathcal{H})$ is equal to the evidence of the previous level. Using Gaussian density at the maximum a posteriori estimates $\mathbf{w}_{\mathrm{MP}}, b_{\mathrm{MP}}$, we obtain:

$$p(\vartheta, \sigma^2|\mathbf{y}, \mathcal{H}) \propto \sqrt{\frac{\vartheta^{N_{\mathcal{F}}}}{\sigma^{2N} \det \mathbf{H}}} \exp(-\mathcal{J}_1(\mathbf{w}_{\mathrm{MP}}, b_{\mathrm{MP}})), \tag{3.38}$$

with the Hessian $\mathbf{H}$ defined in (3.33). The expression for $\det \mathbf{H}$ is given by $N\vartheta^{N_{\mathcal{F}}-N_{\mathrm{eff}}}\sigma^{-2}\prod_{i=1}^{N_{\mathrm{eff}}}(\vartheta + \sigma^{-2}\lambda_{G,i})$, where N_{eff} eigenvalues $\lambda_{G,i}$ are the non-zero eigenvalues of the centered kernel matrix in the feature space and are the solution of the eigenvalue problem

$$(\mathbf{K}_{\mathrm{c}}\mathbf{K}\mathbf{K}_{\mathrm{c}})\boldsymbol{\nu}_{G,i} = \lambda_{G,i}\boldsymbol{\nu}_{G,i}, \quad i = 1, ..., N_{\mathrm{eff}} \leq N - 1. \tag{3.39}$$

The effective number of parameters [10, 61] for LS-SVM, is equal to:

$$\gamma_{\mathrm{eff}} = 1 + \sum_{i=1}^{N_{\mathrm{eff}}} \frac{\lambda_{G,i}}{\vartheta_{\mathrm{MP}}\sigma^2_{\mathrm{MP}} + \lambda_{G,i}} = 1 + \sum_{i=1}^{N_{\mathrm{eff}}} \frac{C_{\mathrm{MP}}\lambda_{G,i}}{1 + C_{\mathrm{MP}}\lambda_{G,i}}, \tag{3.40}$$

where the first term is due to the fact that no regularization is applied on the bias term b of the LS-SVM model. Since $N_{\mathrm{eff}} \leq N - 1$, the estimated number of effective parameters cannot exceed the number of data points N.

In the optimum of the level 2 cost function, the following relations can be obtained: $2\vartheta_{\mathrm{MP}}E_W(w_{\mathrm{MP}}) = \gamma_{\mathrm{eff}} - 1$ and $2E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})/\sigma^2_{\mathrm{MP}} = N - \gamma_{\mathrm{eff}}$. The last equality can be viewed as the Bayesian estimate of the variance $\sigma^2_{\mathrm{MP}} = \sum_{i=1}^{N} e_i^2/(N - \gamma_{\mathrm{eff}})$ of the noise e_i. However in this thesis, when computing the posterior of the class probability, the variances of the noise with different classes may differ, and are approximated in this way:

$$\sigma^2_{\pm} = \textstyle\sum_{j \in \boldsymbol{I}_{\pm}} e^2_{\pm,j}/(N_{\pm} - \gamma_{\mathrm{eff}}\frac{N_{\pm}}{N}). \tag{3.41}$$

In practice, one can reformulate the optimization problem in ϑ and σ^{-2} into a scalar optimization problem in $C = 1/(\vartheta\sigma^2)$:

$$\min_C \mathcal{J}_2(C) = \sum_{i=1}^{N-1} \log[\lambda_{G,i} + \frac{1}{C}] + (N-1)\log[E_W(w_{\mathrm{MP}}) + CE_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})], \tag{3.42}$$

with $\lambda_{G,i} = 0$ for $i > N_{\mathrm{eff}}$. The expressions for E_D and E_W can be given in the dual variables using the relation $\alpha_i = Ce_i$: $E_D = \frac{1}{2C^2}\sum_{i=1}^{N}\alpha_i^2$, $E_W = \frac{1}{2}\sum_{i=1}^{N}\alpha_i(y_i - \frac{\alpha_i}{C} - b_{\mathrm{MP}})$.

This optimal hyperparameter C is then obtained by solving the optimization problem (3.42) with gradients. Given the optimal C_{MP}, one can easily compute ϑ_{MP} and σ^2_{MP} using their relations in the optimum:

$$C = \frac{N - \gamma_{\text{eff}}(C)}{\gamma_{\text{eff}}(C) - 1} \frac{E_W(\mathbf{w}_{\text{MP}}; C)}{E_D(\mathbf{w}_{\text{MP}}, b_{\text{MP}}; C)} \tag{3.43}$$

with

$$\begin{aligned} E_D(\mathbf{w}_{\text{MP}}, b_{\text{MP}}) &= \frac{1}{2C^2}\mathbf{y}^T\mathbf{K}_{\text{c}}\mathbf{V}_G(\mathbf{\Lambda}_G + \boldsymbol{I}/C)^{-2}\mathbf{V}_G^T\mathbf{K}_{\text{c}}\mathbf{y} \\ E_W(\mathbf{w}_{\text{MP}}) &= \frac{1}{2}\mathbf{y}^T\mathbf{K}_{\text{c}}\mathbf{V}_G\mathbf{\Lambda}_G(\mathbf{\Lambda}_G + \boldsymbol{I}/C)^{-2}V_G^T\mathbf{K}_{\text{c}}\mathbf{y} \\ E_W(\mathbf{w}_{\text{MP}}) + CE_D(\mathbf{w}_{\text{MP}}, b_{\text{MP}}) &= \frac{1}{2}\mathbf{y}^T\mathbf{K}_{\text{c}}\mathbf{V}_G(\mathbf{\Lambda}_G + \boldsymbol{I}/C)^{-1}V_G^T\mathbf{K}_{\text{c}}\mathbf{y}, \end{aligned} \tag{3.44}$$

and $\mathbf{V}_G = (\boldsymbol{\nu}_{G,1}, \ldots, \boldsymbol{\nu}_{G,N_{\text{eff}}}) \in \mathbb{R}^{N \times N_{\text{eff}}}$, $\mathbf{\Lambda}_G = \text{diag}(\lambda_{G,1}, \cdots, \lambda_{G,N_{\text{eff}}})$.

Bayesian model comparison (level 3)

After determining the hyperparameter ϑ_{MP} and σ^2_{MP} on the second level of inference, we still have to select a suitable model $\mathcal{H}_j$. The prior $p(\mathcal{H}_j)$ over all possible models is assumed to be uniform. Thus the posterior for the model $\mathcal{H}_j$ is in the form of $p(\mathcal{H}_j|\mathbf{y}) \propto p(\mathbf{y}|\mathcal{H}_j)p(\mathcal{H}_j) \propto p(\mathbf{y}|\mathcal{H}_j)$. At this level, no evidence or normalizing constant is used since it is infeasible to compare all possible models $\mathcal{H}_j$.

A separable Gaussian prior for $p(\log\vartheta_{\text{MP}}, \log\sigma^2_{\text{MP}}|\mathcal{H}_j)$ is assumed for all models $\mathcal{H}_j$, with the constant standard deviations $\sigma_{\log\vartheta}$ and $\sigma_{\log\sigma^{-2}}$. These prior widths of the hyperparameters are generally assumed to be broad and they cancel out when alternative models are compared. Also we assume that $p(\log\vartheta, \log\sigma^2|\mathbf{y}, \mathcal{H}_j)$ can be well approximated by using a separable Gaussian with error bars $\sigma_{\log\vartheta|\mathcal{D}}$ and $\sigma_{\log\sigma^{-2}|\mathcal{D}}$. The posterior likelihood $p(\mathbf{y}|\mathcal{H}_j)$ corresponds to the evidence at the previous level and can be evaluated by:

$$p(\mathbf{y}|\mathcal{H}_j) \propto p(\mathbf{y}|\vartheta_{\text{MP}}, \sigma^2_{\text{MP}}, \mathcal{H}_j)\frac{\sigma_{\log\vartheta|\mathcal{D}}\sigma_{\log\sigma^{-2}|\mathcal{D}}}{\sigma_{\log\vartheta}\sigma_{\log\sigma^{-2}}}. \tag{3.45}$$

The models can thus be ranked according to the evidence $p(\mathbf{y}|\mathcal{H}_j)$, that is the tradeoff between the goodness of fit from the previous level $p(\mathbf{y}|\vartheta_{\text{MP}}, \sigma^2_{\text{MP}}, \mathcal{H}_j)$ and the Occam factor $\frac{\sigma_{\log\vartheta|\mathcal{D}}\sigma_{\log\sigma^{-2}|\mathcal{D}}}{\sigma_{\log\vartheta}\sigma_{\log\sigma^{-2}}}$ [61].

The error bars of $p(\mathbf{y}|\log\vartheta_{\text{MP}}, \sigma^2_{\text{MP}}, \mathcal{H}_j)$ can be approximated by $\sigma^2_{\log\vartheta|\mathcal{D}} \simeq \frac{2}{\gamma_{\text{eff}}-1}$ and $\sigma^2_{\log\sigma^{-2}|\mathcal{D}} \simeq \frac{2}{N-\gamma_{\text{eff}}}$. And the expression for the evidence in the dual space is the following:

$$p(\mathbf{y}|\mathcal{H}_j) \propto \sqrt{\frac{\vartheta_{\text{MP}}^{N_{\text{eff}}}\sigma_{\text{MP}}^{-2(N-1)}}{(\gamma_{\text{eff}}-1)(N-\gamma_{\text{eff}})\prod_{i=1}^{N_{\text{eff}}}(\vartheta_{\text{MP}} + \sigma^{-2}_{\text{MP}}\lambda_{G,i})}}. \tag{3.46}$$

One selects the kernel parameters, e.g. the width of an RBF kernel, with maximal posterior $p(\mathbf{y}|\mathcal{H}_j)$. In the thesis we use a simple coarse to fine search algorithm to find the optimal kernel parameter. The optimization of regularization hyperparameters have already been embedded in the Bayesian learning algorithm as described above.

3.4.2 Design of the LS-SVM classifier in a Bayesian evidence framework

Before building an LS-SVM classifier, it is usually better to normalize componentwise the training inputs to zero mean and unit variance [10]. We denote the normalized training data as $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, with $\mathbf{x}_i$ the normalized inputs and $y_i \in \{-1, 1\}$ the corresponding class label. The new inputs collected in the test set and for evaluating the trained model will also be normalized in the same way as the training data, i.e. using the mean and variance estimates from the training data. Now, we start the design of the LS-SVM classifier in a Bayesian framework. Several procedures including hyperparameter tuning, input variable selection and sparse approximation are to be established.

Hyperparameter tuning

Select the model $\mathcal{H}_j$ by choosing a kernel type k_j with possible kernel parameters, e.g., the width of an RBF kernel r_j. Infer the optimal C_{MP}, ϑ_{MP} and σ^2_{MP} on level 2 inference and evaluate the model evidence as follows:

1. Solve the eigenvalue problem (3.39).
2. Solve the scalar optimization problem (3.42) in $C = 1/(\vartheta\sigma^2)$ using, e.g., a quasi-Newton method.
3. Given the optimal C_{MP}, compute ϑ_{MP} and σ^2_{MP} and γ_{eff}.
4. Calculate $p(\mathbf{y}|\mathcal{H}_j)$ from (3.46) at the third level.

For a kernel k_j with tuning parameters, refine the tuning parameters, such that a higher model evidence $p(\mathbf{y}|\mathcal{H}_j)$ is obtained. For example, for an RBF kernel, the parameter r is inferred on the third level.

Input variable selection

In the context of the Bayesian evidence framework, the evidence of the model $p(\mathbf{y}|\mathcal{H}_j)$ is computed with (3.46) on level 3 inference. A higher $p(\mathbf{y}|\mathcal{H}_1)$ compared to $p(\mathbf{y}|\mathcal{H}_0)$ means the data favor $\mathcal{H}_1$ to $\mathcal{H}_0$. Therefore, given a certain

type of kernel for the model, we propose to select the input variables according to the model evidence $p(\mathbf{y}|\mathcal{H}_j)$.

The procedure performs a forward selection (greedy search), starting from zero variables, and choosing each time the variable which gives the greatest increase in the current model evidence. The selection is stopped when the addition of any remaining variable no longer increases the model evidence.

Sparse approximation

Due to the choice of the 2-norm in the cost function, the sparseness is lost compared with the standard QP type SVMs. However, as has been shown in [95], the sparseness can be imposed to LS-SVMs by a pruning procedure based upon the sorted support value spectrum $|\alpha_i|$. Inspired by the SVM solution whose support vectors are near the decision boundary, we propose here to prune the data points which have negative support values. This is quite intuitive, since in LS-SVMs, $\alpha_i = Ce_i$. Negative support value α_i indicate that the data (x_i, y_i) are easy cases. The pruning of easy examples will focus the model more on the harder cases which lie around the decision boundary.

1. $D_{\text{cur}} = D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$.
2. Based on D_{cur}, select the regularization parameter C and possibly a kernel parameter r within the Bayesian evidence framework. Train the LS-SVM (compute α) on the data D_{cur}, using current C and r.
3. If all the support values are positive, then go to 6.
4. Repeat pruning all the data points with non-positive support values, $D_{\text{cur}} \Leftarrow D_{\text{cur}} \setminus \{(\mathbf{x}_d, y_d)|\alpha_d \leq 0\}$. Based on the reduced data set D_{cur}, recompute α using the same C and the kernel parameter, until all α values on the reduced data set D_{cur} are positive.
5. go to 2.
6. stop pruning, return the current α value and set the support values for the pruned data to zero.

Further issues related to the Bayesian sparse LS-SVM classifier are as follows:

- Variance computation. The variance resulted from parameter estimation can be computed using only the support vectors, and this estimate serves as a upper bound for the variance. Intuitively this is easy to understand. Support vectors represent the data points that are difficult to classify. Variance estimation using only support vectors thus usually provides a higher value than it should be for the data points close to the "easy"

cases (i.e. the non-support vectors). Therefore, in order to get a more informative variance estimate, we in general compute the variance by taking all the training data into account.

- Model evidence computation. In order to compare the model evidence for different models, which use different support vectors or different kernels, the evidence should be calculated based on all the training data. Comparison based only on the partial data is unfair for those who use less support vectors. The evidence, the posterior probability of the sparse LS-SVM model can be approximated then by the evidence of the non-sparse LS-SVM model which use the same type of kernels and the same kernel parameters, e.g. the width of the RBF kernels.

Probabilistic interpretation of the output

The designed LS-SVM classifier $\mathcal{H}_j$ can be used to calculate class probabilities in the following steps:

1. Given the parameters $\alpha, b_{\mathrm{MP}}, \vartheta_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}}, C_{\mathrm{MP}}, \gamma_{\mathrm{eff}}$ and the eigenvalues and eigenvectors in (3.39) available from the designed model $\mathcal{H}_j$, calculate $\mathbf{m}_{e+}, \mathbf{m}_{e-}, \sigma^2_{e+}$ and σ^2_{e-} from (3.31) and (3.34) respectively. Compute σ^2_+ and σ^2_- from (3.41).
2. Calculate $p(\mathbf{x}|y = \pm 1, D, \vartheta, \sigma^2, \sigma^2_\pm, \mathcal{H}_j)$ from (3.35).
3. Calculate $p(y|\mathbf{x}, D, \mathcal{H}_j)$ from (3.36) by using the prior class probabilities or adjusted priors $P'(y = +1)$ and $P'(y = -1)$, respectively.

3.4.3 Toy examples on Bayesian LS-SVM classifiers

We will use again Ripley's synthetic data set and a toy binary classification data set to illustrate how Bayesian LS-SVM and its sparse version can work for classification problems by use of the aforementioned algorithm design.

First, let us take a look at an example on how we select the kernel width by a coarse-to-fine searching. Depicted in Figure 3.3 is the curve of width r for RBF kernels vs. the model evidence for Bayesian LS-SVM when trained on the Ripley's synthetic data set. The searching algorithm locates a local optimum very quickly just within four runs of model evidence calculation (each run with 4 or 5 candidate width values for the RBF kernel). By use of the automatically selected kernel width, the Bayesian LS-SVM classifier obtains a performance illustrated in Figure 3.4, with an error rate on the test set of 9.4% and an AUC of 0.967.

Now take as example a linearly separable classification problem. We show how a sparse Bayesian LS-SVM classifier is mimicking the support vector machine

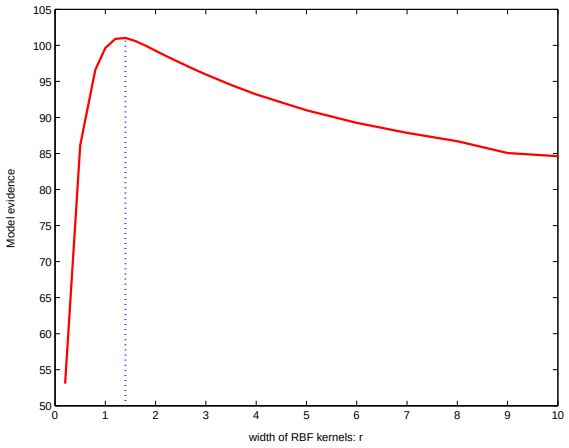


Figure 3.3: Width of RBF kernel r vs. model evidence. The optimal kernel parameter r is chosen as the one which maximizes the model evidence. The curve is obtained from the training process of Bayesian LS-SVM for classification of Ripley's synthetic data set.

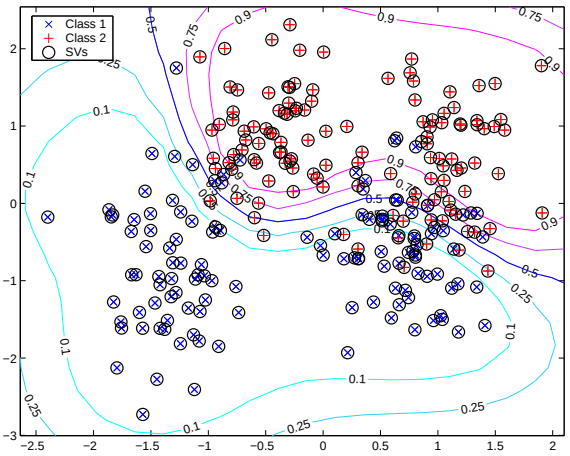


Figure 3.4: Classification of Ripley's synthetic data set with Bayesian LS-SVM classifier. The contours for the posterior probabilities for a point belonging to class 2 have been shown. The training data are all support vectors, which have been circled. The classifier has an error rate of 9.4% on the 1000 test data when taking 0.5 as the probability cutoff, and the area under the ROC curve is 0.967 (SE=0.006).

in the sense that the support vectors are located close to the decision boundary. Figures 3.5 and 3.6 plot the decision boundaries (i.e. the contours with pos-

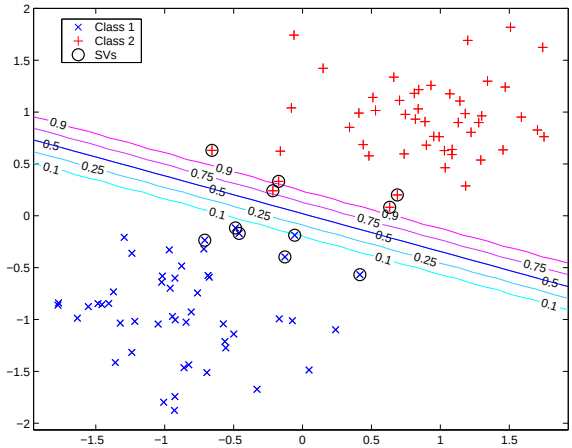


Figure 3.5: Classification on a two-class linearly separable data set using sparse Bayesian LS-SVM classifiers with linear kernel. The contours for the posterior probabilities for a point belonging to class 2 have been shown. Only 11 out of the 100 training data serve as support vectors (training data with a positive support value).

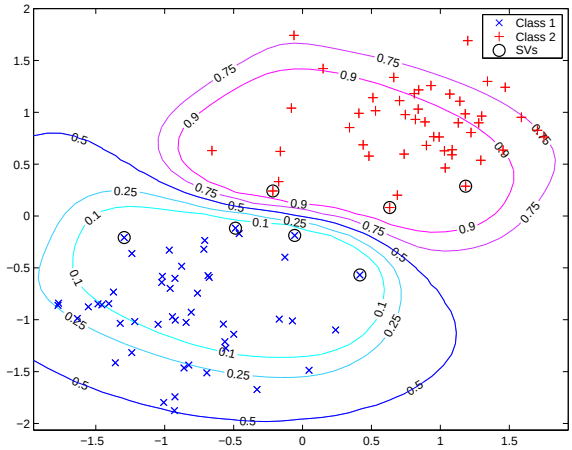


Figure 3.6: Classification on a two-class linearly separable data set using sparse Bayesian LS-SVM classifier with RBF kernel. The contours for the posterior probabilities for a point belonging to class 2 have been shown. Only 7 out of the 100 training data serve as support vectors (training data with a positive support value).

terior probability at the level of 0.5) created by two sparse Bayesian LS-SVM classifiers with linear kernels and RBF kernels, respectively, and the support vectors which have support value > 0 have been circled.

It is worth examining the way in which the variances resulted from parameter estimation are computed for the sparse Bayesian LS-SVM. In general the variance computation is related to all the training data for a non-sparse Bayesian LS-SVM. However, in sparse LS-SVMs, after pruning some data points which are considered to be "easy" to classify, only difficult cases close to the decision boundaries remain in the models. If we compute variance using only these support vectors, a data point being distant from the support vectors will be associated with a rather large variance, even though this data point is close to easily classified cases. This can be observed in Figure 3.7, where the parameter variance is calculated based on only the support vectors. Actually variance computed using only the support vectors can be deemed as a upper bound for the parameter variance. While the variance computed from the whole training data set looks much more reasonable (see Figure 3.8), and can be considered as a good approximation of the variance (or as a lower bound). Nevertheless a more strict theoretical treatment of the variance computation is still needed. Furthermore, in case of large sample size in the training data set, the computation using all the training data is also unfavored from the computational point of view.

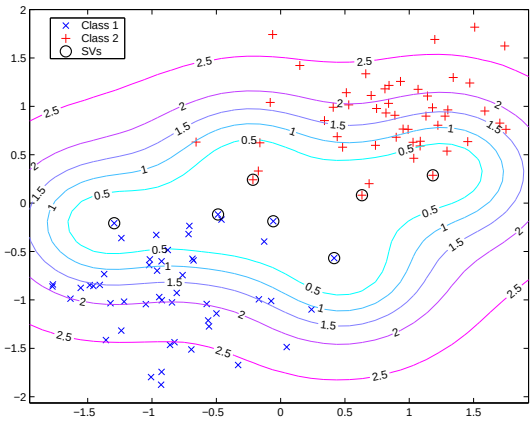


Figure 3.7: The point-wise latent output variance (represented in contours) due to parameter estimation for classification of a linearly separable 2-class data set with sparse Bayesian LS-SVM classifiers with RBF kernels. In this case, only support vectors (circled) were used to compute the variances due to the uncertainty in parameter estimation. Therefore points further away from the support vectors are associated with higher variances.

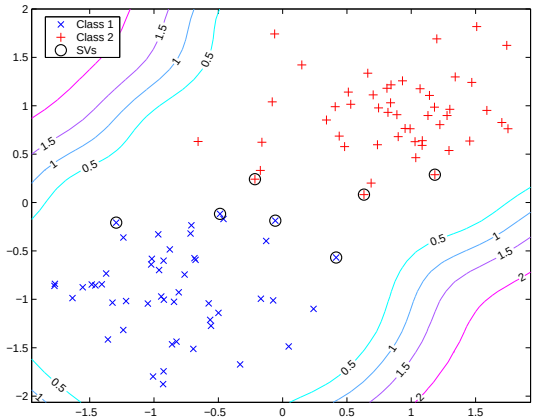


Figure 3.8: The point-wise latent output variance (represented in contours) due to parameter estimation for classification of a linearly separable 2-class data set with sparse Bayesian LS-SVM classifiers with RBF kernels. The original data have also been plotted. In this case, all the training data have been used to compute the variances due to the uncertainty in parameter estimation. Hence the points further away from the training data points are associated with higher variances.

We also check how the Bayesian sparse LS-SVM classifiers perform on Ripley's synthetic data set. Figures 3.9 and 3.10 depict contours for the posterior probabilities computed from the sparse Bayesian LS-SVM classifier using multivariate RBF kernels and additive RBF kernels, respectively. Again similar performance can be achieved by using univariate RBF kernels as multivariate RBF kernels.

Finally, we show how we can gain some insights to the LS-SVM classifiers by means of kernel principal regression. As explained in Section 2.4.6, the latent outcome of LS-SVM can be equivalent to the one obtained from the ridge regression model based on the data projections on all the principal components in the feature space. In order to see roughly how the LS-SVM classifier make its decisions in the the high dimensional feature space, the 2-dimensional training data (marked in '+' and '×') are projected to pairs of PCs in the kernel-inducde feature space, as depicted in Figure 4.18. The approximated decision boundary (solid line) using the ith and jth PCs is obtained from the equation $\tilde{w}_i z_i + \tilde{w}_j z_j + \tilde{b} = 0$. Note that, in the case of sparse LS-SVM classifiers, only the support vectors in the training data are included in the final sparse model fitting. Therefore, only the support vectors (marked with circles) will be used for for performing kPCR. Furthermore, we can superimpose the explanatory variables on the graph, forming into some kind of nonlinear biplot [35]. For the kth variable, pseudosamples can be generated by setting e.g. the data mean

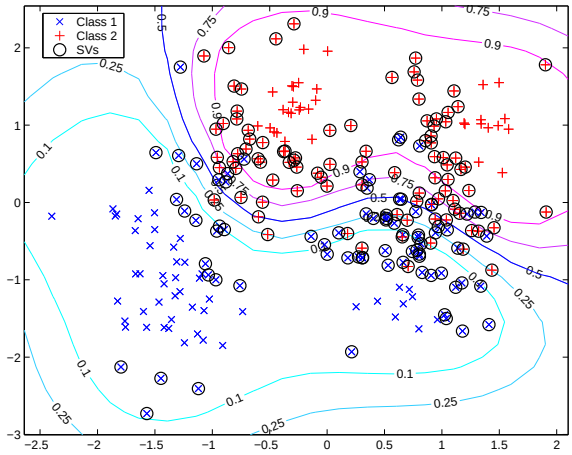


Figure 3.9: Classification of Ripley's synthetic data set with sparse Bayesian LS-SVM classifiers. The contours for the posterior probabilities for a point belonging to class 2 have been shown. There are 168 (67%) support vectors with support value greater than 0, which have been circled. The classifier has an error rate of 9.3% on the 1000 test data, and the area under the ROC curve is 0.968 (SE=0.006).

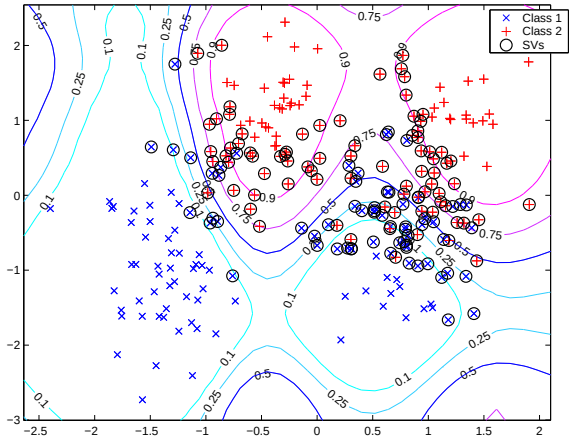


Figure 3.10: Classification of Ripley's synthetic data set with sparse Bayesian LS-SVM classifiers. The contours for the posterior probabilities for a point belonging to class 2 have been shown. There are 143 (57%) support vectors with support value greater than 0, which have been circled. When the cutoff probability is taken as 0.5, the classifier has the same error rate of 9.3% on the 1000 test data as in Figure 3.9, but the area under the ROC curve is slightly lower with value is 0.967 (SE=0.006).

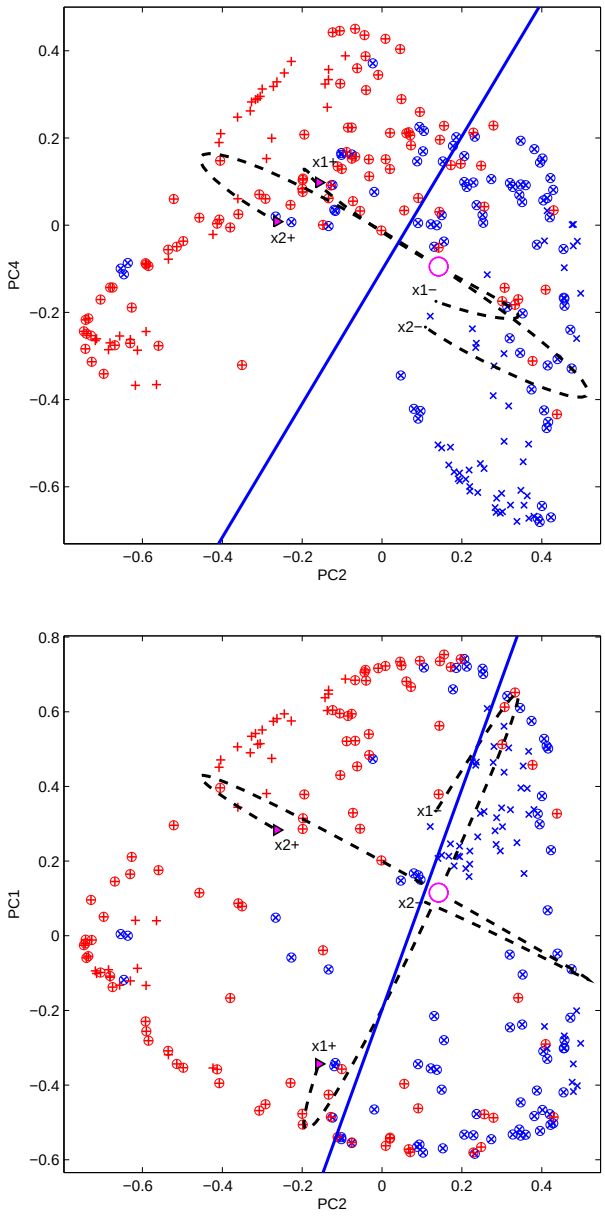


Figure 3.11: Visualization of LS-SVM with RBF kernels via kPCR on classification of Ripley's synthetic data. See Figure 3.9 for the decision boundary in the original space.

as the starting point, varying gradually the value of kth variable while fixing the others; then a trajectory can be obtained by tracing the projection of the pseudosamples onto the pairs of PCs in the feature space. In case of linear kernels, those trajectories can be called variable axis.

The dashed line shows the trajectories of the variables, concurrent at the center of the data (denoted as a circle); the pseudosample is bounded by the minimum and the maximum value of the corresponding variable in the training set (the positive end of the trajectory corresponds to the maximum value of the input variable, and is indicated by a triangle). Interpretation can be done through observing the class distributions, the length of trajectories, and the correlations between the trajectories and classes.

In this example, we only plotted 3 PCs (PC2, PC4 and PC1) which have the largest correlation with the target values, and their eigenvalues are 20.5, 8.9, and 34.8, respectively. As a kind of unsupervised learning technique, the first PCs yielded from kernel PCA does not necessarily have the largest correlations with the target value. The plots of other PCs appeared to be less important for discrimination between the two classes and thus could be ignored for purpose of interpretation. Here PC2 is the most discriminative PC. Input variable x1 and x2 are both important because their trajectories are both almost orthogonal to the approximate decision line in the plot of PC2 vs PC4 (upper biplot), although x1 is is not so well expressed in this plot due to the shorter trajectory compared to the one of x2. And some nonlinearity behaviors appear when the two inputs are close to their maximum or the minimum values. In the plot of PC2 vs PC1 (bottom biplot), we could see that the trajectory of x1 is almost parallel to the partial decision line, and thus less important compared with x2.

We could see such plot could be helpful in understanding the decision of the LS-SVM classifiers. However, one should also be aware that, generating the variable trajectories is a sort of sensitivity analysis. It should be more helpful to see different trajectories starting from different points rather than only the projections of the medians. Hence a computer-based interactive program would be of great help in order to demonstrate the sensitivity analysis on different starting points dynamically.

3.5 Summary

This chapter has addressed the issues of controlling the hyperparameters in learning algorithms for blackbox models, including MLP, RBF-NN, LS-SVMs. By constructing posterior probability distributions for the parameters and hyperparameters in the models, several important steps in probabilistic modelling can be conducted in a unified way, for example, tuning of the regularization parameters, automatic selection of the kernel parameters in case of kernel-based

methods, input variable selection, as well as accounting for the uncertainty in predictions.

We focused on Bayesian kernel-based models instead of MLPs because kernel-based models have the same learning power as MLP, while less nuisance parameters need to be tuned and with the capability to deal with high dimensional data sets. Sparse Bayesian learning (SBL) models employ ARD methods in models linear in parameters, which usually forcing many parameters equal to zero. The use of SVM-like kernels in SBL models results into the so-called relevance vector machine, which can be viewed as a type of Bayesian RBF-NN with the basis parameters directly related to the original training data points. However, SBL models still have to face the problem of local minima and multiple solutions as for Bayesian MLP, and thus the comparison of the model evidence with different basis function might be unstable. On the contrary, least squares support vector machines solutions can be obtained from a set of linear equations, and have the primal-dual interpretation. Analytical expressions in parameter posterior and prediction distribution can be obtained when integrating with Bayesian learning methods. Also since Bayesian LS-SVM classifiers are intrinsically using a regression framework, no iterative algorithms are involved for non-sparse Bayesian LS-SVM classifiers. For RVM classifiers, the logistic link function is usually applied, which necessitates the iterative algorithm for solving the classification problems. In order to improve the generalization performance for LS-SVM, sparseness can be imposed to the LS-SVM solution by iteratively pruning the support vectors (with non positive support values). It is almost (hyper)parameter free for the whole learning procedure provided the type of kernel is given. Illustrative examples also demonstrate that the automatic Bayesian learning algorithms for kernel-based classification are very attractive in practice, without the burden of using time-consuming and often variable cross-validation estimates.

Chapter 4

Linear and nonlinear models for preoperative classification of ovarian tumors

This chapter reports the main application of this thesis on development and evaluation of probabilistic models for preoperative prediction of the malignancy of ovarian tumors. The study consists of two consecutive stages involving first a pilot project conducted in Leuven, and then the extended project on an international multi-center study on ovarian tumor analysis (IOTA). We initially developed and evaluated mathematical models for ovarian tumor classification based on the single-center data set from the pilot project. Carried with the experiences from the pilot project we proceeded with another round of model development and validation using multi-center data set. The general analysis includes exploratory data analysis, input variable selection, parameter estimation, model interpretation and performance evaluation via Receiver Operating Characteristic (ROC) curve analysis. The performance of various linear and nonlinear models have been evaluated and compared, particularly for logistic regression, Bayesian multilayer perceptrons (MLPs), relevance vector machines (RVMs), and Bayesian least squares support vector machines (LS-SVMs). The experimental results show that the probabilistic models have the potential to obtain a reliable preoperative distinction between malignant and benign ovarian tumors, and to assist the clinician in making a correct diagnosis.

4.1 Introduction

Adnexal masses are a very common problem in gynecology, and represent a number of different benign and malignant lesions or masses. Benign lesions can be either functional or non-functional. Malignant tumors are cancerous, including borderline, primary, and metastatic lesions. Detection of ovarian malignancy at an early stage is very important for the survival of the patients. The 5-year survival rate for ovarian cancer when detecting at a late clinical stage is 35% [76]. In contrast, the 5-year survival for patients with stage I ovarian cancer is about 80% [114]. However, nowadays 75% of the cases are only diagnosed at an advanced stage, resulting into the highest mortality rate among gynecologic cancers. The treatment and management of different types of ovarian tumors differ greatly. Conservative management or less invasive surgery suffices for patients with a benign tumor; on the other hand, those with suspected malignancy should be timely referred to a gynecologic oncologist. A reliable preoperative discrimination between malignant and benign ovarian tumors is thus critical for obtaining the most effective treatment and best advice, and will influence the outcome for the patient and the medical costs.

In practice, clinicians have a wide range of diagnostic techniques for the preoperative evaluation of adnexal masses, such as physical examination, serum tumor markers, gray scale and color Doppler ultrasonography, axial computed tomography, and magnetic resonance imaging [100]. However, some of them are of limited value, such as physical examination and tumor markers; others are expensive, such as axial computed tomography and magnetic resonance imaging; and some others are controversial, such as color Doppler ultrasonography. Recently, new approaches based on mathematical models have been proposed [104][6][101][59]. These approaches make it possible to use simultaneously and combine several types of data obtained in the preoperative evaluation of adnexal masses for predicting the malignancy of the tumors.

There are mainly two sources from which the mathematical models can be built and extract knowledge and information. One source is the patient data that are collected clinically. The data driven models can be developed by using various linear and nonlinear techniques [100, 104, 55, 59]. These include logistic regression (LR), multilayer perceptrons (MLPs), least squares support vector machines (LS-SVMs). Recently, more attention has been given to Bayesian approaches, which make induction and model selection based on probability theory and handle the uncertainty in modelling and prediction in a unified way. Bayesian MLPs and Bayesian LS-SVMs have been demonstrated to be powerful modelling tools for such medical diagnosis task [55, 59, 58].

Apart from the patient data, the mathematical models can also be built by use of domain knowledge, which is rich in the experts, literatures, and related documents. The Bayesian belief networks are graphical (white box) models

suitable for representation of such domain knowledge. For example, in this ovarian cancer diagnosis problem, a medical expert can provide both a qualitative and quantitative characterization of the relationships among the input variables and the type of tumors, which form into an expert Bayesian belief network as described in [28]. The relations among the variables and tumor types could also be extracted from the electronic literature in domain of ovarian masses in the context of text mining (for which a set of domain key words related to the variables usually need to be defined). And the extracted quantitative relationships can be taken as the structural priors in training of Bayesian belief networks [6, 5, 28]. It is also possible to incorporate the prior knowledge into the learning of blackbox models by using, for example, pseudo-samples generated from the prior belief networks [5]. This hybrid approach is advantageous to the pure data driven approach when the data is noisy and scarce while large amount of domain knowledge is available. In our case, data sets of a moderate to large size are already available, the gain of using the current prior models is not significant for the blackbox models with powerful learning abilities [5][28]. Therefore in this thesis we would rather concentrate on the data driven blackbox modelling.

We report on the development and evaluation of several probabilistic models, including the classical logistic regression, and other computational intelligent methods: Bayesian MLPs, Bayesian LS-SVMs and relevance vector machines (RVMs). Particularly, the application of kernel-based algorithms within the Bayesian framework is our focus for this preoperative ovarian tumor classification problem. The analysis encompasses exploratory data analysis, optimal input variable selection, parameter estimation, and performance evaluation via Receiver Operating Characteristic (ROC) curve analysis.

The study described in this chapter consists of two consecutive stages involving first a pilot project on ovarian tumor analysis conducted in Leuven, and then the extended project on an international multi-center study on ovarian tumor analysis (IOTA), both initialized and coordinated by Prof. Dr. Dirk Timmerman from the University Hospitals Leuven. The pilot project collected 525 patient data with ovarian masses falling into two major categories: benign or malignant, within a single-center in Leuven. A couple of models have been built and evaluated based on this data set. However, a sufficiently large size and the variability of the data set are essential to ensure the development of reliable diagnostic models, thus an international multi-center cooperation was established within the framework of the IOTA project. Patient recruitment and data collection has been guided by a standard protocol, and supported by a web-based data entry tool [103]. This large IOTA data set contains the records from 1066 non-pregnant women collected in nine different centers from five European countries.

We initially developed and evaluated mathematical models for ovarian tumor classification based on the single-center data set from the pilot project. Then

the experiences and some conclusions from the pilot project can be passed on the next stage of the study, the multi-center study. We have then the chance to test prospectively the model we built from a single center data set on the multi-center data set. Finally, we worked on the IOTA data set, and proceeded with another round of model development and validation.

4.2 Methods

The probabilistic machine learning algorithms introduced in previous chapters have been used in this application. The compared models include logistic regression, Bayesian multilayer perceptrons, Bayesian least squares support vector machines and relevance vector machines. Particularly, for these probabilistic models, except for logistic regression models, the Bayesian learning approach has been used for tuning the hyperparameters and comparison of the models.

In medical classification problems, variable selection can have an impact on the economics of data acquisition and the accuracy and complexity of the classifiers, and is helpful in understanding the underlying mechanism that generated the data.

We have utilized two prominent approaches for input variable selection. One approach uses stepwise logistic regression procedure utilizing partial F-statistics, followed by backward elimination.

In a Bayesian framework, the models (either linear or nonlinear) can be ranked by evaluating the posterior probability for the model $p(\mathbf{y}|\mathcal{H}_i)$. By combining the model ranking with the heuristic search method, we can find the optimal subset of variables as well.

To start the forward selection (greedy search) based on the evidence, we should define the model type a priori by e.g. giving the kernel type. The procedure starts from zero variables, and chooses each time the variable which gives the greatest increase in the current model evidence. The selection is stopped when the addition of any remaining variables can no longer increase the model evidence.

Rather than using a heuristic search, by use of a Bayesian MLP with ARD, one can also obtain the relevance for the sets of candidate variables. However, we met some numerical problems when working on the ovarian tumor data. Moreover, for Bayesian LS-SVMs, given the hyperparameters, the “most probable" estimate of the parameters can be found uniquely by simply solving some linear equations. In contrast, for RVMs, an iterative algorithm is needed, and some costly matrix inversions are also involved. Thus Bayesian LS-SVM appears

faster and more stable than the RVM. Therefore, variable selection based on the Bayesian LS-SVM has been emphasized in this study.

We then compared the performance of the various models with different subsets of variables selected from different methods.

4.3 Preliminary investigation - on the K.U. Leuven pilot project

4.3.1 Data

The data set in the pilot project includes the information of 525 patients who were referred to a single ultrasonographer at the University Hospitals Leuven, Belgium, between 1994 and 1999. These patients have a persistent extrauterine pelvic mass, which was subsequently surgically removed. The study is designed mainly for preoperative differentiation between benign and malignant adnexal masses [100, 104, 103]. Patients without preoperative results of serum CA 125 levels are excluded from this analysis. The gold standard for discrimination of the tumors was the results of histological examination. Among the available 425 cases, 291 patients (68.5%) had benign tumors, whereas 134 ones (31.5%) had malignant tumors.

The medical features on demography and some clinical examinations were acquired before operation, including: age and menopausal status of the patients, serum CA 125 levels from the blood test, the ultrasonographic morphologic findings about the mass, color Doppler imaging and blood flow indexing, etc. [104, 103]. The data set contains 27 variables after preprocessing (e.g. color score was transformed into three dummy variables, CA 125 serum level was rescaled by taking its logarithm). Table 4.1 lists the important variables that were considered.

Figure 4.1 shows the biplot generated by the first two principal components of the data set, visualizing the correlation between the variables, and the relations between the variables and classes. In particular, a small angle between two variables such as (Age, Meno) points out that those variables are highly correlated; the observations of malignant tumors (indicated by '+') have relatively high values for variables Sol, Age, Meno, Asc, L_CA125, Colsc4, Pap, Irreg, etc; but relatively low values for the variables Colsc2, Smooth, Un, Mul, etc. The biplot reveals that many variables are correlated, implying the need of variable selection. On the other hand, quite a lot of overlap between the two classes can also be observed, suggesting that the classical linear techniques might not be enough to capture the underlying structure of the data.

Table 4.1: Descriptive statistics for variables of demography, serum marker, color Doppler imaging and morphology

	Variable (Symbol)	Benign	Malignant
Demographic	Age (Age)	45.6±15.2	56.9±14.6
	Postmenopausal (Meno)	31.0 %	66.0 %
Serum marker	CA 125 (log)(L_CA125)	3.0±1.2	5.2±1.5
CDI	Weak blood flow (Col2)	41.2 %	14.2 %
	Normal blood flow (Colsc3)	15.8 %	35.8 %
	Strong blood flow (Colsc4)	4.5 %	20.3 %
	Pulsatility index (PI)	1.34±0.94	0.96±0.61
	Resistance index (RI)	0.64±0.16	0.55±0.17
	Peak systolic velocity (PSV)	19.8±14.6	27.3±16.6
	(Time-averaged)Mean velocity (TAMX)	11.4± 9.7	17.4±11.5
B-Mode Ultrasono Graphy	Abdominal fluid (Asc)	32.7 %	67.3 %
	Unilocular cyst (Un)	45.8 %	5.0 %
	Unilocular solid (Unsol)	6.5 %	15.6 %
	Multilocular cyst (Mul)	28.7 %	5.7 %
	Multilocular solid (Mulsol)	10.7 %	36.2 %
	Solid tumor (sol)	8.3 %	37.6 %
Morphology	Bilateral mass (Bilat)	13.3 %	39.1 %
	Smooth wall (Smooth)	56.8 %	5.8 %
	Irregular wall (Irreg)	33.8 %	73.2 %
	Papillations (Pap)	13.0 %	53.2 %
	Septa> 3mm (Sept)	13.0 %	31.2 %
	Acoustic shadows(Shadows)	12.2 %	5.7 %
Echogenicity	Anechoic cystic content(Lucent)	43.2 %	29.1 %
	Low level echogenicity (Low_level)	12.0 %	19.9 %
	Mixed echogenicity (Mixed)	20.3 %	13.5 %
	Ground glass cyst (G_glass)	19.8 %	8.5 %
	Hemorrhagic cyst (Haem)	3.9 %	0.0 %

Note: for continuous variables, mean±SD over the class of benign and malignant tumors respectively, are reported; for binary variables, the occurrences (%) of the corresponding features are reported.

4.3.2 Experimental settings

Our experiments consist of two parts. One part is the so-called "temporal validation". In these experiments, the data set is split according to the time scale: the data from the first treated 265 patients (collected between 1994 and 1997) are taken as training set, 160 of the remaining data (collected between 1997 and 1999) are used as test set. In this way, we are trying to mimic a kind of prospective evaluation of our models, which is important before using the models in clinical practice. The proportion of malignant tumors in the training set and in the test set is both about 1/3. However, such hold out cross validation is somehow biased and depends on the splitting of the training and test sets [3]. Hence in the second part of the experiment, we try to reduce

the bias by randomizing the data set and repeat the hold out cross-validation 30 times. The generalization performance of the models will be evaluated by averaging. The proportion of the malignant cases has been kept to be the same (about 1/3) for all the realizations of the training and test sets.

All the input data have been normalized using the mean and variance estimated from the training data. Several competitive models are built and evaluated using the same variables selected from the procedures including stepwise logistic regression and forward selection based on the Bayesian LS-SVM model evidence. Various models have been developed and evaluated using the following probabilistic modelling techniques: logistic regression, Bayesian MLPs, Bayesian LS-SVM with linear and RBF kernels, and the relevance vector machines with linear and RBF kernels. We also show the performance for non-probabilistic models such as standard SVMs and LS-SVMs, and we can see why the probabilistic output is needed for our cancer diagnosis tasks.

All the experiments have been done in matlab, except for the explorative data analysis and stepwise logistic regression which were conducted in SAS. Several matlab toolboxes have been employed, for example Netlab [68] was used for logistic regression and Bayesian MLP modelling, the LS-SVMlab [78] for standard and Bayesian LS-SVM modelling, the SparseBayes V1.0 [107] for fast sequential sparse Bayesian modelling after some modification.

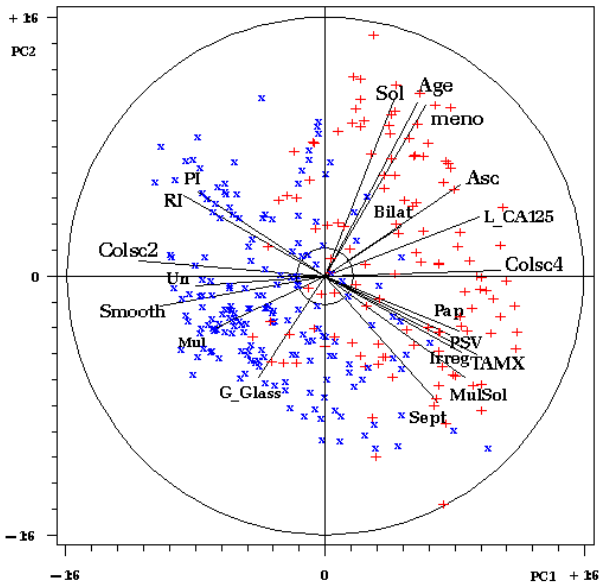


Figure 4.1: Biplot of ovarian tumor data ('×'- benign, '+'- malignant), projected on the first two principal components.

4.3.3 Selecting predictive input variables

Consistent with our "temporal" validation, we selected the variables based on only the 265 training data collected between 1994 and 1997. Firstly, we adapted forward selection which tries to maximize the evidence of the LS-SVM classifiers with either linear or RBF kernels. In order to stabilize the selection, the three variables with the smallest univariate model evidence are first removed. Then the selection was limited within the remaining 24 candidate variables. Figure 4.2 shows the evolution of the model evidence during variable selection using RBF kernels. The Bayes factor for the univariate model is obtained by comparing it to a model with only a random variable, the other Bayes factors are obtained by comparing the current model to the previously selected models. The variable added to the model at each selection step and the corresponding Bayes factor have been depicted. We ended up with 10 variables based on LS-SVM with RBF kernels. The order of selection for the 10 variables are the following : *L_CA125, Pap, Sol, Colsc3, Bilat, Meno, Asc, Shadows, Colsc4, Irreg*, and we denote this subset of variables as MODEL1.

LS-SVMs with linear kernels have also been tried for variable selection, but resulted into a smaller evidence and an inferior model performance. Thus no discussions will be given on the models based on the subset of variables selected using linear kernels.

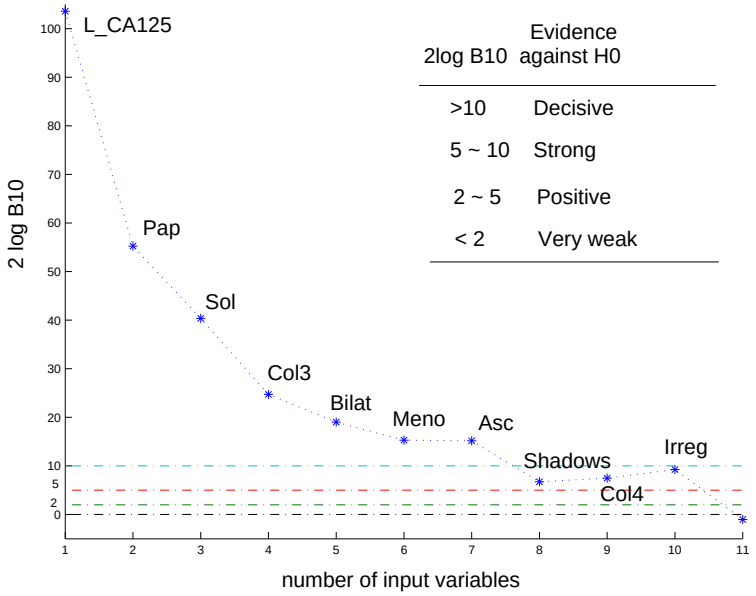


Figure 4.2: Evolution of the model evidence during forward input selection for LS-SVM with RBF kernels.

We have also performed classical stepwise logistic regression, and only eight variables are needed for the final logit model to obtain a satisfied goodness of fit (compared with the full model that uses all the variables): *L_CA125, Pap, Sol, Asc, smooth, Colsc3, Unsol, Shadows*. This model seems a little bit disappointing for us. According to the medical expert, variable *Meno* should be included in the model, since the postmenopausal patients have higher risk of ovarian cancer than for the premenopausal patients. However, in order to have a fair comparison for the variable selection method, i.e., without considering the expert knowledge, we keep using this subset of variables in this study.

In practice, many other models have been built by trying different combinations of the variables and taking into account the opinion of the experts. Other types of variable selection procedure based on logistic regression have also been tried, such as backward, forward selection, and the score selection based on the likelihood score (chi-square) statistic for various model sizes. Here, we won't describe all those models. Instead, we pay more attention to the models selected with least human interaction, although in the end we hope the automatically selected variables to be meaningful and interpretable for the experts.

4.3.4 Model fitting and prediction

For MLP models, we use MacKay's Bayesian MLP classifier [60, 10], which is limited to one hidden layer with two hidden neurons, with hyperbolic tangent activation function for the hidden layer, and sigmoidal logistic activation function for the output layer. Other models with various numbers of hidden neurons were also tried, but not reported here due to their smaller evidence and inferior performance on the test set. Because of the existence of multiple local minima, the MLP classifier was trained 10 times with different initialization of the weights, and the one with the highest evidence was chosen.

The model fitting procedure for Bayesian LS-SVM classifiers has two stages. The first is the construction of the standard LS-SVM model within the evidence framework. Sparseness can be imposed to LS-SVMs at this stage in order to improve the generalization ability, by iteratively pruning, e.g. the 'easy' cases which have negative support values. At the second stage, all the available training data will be used to compute the output probability, indicating the posterior probability for a tumor to be malignant.

As to relevance vector machines, given the kernel type and corresponding kernel parameter (e.g. the bandwidth r for RBF kernels), the fast sequential learning algorithm was used for training [105]. In our experiments, for each iterative optimization step, 10 relevance values (corresponding to 10 different kernel basis functions) are randomly selected as candidates for modification. We used the same approach to handle the same local minima problem as in the MLP learning: choosing the parameters with the highest model evidence from

multiple (five in this case) runs of optimization process.

In risk minimization decision making, different error costs should be considered in order to reduce the expected loss. Since misclassification of a malignant tumor is very serious, we need a model with high sensitivity while keeping an acceptable low false positive rate. The unbalance of the training data set will normally influence the predicted outcome of the models, which minimize the total errors regardless of misclassification cost. That is, the resulting model will tend to give a higher probability for the prevalent class, which is the benign one in this case. Thus the often used probability cutoff level 0.5 will lead to a low sensitivity for the malignant class, and therefore not suitable for the cancer diagnosis. Without incorporating the prior costs, an acceptable cutoff level would be less than 0.2 or so, in order to reach a sensitivity of 80%. Therefore, the adjusted prior for the malignant class was used in the experiments, which is intuitively set to 2/3, higher than that of the benign class 1/3.

The same adjusted class priors have been combined for the computation of the posterior output for all the compared models. For Bayesian LS-SVM models, this is done via (3.36). For models using logistic output activation functions, such as LRs, MLPs and RVM models, we incorporate the $2:1$ ratio of the adjusted prior class probability between the malignant and benign cases, into the computation of the probability outcome. That is, by correcting the bias term b_0 in the logit function as follows: $b = b_0 - \log(N_+/N_-) + \log(2/1)$, where N_+ and N_- denote the number of malignant and benign cases in the training set respectively.

4.3.5 Model evaluation

The model performance is assessed by ROC analysis. Unlike the classification accuracy, ROC is independent of class distributions or error costs, and has been widely used in the biomedical field [80]. The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) for the different cutoff values of a diagnostic test. Here the *sensitivity* and *specificity* is the correct classification rate for the malignant and benign class, respectively. The area under the ROC curve (AUC) can be statistically interpreted as the probability of the classifier to correctly classify malignant cases and benign cases [38]. In this work, the AUC was obtained by a nonparametric method based on the Wilcoxon statistic, using the trapezoidal rule, to approximate the area [38]. The method proposed by DeLong *et al.* [20] was used to compute the variance and covariance of the nonparametric AUC estimates derived from the same set of cases. AUC can be used for comparing two different ROC curves. Related performance measures include also positive predictive value (PPV) and negative predictive value (NPV). The PPV is the proportion of all positive tests that are true positive; the NPV is the proportion of all negative tests that are true negative. The predictive value can help in interpreting the

test result for an individual.

Results from temporal validation

Table 4.2 reports the performance of different models on the test set. The performance of RMI, a widely used score system (calculated as the product of the CA 125 level, a morphologic score, and a score for the menopausal status), is also shown as a reference [44]. The AUC and its corresponding standard error (SE) [20], which are independent of the cutoff value, are reported in the second column. The four performance measures, including the accuracy, sensitivity, specificity, PPV and NPV, were calculated at different decision levels. Except for RMI which is not a probabilistic model, all the other decision levels are actually probability cutoff levels. For RVM models, the numbers between the parentheses in the first column indicate the number of support (or relevant) vectors (N_{SV}) used for prediction. For LS-SVM classifiers, N_{SV} refers to the number of support vectors used in the first stage of model fitting other than in prediction, since all the training data have been be used in prediction (when computing the probability distributions of the two classes over the latent output) in this study. Note that, the decision levels we used in the experiments are considered 'good' according to our goal. They lead to a high sensitivity and low false positive rate on the training set; those decision levels, which result into a high accuracy but a too low sensitivity or specificity, are considered unacceptable in this context.

The ROC curves for the series of linear models and nonlinear models have been depicted in Figure 4.3 and Figure4.4, respectively. The ROC curve for RMI on the test set is shown in both figures for comparison.

All the competitive models, both linear and nonlinear models seem to perform much better than RMI. Let us consider now the case in which the models are given the same sets of input variables. The LR models do not perform as well as the other linear and nonlinear models. This can be partially explained by the fact that no regularization has been used in LR modelling. And no significant difference can be observed among the other models including the Bayesian MLPs, the Bayesian LS-SVMs with linear and RBF kernels, and the RVMs with linear and RBF kernels.

The sparsity of the RVM models deserves our special attention. The parsimony of the models can not only improve the generalization capability but also increase the interpretability of the model. Only five data points are needed for computing the outcome of RVM models both with linear and RBF kernels given MODEL1.

As to the input variable selection methods, the forward selection based on the RBF LS-SVMs performed better than the stepwise logistic regression. This is quite obvious, since models using the 10 variables of MODEL1 obtained better

Table 4.2: Comparison of the temporal validation performance on the **test set** ($N_{\text{train}} = 265, N_{\text{test}} = 160$).

Model Type (N_{SV})	AUC $\pm SE$	Decision Level	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
RMI	0.8733	100	78.13	74.07	80.19	65.57	85.86
	±0.0298	**75**	**76.88**	**81.48**	**74.53**	**61.97**	**88.76**
LR1	0.9111	0.5	81.25	74.07	84.91	71.43	86.54
	±0.0246	0.4	80.63	75.96	83.02	69.49	87.13
		0.3	80.63	77.78	82.08	68.85	87.88
		0.2	**80.63**	**81.48**	**80.19**	**67.69**	**89.47**
MLP1	0.9174	0.5	82.50	77.78	84.91	72.41	88.24
	±0.0224	0.4	83.13	81.48	83.96	72.13	89.90
		0.3	**81.87**	**83.33**	**81.13**	**69.23**	**90.53**
LS-SVM1$_{\text{Lin}}$	0.9141	0.5	82.50	77.78	84.91	72.41	88.24
(118)	±0.0236	0.4	81.25	77.78	83.02	70.00	88.00
		0.3	**81.87**	**83.33**	**81.13**	**69.23**	**90.53**
LS-SVM1$_{\text{RBF}}$	0.9184	0.5	84.38	77.78	87.74	76.36	88.57
(97)	±0.0225	0.4	83.13	81.48	83.96	72.13	89.90
		0.3	**84.38**	**85.19**	**83.96**	**73.02**	**91.75**
RVM1$_{\text{Lin}}$	0.9196	0.5	83.75	81.48	84.91	73.33	90.00
(5)	±0.0231	0.4	83.13	85.19	82.08	70.77	91.58
		0.3	**83.13**	**85.19**	**82.08**	**70.77**	**91.58**
RVM1$_{\text{RBF}}$	0.9195	0.5	84.38	85.19	83.96	73.02	91.75
(5)	±0.0222	0.4	83.75	85.19	83.02	71.88	91.67
		0.3	**83.75**	**87.04**	**82.08**	**71.21**	**92.55**
LR2	0.9106	0.5	82.50	74.07	86.79	74.07	86.79
	±0.0229	0.4	80.63	77.78	82.08	68.85	87.88
		0.3	**80.63**	**79.63**	**81.13**	**68.25**	**88.66**
		0.2	77.50	83.33	74.53	62.50	89.77
MLP2	0.9144	0.5	83.75	79.63	85.85	74.14	89.22
	±0.0225	**0.4**	**80.63**	**79.63**	**81.13**	**68.25**	**88.66**
		0.3	78.75	83.33	76.42	64.29	90.00
LS-SVM2$_{\text{Lin}}$	0.9144	0.5	84.38	79.63	86.79	75.44	89.32
(106)	±0.0224	0.4	80.63	79.63	81.13	68.25	88.66
		0.3	**80.00**	**83.33**	**78.30**	**66.18**	**90.22**
LS-SVM2$_{\text{RBF}}$	0.9158	0.5	84.38	79.63	86.79	75.44	89.32
(97)	±0.0224	0.4	83.13	79.63	84.19	72.88	89.11
		0.3	**79.37**	**83.33**	**77.36**	**65.22**	**90.11**
RVM2$_{\text{Lin}}$	0.9158	0.5	83.75	75.93	87.74	75.93	87.74
(4)	±0.0222	**0.4**	**80.63**	**83.33**	**79.25**	**67.16**	**90.32**
		0.3	79.37	85.19	76.42	64.79	91.01
RVM2$_{\text{RBF}}$	0.9147	0.5	81.25	79.63	82.08	69.35	88.78
(10)	±0.0225	**0.4**	**80.63**	**79.63**	**81.13**	**68.25**	**88.66**
		0.3	79.37	87.04	75.47	64.38	91.95

The 'best' results of each model obtained at a certain decision level are indicated in bold. A number is added to the name of the classifier in order to distinguish the subsets of variables in use: '1' stands for MODEL1 and '2' for MODEL2. Subscripts 'RBF' and 'Lin' indicate the kernel type that is used.

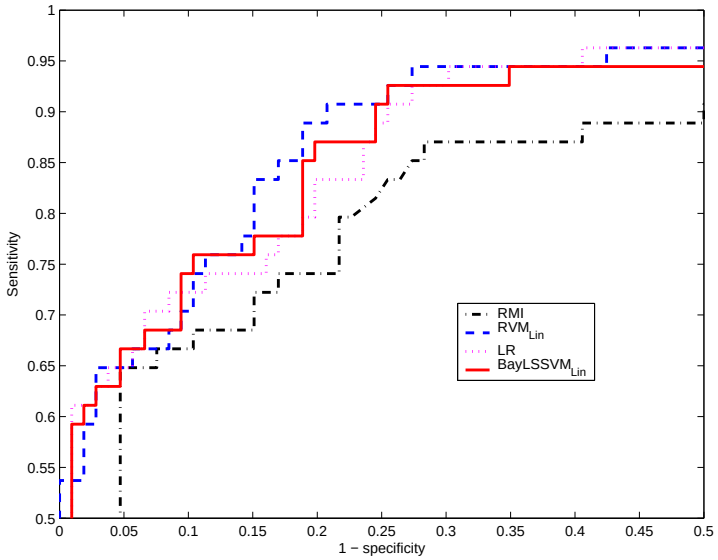


Figure 4.3: ROC curves on the test set. Three types of linear models using MODEL1 (ten variables) and the RMI as a reference model have been compared.

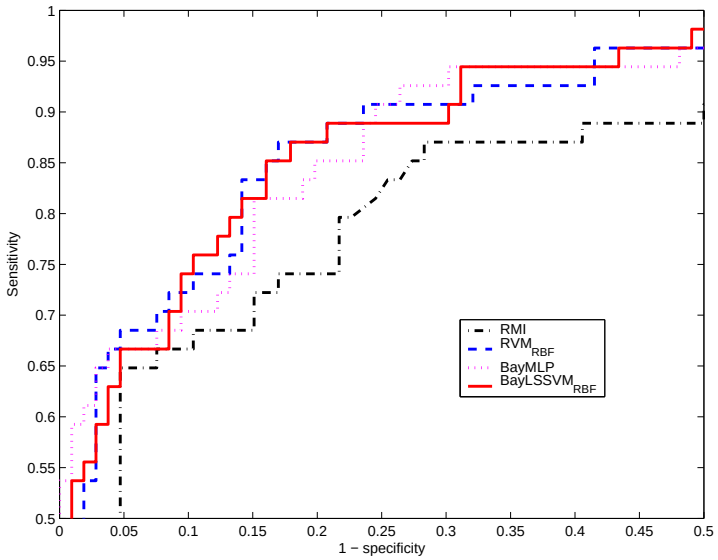


Figure 4.4: ROC curves on the test set. Three types of nonlinear models using MODEL1 (ten variables) and the RMI as reference model have been compared.

performance compared to using MODEL2. As we mentioned previously, the lack of menopausal status for a patient might be one reason why the models with MODEL2 do not generalize as well as with MODEL1.

We also tried to check the ability of our classifiers in rejecting uncertain test cases which need further examination by a human expert. The discrepancy (in absolute value) between the posterior probability and the cutoff value reflects the uncertainty of the prediction: the smaller the discrepancy, the larger the uncertainty. The performance of the models has been re-evaluated after rejecting a certain number of the most 'uncertain' test cases. It is shown in Figures from 4.5 to 4.8, how the rejection of uncertain cases can improve the performance of the probabilistic models.

The performance with rejection for standard SVM models and standard LS-SVM models is also depicted for comparison. Here the hyperparameters for both standard SVM and LS-SVM models were selected using 5-fold cross validation. Due to the absence of the probability outcome for these non-probabilistic models, the rejection is based on the absolute value of the predictive outcome (as the discrepancy between zero and the latent output value) of the model. If this value reflects the uncertainty in prediction, the similar trend should be observed, that is, the performance of the model should increase when a larger fraction of test cases is rejected.

Figure 4.5 and Figure 4.6 show the AUC-rejection curves for the linear and non-linear models using MODEL1, respectively. In both cases, it is observed that all probabilistic models keep getting higher performance after rejecting more and more uncertain cases, resulting into some sort of smoothly increasing curves. Some models such as linear Bayesian LS-SVM, linear RVM and Bayesian MLP and RVM with RBF kernels can even reach AUC of 0.95 after rejecting 18% of uncertain cases. On the contrary, the curves for non-probabilistic models of SVM and LS-SVM models appear to be bumpy and even keep decreasing in some regions. Moreover, probabilistic models obtain much higher AUCs compared to the others. This indicates that the posterior probability of the models does provide a way of quantifying uncertainty, which is essential for decision making in medical diagnosis.

Now we focus on the models of our interest - probabilistic models. More details of the performance at the cutoff level of 0.5 could be seen in Figure 4.7 and Figure 4.8, where the sensitivity and the false positive rate have been plotted against the rejection rate. We see clearly how the sensitivity rises slowly with the increase of the rejection rate, while the false positive rate decreases with a slightly higher speed. This shows that the decrease in false alarm rate contributes more in the improvement of the overall model performance.

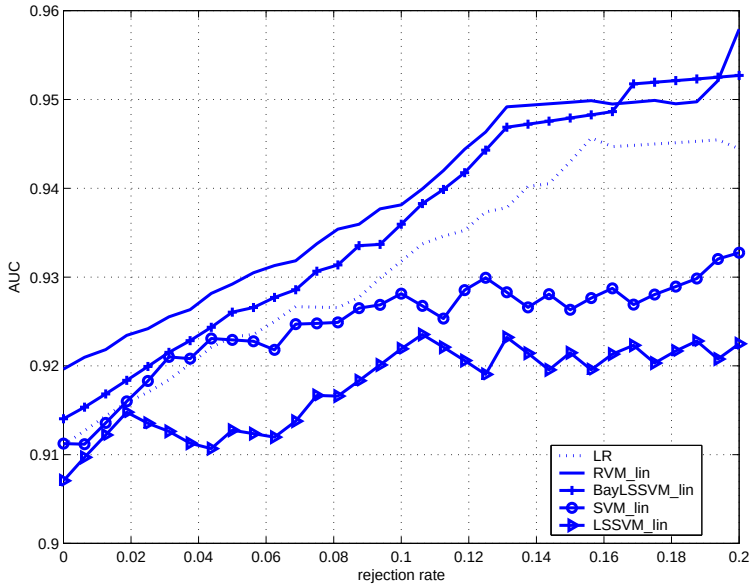


Figure 4.5: Test AUC vs. increasing rejection rate for different linear models.

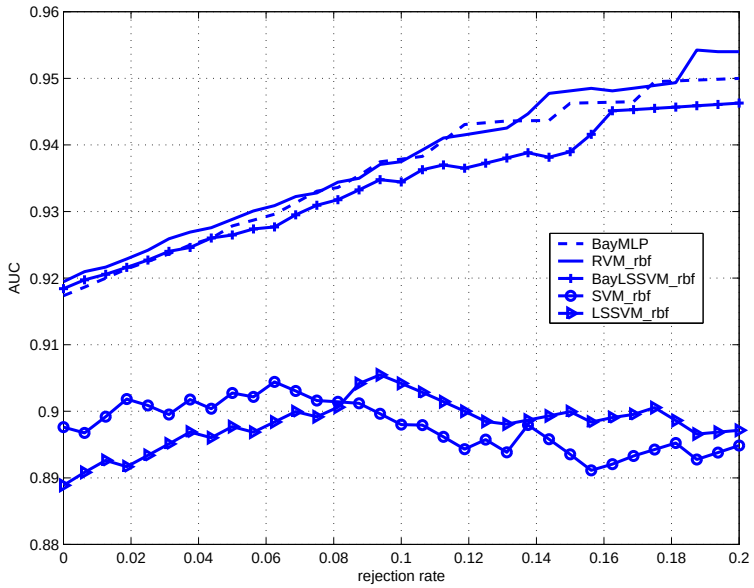


Figure 4.6: Test AUC vs. increasing rejection rate for different nonlinear models.

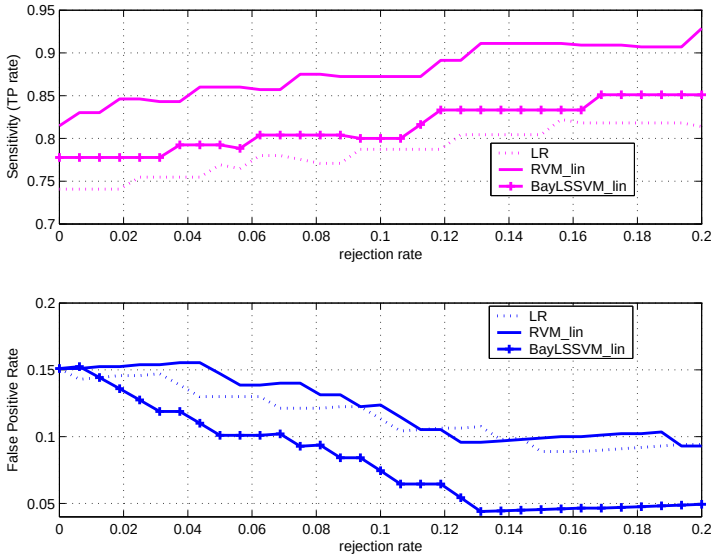


Figure 4.7: Test sensitivity (upper plot) and false positive rate (bottom plot) vs. increasing rejection rate for different linear models (at 0.5 probability cutoff level).

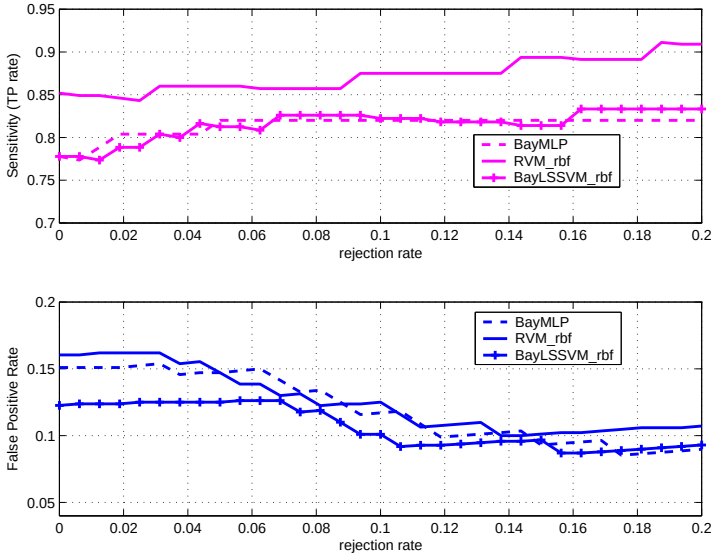


Figure 4.8: Test sensitivity (upper plot) and false positive rate (bottom plot) vs. increasing rejection rate for different nonlinear models (at 0.5 probability cutoff level).

Results from multiple runs of randomized cross validation

In order to see the performance of our model with less bias, we performed another set of experiments in which the holdout cross-validation based on random splitting of training and test set was repeated 30 times. The average performance is considered as the model performance in the ideal situation where the training and test set have similar distributions.

Table 4.3 presents the averaged performance of the competing models from the 30 runs of randomized cross validation by using MODEL1 and MODEL2. The average of AUC ($\overline{\text{AUC}}$), the corresponding standard deviation (SD, derived from 30 AUC values) are reported. The number between the parentheses indicates the mean of the number of support vectors or relevance vectors ($\overline{N_{\text{SV}}}$).

From this experiment, an increase of the validation performance is observed, which is mainly due to the randomization of the training set and test set. Among all the models, RMI has the worst performance. LS-SVM1$_{\text{RBF}}$ obtained the best averaged performance, with mean AUC=0.9424. However, mixed performances were obtained by both the linear models and nonlinear models. No conclusions can be drawn that nonlinear models are superior to the linear models (or models using linear kernels), or vice versa. It is also difficult to tell if MODEL1 gives better results than MODEL2.

Comparison with human investigators

One might be curious to know: can the computer model beat the expert? Trying to answer this question, we would like to compare the diagnostic results of our models with those of human investigators examining the same patients. The investigators were given all the available information and measurements of the patients before operation [101]. Table 4.4 shows the diagnostic performance of both the LS-SVM1$_{\text{RBF}}$ and the three human investigators. Assessor 1 (DT) is a very experienced expert, who had examined ultrasonographically more than 5000 patients. Assessor 2 and 3 are less experienced, who had performed about 200 and 300 ultrasonographical examinations respectively. Unfortunately, the assessment of the two less experienced assessors are not available on the cases collected after 1997, hence we will mainly focus on the comparison with the expert.

The expert's diagnosis on the 160 newest patients (test set) results in a sensitivity of 81.48%, specificity 93.40% and PPV 86.27%. The LS-SVM1$_{\text{RBF}}$ model gives a diagnostic performance at 0.4 decision level with a sensitivity of 81.48% (same as for the expert), however, a lower specificity of 83.96%, and PPV 72.13%. When looking at the averaged performance on the same randomized cross-validation, similar conclusions can be drawn. The human expert has a sensitivity 91.13%, specificity 90.33% and PPV 81.23%, while the LS-SVM1$_{\text{RBF}}$ has an averaged sensitivity of 90.00%, specificity 80.58% and PPV 67.98%. In

Table 4.3: Averaged performance on the **test set** from 30 runs of randomized cross validation ($N_{\text{train}} = 265, N_{\text{test}} = 160$)

Model Type ($\overline{N_{\text{SV}}}$)	$\overline{\text{AUC}}$ $\pm SD$	Decision Level	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
RMI	0.8882	100	82.65	81.73	83.06	68.89	90.96
	±0.0284	**80**	**81.10**	**83.87**	**79.85**	**65.61**	**91.63**
LR1	0.9397	0.6	84.94	86.33	84.30	71.64	93.24
	±0.0209	**0.5**	**83.29**	**89.33**	**80.55**	**67.81**	**94.43**
		0.4	81.94	91.60	77.55	65.16	95.38
MLP1	0.9409	0.6	84.44	87.20	83.18	70.40	93.55
	±0.0198	**0.5**	**82.15**	**90.73**	**78.24**	**65.64**	**94.96**
		0.4	79.65	93.27	73.42	61.64	96.08
LS-SVM1$_{\text{Lin}}$	0.9405	0.5	84.31	87.40	82.91	70.09	93.62
(150.2)	±0.0199	**0.4**	**82.77**	**90.47**	**79.27**	**66.61**	**94.88**
LS-SVM1$_{\text{RBF}}$	0.9424	0.5	84.85	86.53	84.09	71.46	93.31
(137.1)	±0.0207	**0.4**	**83.52**	**90.00**	**80.58**	**67.98**	**94.71**
RVM1$_{\text{Lin}}$	0.9411	0.6	84.81	86.93	83.85	71.21	93.49
(4.6)	±0.0199	**0.5**	**83.06**	**90.00**	**79.91**	**67.29**	**94.71**
		0.4	81.15	92.67	75.91	63.81	95.87
RVM1$_{\text{RBF}}$	0.9407	0.6	84.96	86.53	84.24	71.60	93.32
(6.1)	±0.0195	**0.5**	**83.13**	**89.67**	**80.15**	**67.39**	**94.55**
		0.4	81.25	92.20	76.27	64.04	95.61
LR2	0.9399	0.6	86.06	88.47	84.97	73.17	94.29
	±0.0175	**0.5**	**83.31**	**91.40**	**79.64**	**67.44**	**95.44**
		0.4	80.83	94.20	74.76	63.18	96.66
MLP2	0.9404	0.6	85.81	88.47	84.61	72.65	94.26
	±0.0181	**0.5**	**82.94**	**92.27**	**78.70**	**66.68**	**95.82**
		0.4	79.60	94.20	72.97	61.53	96.57
LS-SVM2$_{\text{Lin}}$	0.9413	0.6	86.71	88.07	86.09	74.44	94.13
(149.7)	±0.0173	**0.5**	**84.60**	**91.00**	**81.70**	**69.59**	**95.30**
		0.4	82.35	93.47	77.30	65.40	96.37
LS-SVM2$_{\text{RBF}}$	0.9376	0.6	86.33	87.00	86.03	74.12	93.63
(145.2)	±0.0198	0.5	84.63	90.00	82.18	69.90	94.85
		0.4	82.42	92.67	77.76	65.72	95.98
RVM2$_{\text{Lin}}$	0.9419	0.6	85.98	89.13	84.55	72.69	94.57
(4.8)	±0.0172	**0.5**	**82.85**	**92.60**	**78.42**	**66.42**	**95.97**
		0.4	79.96	94.47	73.36	61.89	96.74
RVM2$_{\text{RBF}}$	0.9405	0.6	85.23	89.40	83.33	71.18	94.63
(6.2)	±0.0185	**0.5**	**82.67**	**92.80**	**78.06**	**66.13**	**96.07**
		0.4	79.87	94.80	73.09	61.80	96.93

The 'best' results of each model obtained at a certain decision level are indicated in bold. A number is added to the name of the classifier in order to distinguish the subsets of variables in use: '1' stands for MODEL1 and '2' for MODEL2. Subscripts 'RBF' and 'Lin' indicate the kernel type that is used.

Table 4.4: Comparison in diagnostic performance of the model and the human assessors.

(a) on the 265 cases collected between 1994 and 1997

	Accuracy	Sensitivity	Specificity	PPV	NPV
LS-SVM1$_{RBF}$	0.8981	0.9750	0.8649	0.7573	0.9877
Assessor1	0.9132	0.9750	0.8865	0.7879	0.9880
Assessor2	0.8189	0.9000	0.7838	0.6429	0.9477
Assessor3	0.8113	0.8750	0.7838	0.6364	0.9355

(b) on the 160 cases collected between 1997 and 1999

	Accuracy	Sensitivity	Specificity	PPV	NPV
LS-SVM1$_{RBF}$	0.8313	0.8148	0.8396	0.7213	0.8990
Assessor1	0.8938	0.8148	0.9340	0.8627	0.9083

(c) averaged on the test sets from the randomized cross-validation

	Accuracy	Sensitivity	Specificity	PPV	NPV
LS-SVM1$_{RBF}$	0.8352	0.9000	0.8058	0.6798	0.9471
Assessor1	0.9058	0.9113	0.9033	0.8123	0.9574

Note: the results of LS-SVM1$_{RBF}$ are obtained at the decision level of 0.4 in (a)(b)(c), and in (a) are the performances of recalling on the training set.

summary, the LS-SVM model can achieve the same sensitivity as the expert, however at the cost of a higher false positive rate.

The comparison points out that the models we have till now have not yet been able to beat the experienced human expert. However, from Table 4.4(a), we observe that the model performs significantly better than the other less experienced assessors 2 and 3 on the old patient group. If the model is assessed by the average performance from the randomized cross-validation, it can also be inferred that the model can better discriminate preoperatively between benign and malignant tumors than the less experienced assessors.

4.4 Extensive study - on the multi-center IOTA project

4.4.1 The IOTA project

The work reported in this section was within the framework of the International Ovarian Tumor Analysis (IOTA), a multi-center study on the pre-operative characterisation of ovarian tumors based on artificial intelligence models.

A protocol for data collection has been setup [103]. All patients that entry into

Table 4.5: The participant centers in IOTA project.

Center	City, Country	Address
MSW	Malmö, Sweden	Obstetrics and Gynecology, University Hospital, SE20502 Malmö, Sweden
RIT	Roma, Italy	Istituto di Clinica Ostetrica e Ginecologica, Università Cattolica di Sacro Cuore, Italy
MIT	Milan, Italy	Obstetrics and Gynecology, ISBM L. Sacco, Università di Milano, Milan, Italy
OIT	Milan, Italy	Obstetrics and Gynecology, ISBM Ospedale San Gerardo, Università di Milano, Monza, Italy
NIT	Napoli, Italy	Obstetrics and Gynecology, Università degli Studi di Napoli "Federico II", Italy
LBE	Leuven, Belgium	Department of Obstetrics and Gynaecology, University Hospitals Leuven, Belgium
BFR	Paris, France	Service de Gynécologie et Obstétrique, Hôpital Boucicaut, France
MFR	Paris, France	Service Gyn.-Obstétrique, Centre Médical des Pyramides, Maurepas, France
KUK	London, UK	Department of Obstetrics and Gynaecology, King's College Hospital, UK

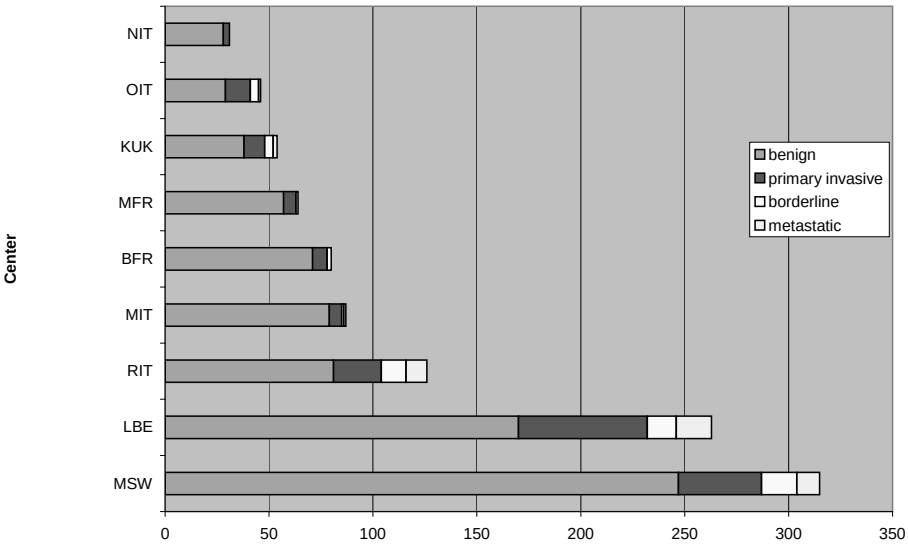
the study were found to have an apparent persistent extrauterine pelvic mass, and took the surgery within one week after the ultrasonography scans. Tumors are classified according to the criteria recommended by the International Federation of Gynecology and Obstetrics (FIGO). In malignant tumors the degree of differentiation is also given. Quality control has been conducted over the examination of the adnexal masses as well as over the surgical investigation and tumor classification and staging.

In the first stage of the trial, more than 1000 patients with at least one histogically examined adnexal mass have been recruited. Compared with the pilot project between 1997 and 1999, the IOTA project has collected a much larger amount of data, including the information of 1233 tumors for 1066 non-pregnant patients from 9 different centers across over 5 different countries (see Table 4.5 for the list of the participant centers and their addresses). However, only the 1066 records of the dominant tumor for patients were used in this study for development of models distinguishing between malignant and benign tumors. For a description of the considered variables in the IOTA data set, please see Appendix B.

4.4.2 Data analysis

Statistical analysis has been performed and a logistic regression model has been built to discriminate between malignant and benign ovarian masses based upon

the 1066 data from the IOTA project [102]. Among the 1066 ovarian masses, 800 (75%) of them are benign, and 266 (25%) are malignant. Among the malignant tumors, 169 (15.9%) are primary invasive, 55 (5.2%) are borderline, and 42 (3.9%) are metastatic. The distribution of the outcome of tumors has been shown in Figure 4.9.



	MSW	LBE	RIT	MIT	BFR	MFR	KUK	OIT	NIT
metastatic	11	17	10	1	0	0	2	1	0
borderline	17	14	12	1	2	1	4	4	0
primary invasive	40	62	23	6	7	6	10	12	3
benign	247	170	81	79	71	57	38	29	28

Figure 4.9: The distribution of the types of tumors over different centers.

The same data preprocessing steps have been performed as described in paper [102], including transformation of several variables such as taking logarithm over CA125 levels, generating dummy variables for some nominal variables, trimming the extreme high values to an upper bound for a certain continuous variables, and filling the missing data (in particular the structural missing data[1]) in an appropriate way.

Table 4.6 presents the descriptive statistics for the most important variables of

[1]Unlike random missing, the structural missing is resulted from the fact that the measurement for this variable is impossible or invalid in a certain circumstances. For example, the postmenopausal bleeding is indeed only a description of a possible symptom of a postmenopausal patient; this variable will then be missing for premenopausal patients. And the blood flow index such as TAMX is only available for patients with the presence of blood flow, i.e. with a color score greater then 0.

the 1066 IOTA data in this study before filling the structural missing data.

4.4.3 Experimental settings

There are a considerable amount of missing values in the variable CA125 level. Therefore we tend to build models without the use of CA125 as well. We would like to answer the following questions and select some models for further prospective testing after the stage of model development and evaluation.

Questions of interest

- What is the performance of the models built from the pilot project on the new multi-center data? This could be seen as a kind of prospective test, which is important for a diagnostic model to be used in clinical practice.
- And how far can the modern machine learning models improve the traditional linear statistical models such as logistic regression?
- Is CA 125 blood test very important for diagnosis of the ovarian cancer even though it is notorious for a high false positive rate for patients in premenopausal status; especially when other diagnostic tests such as ultra sound sonographic examinations have already been performed?
- Does the combination of several models improve the performance of a single model even if the number of training data is already considerable large?
- Can a model built in one center be used in a different center in such an international multi-center setting?

Considered models

According to our experiences on previous study, we focused on three types of models, including logistic regression, relevance vector machines with linear or RBF kernels, Bayesian LS-SVM with linear or RBF kernels. As in the initial investigation, sparseness was always imposed to LS-SVM models within the Bayesian framework using the pruning scheme proposed in Chapter 3.

Additionally, we also try to give interpretation to the models. Unlike logistic regression models, kernel-based models such as RVMs and LS-SVMs are usually considered as blackbox models, which are difficult to interpret. In order to explain the behavior of the models, the following characteristics of the kernel-based probabilistic models have been taken into consideration.

Table 4.6: Descriptive statistics of the IOTA data set.

Variable (Symbol)	Benign	Malignant
∼ Medical history ∼		
Age (Age)	44.4±15.0	55.5±14.4
Personal history of ovarian cancer(PERHISTOVCA)	0.8%	3.0%
Postmenopausal (menopause)	32.9 %	63.5 %
Hormonal therapy (HORMTHERAPY)	23.5%	17.7%
Pelvic pain during the scan (PAIN)	28.8 %	19.6%
Presence of postmenopausal bleeding (PMB)	14.9 %	17.9%
∼ Serum marker∼		
CA 125 (log)(L_CA125)	3.0±0.9	5.1±1.9
∼ Color Doppler Imaging (CDI)∼		
Color score (0,1-4) (COLORSCORE)	2.1±0.9	3.2±0.8
Weak blood flow (col2)	41.2 %	14.2 %
Normal blood flow (col3)	15.8 %	35.8 %
Strong blood flow (col4)	4.5 %	20.3 %
Pulsatility index (PI)	1.11±0.68	0.82±0.38
Resistance index (RI)	0.61±0.14	0.52±0.15
Peak systolic velocity (PSV)	14.9±11.8	31.5±24.7
(Time-averaged)Mean velocity (TAMX)	9.0± 7.7	22.1±17.9
∼ B-Mode Ultrasonography∼		
Presence of abdominal fluid (Ascites)	2.9 %	42.1 %
Fluid in the pouch of Douglas (FLUID)	2.7±7.2	15.1±19.6
Tumor ovarian origin (origin1)	81.9 %	83.5 %
Ovary size in mm (OvDmax)	70.6±43.7	106.7±60.1
Lesion size in mm (LesDmax)	72.1±43.8	108.0±58.3
Unilocular cyst (uni)	38.9 %	0.8 %
Unilocular solid (unisol)	11.0 %	16.5 %
Multilocular cyst (mul)	22.0 %	7.5 %
Multilocular solid (mulsol)	21.0 %	43.6 %
Solid tumor (sol)	6.5 %	31.6 %
Solid component size in mm (SolidDMax)	29.4±29.5	63.0±40.9
Number of locules (NRLOCULES)	2.1±1.7	3.1±2.5
∼ Morphology∼		
Number of papillary projections (PAPNR)	1.8±1.1	3.0±1.2
Presence of papillary projections (PAPILLATION)	19.5 %	45.5 %
Presence of flow within papillary projections (PAPFLOW)	33.3%	84.3%
Smooth papillation (PapSmooth)	56.8 %	5.8 %
Bilateral mass (bilateral)	16.6 %	31.2 %
Irregular wall (WALLREGULARITY)	32.8 %	81.6 %
Septa> 3mm (septumg3)	13.9 %	35.3 %
Acoustic shadows(Shadows)	13.0 %	1.5 %
∼ Echogenicity∼		
Anechoic cystic content(lucent)	37.9 %	40.2 %
Low level echogenicity (low_level)	18.6 %	22.6 %
Mixed echogenicity (mixed)	14.3%	6.8 %
Ground glass cyst (g_glass)	24.0 %	12.4 %
Hemorrhagic cyst (haem)	1.0 %	0.8 %

Note: for continuous variables, mean±SD over the class of benign and malignant tumors respectively, are reported; for binary variables, the occurrences (%) of the corresponding features are reported.

- The models with linear kernels can be transformed into generalized additive models, which can be interpreted in the same way as a standard logistic regression model or a linear discriminant analysis (LDA) model.
- For a kernel-based model with nonlinear kernels (e.g. RBF kernels), we can make a compromise between the model complexity and the model interpretability, i.e. by choosing additive RBF kernels, such that the impact of each variable on the final outcome of the model can be isolated and visualized.
- Moreover, Bayesian LS-SVMs compute the posterior probability for a case being malignant based on the two Gaussian distributions of the latent outcome for the two classes. Normally the uncertainty resulting from parameter estimation should also be added to the variances of the two distributions. The posterior probability obtained after accounting for the uncertainty of parameter estimation is called marginal (or *moderated*) probability. The class probability without considering the uncertainty in parameter is denoted by raw class probability. However, this uncertainty in parameter is data dependent and not easy to visualize for high dimensional data. Thus we could only visualize of the raw class probability computed before accounting for the uncertainty in parameter estimation.

The various types (and their abbreviations) of the probabilistic models that were considered in this extensive study have been listed in Table 4.7, as well as the types of their corresponding latent and probabilistic outcome. It is indicated explicitly whether or not the probability output of the model has been integrated with the uncertainty in parameter estimation before model evaluation.

4.4.4 Prospective evaluation of the models built in the K.U. Leuven pilot project

In order to find out the answer of the aforementioned questions, we performed a series of experiments. Basically, we would use the variables selected from the previous study, and simply refit the model using the whole 425 available data in the pilot project from Leuven. However, previously for the reason of evaluation, we did not report any models based on the complete 425 data.

Input variables

We kept on using the variables selected from the previous study, particularly MODEL1, selected from the first 265 data in a forward procedure, which

Table 4.7: The types of models used in this study.

Model Type (Abbreviation)	Model full name Latent outcom interpretation Class probability output (raw / moderated)
LR	Logistic regression model Log odds of malignancy Raw class probability
RVM$_{\text{Lin}}$	Relevance vector machine classifier with linear kernels Log odds of malignancy Raw class probabilities
RVM$_{\text{RBF}}$	Relevance vector machine classifier with RBF kernels Log odds of malignancy Raw class probabilities
RVM$_{\text{RBFadd}}$	Relevance vector machine classifier with additive RBF kernels Log odds of malignancy Raw class probabilities
LSSVM$_{\text{Lin}}$	Bayesian LS-SVM classifier with linear kernels Assumed to follow two Gaussian densities over two classes Moderated class probabilities
LSSVM$_{\text{RBF}}$	Bayesian LS-SVM classifier with RBF kernels Assumed to follow two Gaussian densities over two classes, Moderated class probabilities
LSSVM$_{\text{RBFadd}}$	Bayesian LS-SVM classifier with RBF kernels Assumed to followed two Gaussian densities over two classes, Raw class probabilities

Note: Moderated class probability output refers to the derived posterior probability after considering the variances resulted from parameter estimation. Conversely, a raw class probability is computed without incorporating the variances from parameter estimation.

tried to maximize the posterior probability (evidence) of the Bayesian LS-SVM model with RBF kernels [59][58]. MODEL1 consists of the following 10 variables:

`l_ca125, PAPILLATION, sol, col3, BILATERAL, menopause, ASCITES, SHADOWS, col4, WALLREGULARITY.`

Model fitting

In general, the more training data, the better the models. The models will be built using all 425 patient data, which all contain a value for CA125 level. The considered structured models include logistic regression (LR) models, relevance vector machine (RVM) classifiers with linear kernels, Bayesian least squares support vector machine (LS-SVM) classifiers with linear and RBF kernels.

Results

The fitted models were evaluated in the following settings.

- The training set includes the 425 data collected at the University Hospitals Leuven, between 1994 and 1999. The test set consists of 1066 data collected in the IOTA project, which consists of the data from 9 different centers from various countries.
- When computing the posterior class probabilities for each test case, the default prior class probabilities were used, i.e. the priors for malignant and benign were set as the proportions of the malignant cases and the benign cases in the training set, at around 1/3 and 2/3, respectively. This is different from what we have done in the preliminary investigation, where we used instead the adjusted priors 2/3 and 1/3 for the malignant and benign classes in order to incorporate the different misclassification costs. In practice, it is however very difficult for clinicians to decide the relative misclassification cost in an accurate way. Therefore, in these experiments for evaluation, we simply kept the original priors without adjusting for the unbalance in the training data set and in the misclassification costs. In order to obtain a "good" classification, we then chose the probability cutoff values based on the training set. For each model, two specific probability cutoff values were set based on the training set. The first cutoff value was chosen to balance the sensitivity and specificity, i.e. to make the sensitivity close to the specificity as much as possible. The second cutoff was set in such a way that the sensitivity was the highest with an acceptable false positive rate (intuitively set to 20%).
- In order to use the model which includes a logarithm of CA125 value, we first fill in the missing log (CA125) with the mean value of the log(CA125) obtained from the 425 records in the training data set from Leuven (i.e. 3.7).
- Model performance is measured by area under the ROC curves (AUC).

The area under the ROC curve (AUC) and the corresponding standard error (SE) for various training and test sets have been presented in Table 4.8. The test set 1 consists of the 808 records with a CA125 value, which can be applied to the fitted model directly; test set 2 consists of the 258 records missing a value of CA125 level, and which has to be filled by a plausible value; whereas test set 1&2 is then the combination of the test set 1 and test set 2, which include all the 1066 IOTA data.

And Table 4.9 reports the performance of the models when being tested using data sets from various centers. This time only the data which have a value for CA125 level were included in the test sets.

Table 4.8: Area under the ROC curves for different models on the training and test sets. The models were built with variable subset MODEL1 based on 425 data in the K.U. Leuven pilot project, and tested on the data from the IOTA project.

	Training (N=425, 31.5% + from Leuven)		Test1 (N=808, 29.8% + IOTA with CA125)		Test2 (N=258, 9.7% + missing CA125)		Test1&2 (N=1066, 25.0% + Test1+Test2,IOTA	
Model Type	AUC	SE	AUC	SE	AUC	SE	AUC	SE
LR	0.9527	0.0131	0.9258	0.0121	0.8058	0.0540	0.9144	0.0122
RVM lin	0.9526	0.0131	0.9263	0.0120	0.8043	0.0541	0.9146	0.0122
RVM rbf	0.9513	0.0133	0.9277	0.0120	0.8024	0.0543	0.9157	0.0121
RVM rbf add	0.9538	0.0130	0.9262	0.0120	0.8067	0.0539	0.9147	0.0122
LSSVM lin	0.9522	0.0132	0.9274	0.0119	0.8026	0.0543	0.9155	0.0121
LSSVM rbf	0.9670	0.0110	0.9229	0.0123	0.7760	0.0564	0.9092	0.0125
LSSVM rbf add	0.9551	0.0128	0.9272	0.0119	0.8000	0.0545	0.9144	0.0122

Note: the missing log (CA125) value is replaced by the mean value in the training set, i.e. 3.7, the mean log (CA125) over the 425 records in the training data set from Leuven. "%+" refers to the proportion of the malignant tumors in a specific data set.

Table 4.9: AUC (SE) for different models on the test set in different centers. The models were built with variable subset MODEL1 based on 425 data in the K.U. Leuven pilot project, and tested on the data from the IOTA project. Here the cases with missing CA125 measurement have been excluded from the test set.

Center	MSW	RIT	MIT	OIT	NIT	LBE	BFR	MFR	KUK
#data	256	68	87	46	28	204	61	4	54
(%malignant)	(25.4)	(50.0)	(9.20)	(37.0)	(10.7)	(44.1)	(11.5)	(25.0)	(29.6)
Model Type									
LR	0.8901	0.9040	0.9873	0.9209	0.9600	0.9591	0.9735	1	0.8676
	(0.0276)	(0.0383)	(0.0281)	(0.0480)	(0.0811)	(0.0151)	(0.0431)		(0.0614)
RVM lin	0.8921	0.9092	0.9905	0.9249	0.9467	0.9580	0.9709	1	0.8775
	(0.0274)	(0.0373)	(0.0244)	(0.0468)	(0.0927)	(0.0153)	(0.0452)		(0.0594)
RVM rbf	0.8943	0.9118	0.9905	0.9270	0.9733	0.9569	0.9762	1	0.8791
	(0.0271)	(0.0368)	(0.0244)	(0.0462)	(0.0668)	(0.0155)	(0.0410)		(0.0590)
RVM rbf add	0.8913	0.9066	0.9889	0.9229	0.9467	0.9574	0.9762	1	0.8775
	(0.0275)	(0.0378)	(0.0263)	(0.0474)	(0.0927)	(0.0154)	(0.0410)		(0.0594)
LSSVM lin	0.8940	0.9022	0.9905	0.9270	0.9600	0.9593	0.9788	1	0.8824
	(0.0272)	(0.0387)	(0.0244)	(0.0462)	(0.0811)	(0.0151)	(0.0387)		(0.0583)
LSSVM rbf	0.8856	0.9135	0.9937	0.9209	0.9467	0.9514	0.9656	1	0.8528
	(0.0281)	(0.0364)	(0.0199)	(0.0480)	(0.0927)	(0.0165)	(0.0489)		(0.0642)
LSSVM rbf add	0.8966	0.9100	0.9889	0.9128	0.9600	0.9568	0.9709	1	0.8824
	(0.0269)	(0.0371)	(0.0263)	(0.0503)	(0.0811)	(0.0155)	(0.0452)		(0.0583)

Remarks

We now give some remarks on the results of the prospective evaluation of the models built from the pilot project.

- Let us first consider the case when the CA125 measurements are available. All the linear or nonlinear models generalized quite well when tested prospectively within the same center in Leuven. The training and test performance both reached an AUC over 0.95, though the linear models generalized slightly better than the nonlinear models. Performance degradation can be observed when tested prospectively across various centers, i.e. the AUCs decreased to the level of 0.92.

  The models performed the best in the centers of MIT, NIT, BFR and LBE, achieving AUC of > 0.95 for all the models. Although the AUC value for MFR is 1, which is almost perfect, only 4 patients are considered in this center (and 1 with malignancy), therefore the performance of these models when applied in this center still needs further examination.

  We would conclude that there are no significant differences among these models in AUC for test performance. It seems that linear models perform sufficiently well for these data. For use in the center of Leuven, the performance on the test set suggests that the linear Bayesian LS-SVM or the LR model is preferred.

- It is worth mentioning that, for Bayesian LS-SVMs, the overall performance (in AUC) using raw class probability computation are usually slightly different from using the marginal class probability after accounting for the parameter uncertainty. This can be understood in the following way: when increasing the variances of the latent outcome distributions, the posterior probability will shrink towards a certain cut-off probability that results in a balance in the sensitivity and specificity (i.e. in the situation when the sensitivity is very close to the specificity), which might lead to lower or higher AUC values. However, the accuracy, sensitivity and specificity will not change when the cutoff value is set at that specific cutoff point. Nevertheless, taking into account the uncertainty due to parameter estimation should still be considered important. For example, when a new case is presented to the model, which is very different from the cases in the training set, the raw class probability might give an over confident value. In contrast, the moderated class probability will tell us that the model is less confident in its prediction on this case, which might need further examination. The best practice will be to provide both the raw and moderated probability to the decision makers.

- We checked the performance of these models on the 258 record of the IOTA database whose missing values of CA125 were replaced by the mean value of the training set. The AUCs of all these models went down

to the level of 0.80. Be aware that the patients who had no measurement of CA125 are also in general less risky to malignancy (there are only 10% malignant cases among the 258 patients compared to 30% among the 808 patients with CA125 values provided).

On the other hand, we have also developed models without using CA125 based on the 425 training data from the K.U. Leuven pilot project. These models without use of CA125 did not perform better than our models using CA125 in this case. Moreover, models without use of CA125 gave in general inferior performance on the 808 data with CA125 values. In conclusion, based on the variables given in the pilot project, it is better to use CA125 in the models. In cases where a CA125 value is not available, the value can be filled in by a most probable value (e.g. the mean in the training set) in order to use the model. However, one should be aware that the missing of CA125 is actually not random, which is also indicated by the small percentage of the malignant cases in the data set with missing CA125. Furthermore, use of the mean value here leads the bias towards benign tumors. Therefore, interpretation of the results from the data missing CA125 should be taken with caution.

Model interpretation and visualization

For standard logistic regression models, interpretation can be based on the model parameters, which can be translated in terms of the change of the log (odds) for one unit increase/decrease in the corresponding variable.

For RVM classifiers with linear kernels, we can also transform the model from the kernel representation into a linear combination of the original input variables, and then the parameters can be interpreted in the same way as the standard LR models.

The LR model based on 10 variables using 425 training data from the K.U. Leuven pilot project is the following.

$$\log(P_{\text{malignant}}/P_{\text{benign}}) = -5.4737 + 0.4224 \times \texttt{l_ca125} + 1.6415 \times \texttt{PAPILLATION} + 2.0271 \times \texttt{sol} + 0.9324 \times \texttt{col3} + 2.0212 \times \texttt{BILATERAL} + 1.7624 \times \texttt{menopause} - 0.1510 \times \texttt{ASCITES} - 0.9972 \times \texttt{SHADOW} + 2.3356 \times \texttt{col4} - 0.6469 \times \texttt{WALLIRREGULARITY}$$

The RVM classifier with linear kernels can be transformed and expressed in the following formula, and thus can be viewed as a regularized logistic regression model:

$$\log(P_{\text{malignant}}/P_{\text{benign}}) = -5.4685 + 0.4835 \times \texttt{l_ca125} + 1.4729 \times \texttt{PAPILLATION} + 1.9768 \times \texttt{sol} + 1.0120 \times \texttt{col3} + 1.8359 \times \texttt{BILATERAL} +$$

$$1.5270 \times \texttt{menopause} - 0.3835 \times \texttt{ASCITES} - 0.9454 \times \texttt{SHADOW} + 2.0933 \times \texttt{col4} - 0.3025 \times \texttt{WALLIRREGULARITY}$$

For **the LS-SVM classifier with linear kernels**, its latent outcome can be written as a simple linear combination of original input variables, and can be viewed as a regularized linear discriminant analysis model:

$$\text{Latent outcome} = -1.5273 + 0.1272 \times \texttt{l_ca125} + 0.4948 \times \texttt{PAPILLATION} + 0.5155 \times \texttt{sol} + 0.2658 \times \texttt{col3} + 0.5638 \times \texttt{BILATERAL} + 0.4547 \times \texttt{menopause} - 0.0857 \times \texttt{ASCITES} - 0.3101 \times \texttt{SHADOW} + 0.5534 \times \texttt{col4} - 0.1736 \times \texttt{WALLIRREGULARITY}$$

For **kernel-based models with additive RBF kernels**, we can isolate the impact of each variable in the model outcome. See examples of visualizing the RVM classifier and Bayesian LS-SVM classifier with additive RBF kernels in Figure 4.10 and Figure 4.11, respectively.

In the first two rows of the figures, the solid line shows the relationship between the input and partial latent outcome (or log(odds) in case of RVMs) of the model for a certain variable. The final latent outcome for a tumor being malignant can be computed by summing all the partial latent outcome contributed from the individual variables. The intercept term in the original model has already been absorbed by adding to the partial latent outcome for each variable. The markers on the top and bottom of the sub plot for a single variable, represent the values of this variable in the training set for the positive and negative class, respectively. For continuous variables, they are denoted by "+"; for discrete categorical variables, the training data are represented by circles, and the size of the circle is proportional to the percentage of the data of a certain value within the positive or negative class.

The plots on the bottom row of the figures relate the latent outcome to a posterior probability for a case being benign or malignant. As seen in Figure 4.10, for RVM classifiers, the posterior probability for being malignant or benign can be obtained by checking the relationship between the log(odds) and the posterior probability. Whereas for Bayesian LS-SVM classifiers, the left bottom plot shows the two class conditional densities of the latent outcome of the LS-SVM model, which are assumed to be Gaussian. Depicted in the right bottom plot are the posterior class probabilities for a given latent outcome, where the class priors are set as the default value which are the percentage of the tumors of different types in the training set.

And we could also visually asses the importance of a certain variable within an additive model by checking the vertical span of the partial latent outcome (or log(odds)). The wider the span, the larger the influence of the variable to the model outcome. For example, in these two additive nonlinear models, the

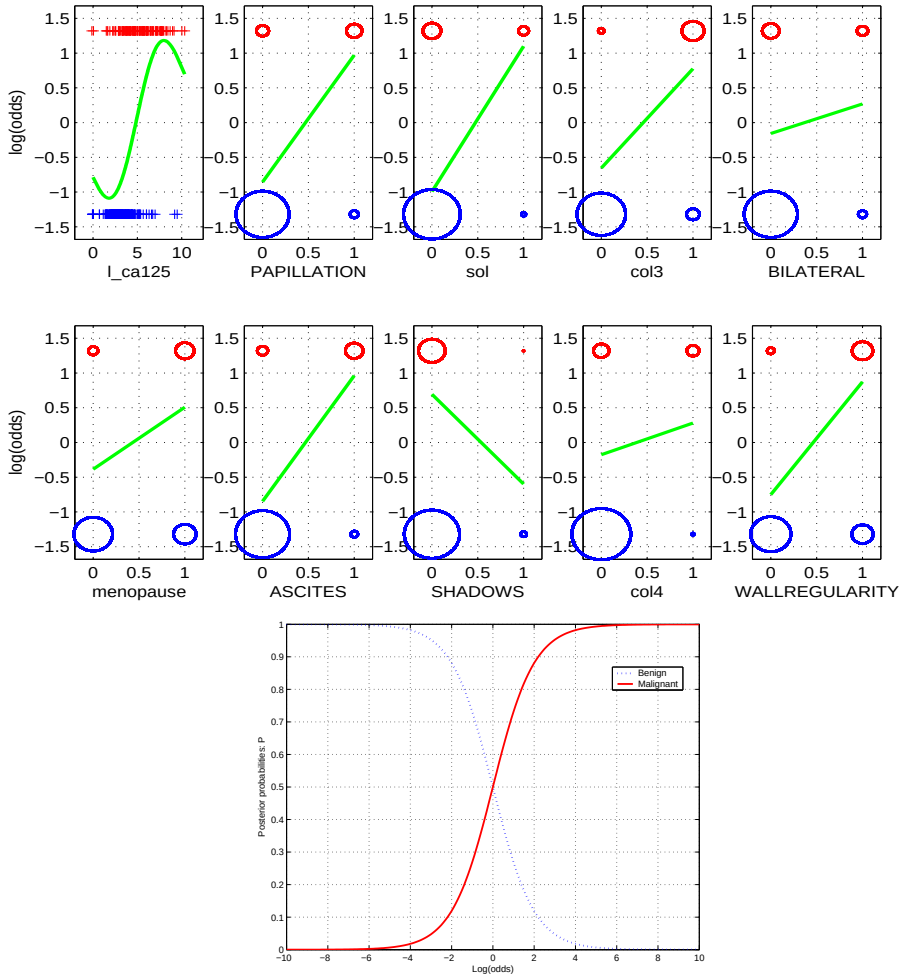


Figure 4.10: Visualizing the impact of each variable in an RVM classifier with additive RBF kernels. This model was developed using the 425 patient data from the K.U. Leuven pilot project.

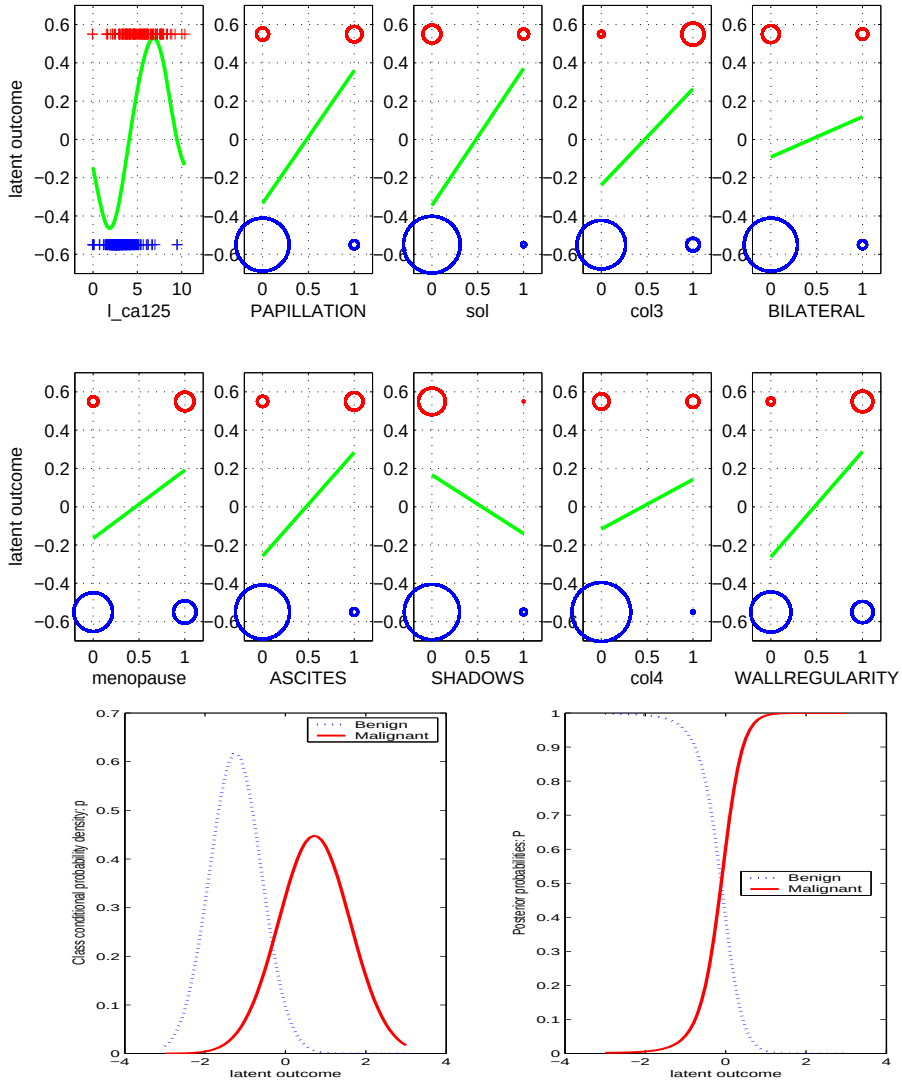


Figure 4.11: Visualizing the impact of each variable in the Bayesian LS-SVM classifier with additive RBF kernels. This model was developed using the 425 patient data from the K.U. Leuven pilot project.

log CA125 seems to be the most important one with the widest span in both figures. On the contrary, variable `BILATERAL` is the least important since the lines for partial latent outcome vs. Bilateral are quite flat in both figures.

4.4.5 Model development using the IOTA data set

Experimental settings

We used exactly the same training and test set as used in the paper [102]. There are 754 patient data in the training set and 312 in the test set, the splitting of the data set is randomized and stratified for centers and the types of tumors, i.e. the proportion of the malignant tumors and the percentage of each center are kept approximately the same for the training and test set.

Variables are selected based upon the training data. The models are then built using the 754 training data, and evaluated using the 312 test data. The performance of the models over different individual centers have also been evaluated. In prediction, the prior class probability are set again as the default values, which are about 3/4 for benign tumors and 1/4 for malignant ones according to the proportions of the malignancy in the training data set.

Input variable selection

The variable selection procedure was conducted in two scenarios: one with the consideration of possible use of CA125 levels (i.e. include variable `l_ca125` into the pool of variable candidates for further selection) and the other one without (i.e. exclude `l_ca125` from the candidate pool for further selection). CA125 serum test is a controversial test for detection of ovarian cancer, and this measurement is unavailable for 258 (24%) patients among the 1066 patients in this study. In order to see whether the inclusion of CA125 is necessary or not, the models built using the subsets with and without CA125 have been compared according to their performance on the common test set which have the measurement of CA125. The conclusion from this comparison implies that the inclusion of CA125 into the model did not give higher performance than the models using selected variables ignoring CA125, and thus not very crucial to be included into the model for prediction. Actually even before obtaining the test performance, the model evidence of the Bayesian LS-SVM has already indicated the preference to the models without using CA125 values. Therefore, in this thesis, only the models and the related results without using the value of CA125 are reported. In this case, the complete training and test set in total of 1066 patient data can be used for model development and evaluation. From now on, the report about the variable selection and model building and evaluation are all under the circumstance that excludes the CA125 from the analysis. Again the forward selection using Bayesian LS-SVMs has been conducted.

MODELaa denotes the subset of 18 variables obtained from forward selection using Bayesian linear LS-SVMs. The input variables selected (in order of selection) are the following:

```
SolidDMax, ASCITES, PAPFLOW, SHADOWS, TAMX, AGE,
WALLREGULARITY, PAPILLATION, PAPNR, origin1, PMB, col4,
HORMTHERAPY, mulsol, OvDmax, FLUID, PERSHISTOVCA, PAIN.
```

The forward selection based on the Bayesian LS-SVM with RBF kernels has also been tried, and selected 19 variables. However, this subset of variables does not give a higher model evidence compared with the subset obtained simply using linear kernels. Thus MODELaa is in favor and considered for further use of model building.

To assess the selected variables, apart from checking the model performance retrospectively, we should also examine the clinical meaning and reliability of those variables. The 18 variables were selected out from 56 variables which include various sources of information, and many of the variables are highly correlated. Some variables are simply the discrete version of another continuous variable, some are the subjective or semi-subjective evaluation of the same physiology character, described also by another variable objectively acquired from a device. Moreover, our forward selection strategy is somehow greedy. Therefore a pruning procedure is still needed in order to have a more compact and reliable model. This will be done with the guidance of our knowledge about the variables and the correlations among them.

There are several clusters of variables which are highly correlated: 1) `col4` and `TAMX`, 2) `ASCITES` and `FLUID`, 3) `PAPFLOW`, `PAPILLATION` and `PAPNR`.

Differences in Doppler sensitivity between ultrasound systems may affect both color detection and the type of Doppler shift spectra recorded including `TAMX`, and differences in built in algorithms may affect the results of calculations of pulsatility index and blood flow velocities. Furthermore because tumor color score was found to be superior to other Doppler variables in distinguishing malignant from benign adnexal masses, and evaluation of the color content of tumor scans is less time consuming than measurement of blood flow velocities in tumor vessels, it would seem sufficient to characterize tumors by their tumor color score without recording Doppler shift spectra from the tumor vessels [109]. Therefore we would prefer to use color score and remove the more time consuming and less reliable variable `TAMX` from the model; even though the inclusion of `TAMX` could further improve the model performance in the test set. This is a great advantage, because it saves time. It takes much more time to sample a tumor for its highest or lowest pulsatility index value or highest blood flow velocity than to evaluate the color content of a tumor scan. Nevertheless, the use of tumor color score has also several drawbacks since the tumor color

score is purely subjective and therefore "user-dependent."

Concerning FLUID vs ASCITES, actually only patient with presence of ascites could have the measure of FLUID. It seems the measure of the amount of fluid is however not very reliable. Therefore it is decided to remove variable FLUID from MODELaa.

As to the papillation related variables, we know that PAPFLOW could be present only if PAPILLATION is present. And PAPILLATION is actually the dichotomized version of PAPNR. Consequently, PAPILLATION becomes redundant after PAPNR entered the model as well. Hence variable PAPILLATION can be removed from the model.

Concerning variable PMB, presence of postmenopausal bleeding, it makes sense only among postmenopausal patient. A detail univariate analysis show that there was no significant difference in PMB between benign and malignant ovarian tumors [102]. Although after filling the missing value for premenopausal patients, PMB become significant in distinguishing between benign and malignant tumors, this could be due to the fact that, postmenopausal patients have higher risk to malignancy compared with premenopausal patients. Hence this variable is also eliminated from the model.

After removing the 4 variables purposely, 14 variables remain in the model. Then a backward elimination procedure was performed guided by the model evidence of Bayesian linear LS-SVM. This time variable PAPNR and PAIN were removed. Finally, 12 variables remain in the final subset, which is referred to as MODELa. The subset MODELa is more compact, more economic and reliable than MODELaa from the clinical point of view. The subset MODELa consists of the following 12 variables:

```
SolidDMax, ASCITES, PAPFLOW, SHADOWS, AGE, WALLREGULARITY,
origin1, col4, HORMTHERAPY, mulsol, OvDmax, PERSHISTOVCA.
```

As done in the study of data set from the pilot project, a subset of variables selected based on stepwise logistic regression was taken as a baseline variable subset, which is denoted as MODELb [102]. Subset MODELb includes the following 12 variables:

```
PERSHISTOVCA, HORMTHERAPY, AGE, LesDmax, PAIN, ASCITES,
PAPFLOW, sol, SolidDMax, WALLREGULARITY, SHADOWS, COLSCORE.
```

Quite some variables have been selected in common by the both of two selection methods, as shown in Table 4.10. And some variables, although selected differently by the two methods, are indeed highly correlated. For example,

Table 4.10: The variable subsets selected from different procedures. The selected variables within each subset have been marked with bullets. MODELaa, MODELa and MODELb were selected based on the the IOTA data set. MODEL1 was selected based on the K.U. Leuven pilot project. In the column for MODEL1, the variables that were not available in the pilot project have been crossed out and indicated by hyphens.

Variable (Symbol)	MODELaa	MODELa	MODELb	MODEL1
Medical history				
Age (Age)	•	•	•	
Personal history of ovarian cancer(PERHISTOVCA)	•	•	•	–
Postmenopausal (Meno)				•
Hormonal therapy (HORMTHERAPY)	•	•	•	–
Pelvic pain during the scan (PAIN)	•		•	–
Presence of postmenopausal bleeding (PMB)	•			–
∼ Serum marker ∼				
CA 125 (log)(L_CA125)				•
∼ Color Doppler Imaging (CDI) ∼				
Color score (0,1-4) (COLORSCORE)			•	
Weak blood flow (col2)				
Normal blood flow (col3)				•
Strong blood flow (col4)	•	•		•
Pulsatility index (PI)				
Resistance index (RI)				
Peak systolic velocity (PSV)				
(Time-averaged)Mean velocity (TAMX)	•			
∼ B-Mode Ultrasonography ∼				
Presence of abdominal fluid (Ascites)	•	•	•	•
Fluid in the pouch of Douglas (FLUID)	•			–
Tumor ovarian origin (origin1)	•	•		–
Ovary size in mm (OvDmax)	•	•		–
Lesion size in mm (LesDmax)			•	–
Unilocular cyst (uni)				
Unilocular solid (unisol)				
Multilocular cyst (mul)				
Multilocular solid (mulsol)	•	•		
Solid tumor (sol)			•	•
Solid component size in mm (SolidDMax)	•	•	•	–
Number of locules (NRLOCULES)				–
∼ Mophologic ∼				
Number of papillary projections (PAPNR)	•			–
Presence of papillary projections (PAPILLATION)	•			•
Presence of flow within papillary projections (PAPFLOW)	•	•	•	–
Bilateral mass (Bilateral)				•
Smooth wall (Smooth)				
Irregular wall (WALLREGULARITY)	•	•	•	•
Septa> 3mm (Sept)				
Acoustic shadows(Shadows)	•	•	•	•

MODELa picks up the size of the ovary (`OvDmax`) and MODELb picks up the size of the lesion (`LesDmax`). However, in many cases these two variables are just equal or very similar to each other.

Evaluation of model performance

The performance of the models using three different variable subsets is shown in Figure 4.12, where the AUCs both on the training data set and the test data set have been plotted. Similar to the preliminary study, there are no significant differences in test AUC between the linear and nonlinear models. Linear models performed as well as and sometimes even better than the nonlinear models (i.e. the kernel-based model with nonlinear kernels). MODELaa gave the best performance among the three subsets, with all the test AUCs for various models lying between 0.950 and 0.955, and with the AUCs on the training set very close to the ones obtained on the test set. Concerning the other two variable subsets which are considered more clinically relevant and reliable, MODELa performed slightly better than MODELb, obtaining AUC above 0.945 for all the linear models. Again, as to the comparison between standard LR and RVM with linear kernels, RVM classifiers with linear kernels behave as a regularized LR model, yielding always a slightly better generalization performance than the LR models. Bayesian LS-SVMs with linear kernels performed similarly well as the linear RVMs.

To see more details on the performance of the models, the ROC curves for the different linear and nonlinear classifiers with different variable subsets have been plotted in Figure 4.13.

Two cutoff probability values have been selected for each model individually based on the classification performance on the training sets. The first one was selected so as to obtain a balance in the sensitivity and specificity, i.e. to have two values as close as possible. The second one was chosen to have a sensitivity value as high as possible provided the specificity is at an acceptable level (here was set as 0.80). The classification performance for the 7 considered models types using MODELa and MODELb has been presented in Table 4.11. Besides the AUCs and its standard errors (SE), the sensitivity, specificity, and the two predictive values have been reported for each selected cutoff probability values.

Performance within different centers

Next we analyze the performance of the models for each individual centers. Figure 4.14 depicts the highest AUC value and lowest AUC value given by the 7 different models for each variable subsets. More detail reports of the performance in AUC in different centers are presented in Table 4.12 for each individual model using MODELa, and in Table 4.13 for each model using MODELb.

Table 4.11: Comparison of the validation performance on the **test set** ($N_{\text{train}} = 754, N_{\text{test}} = 312$) for various models. The IOTA data set was randomly split into the training set and test Set.

Model Type	AUC $\pm SE$	Cutoff probability	Accuracy (%)	Sensitivity (%)	Specificity (%)	PPV (%)	NPV (%)
LRa	0.9461	0.30	89.42	85.33	90.72	74.42	95.13
	±0.0185	0.15	84.29	92.00	81.86	61.61	97.00
RVMa_{Lin}	0.9485	0.30	88.78	89.33	88.61	71.28	96.33
	±0.0181	0.2	84.62	92.00	82.28	62.16	97.01
RVMa_{RBF}	0.9463	0.30	**89.74**	**88.00**	**90.30**	74.16	95.96
	±0.0185	0.15	83.33	92.00	80.59	60.00	96.95
RVMa_{RBFadd}	0.9462	0.25	87.82	84.00	89.03	70.79	94.62
	±0.0185	0.15	**84.29**	**93.33**	**81.43**	61.40	97.47
LS-SVMa_{Lin}	0.9462	0.25	88.46	85.33	89.45	71.91	95.07
	±0.0185	0.15	85.90	90.67	84.39	64.76	96.62
LS-SVMa_{RBF}	0.9452	0.24	87.82	82.67	89.45	71.26	94.22
	±0.0187	0.12	83.97	93.33	81.01	60.87	97.46
LS-SVMa_{RBFadd}	0.9430	0.24	88.78	84.00	90.30	73.26	94.69
	±0.0190	0.12	84.29	90.67	82.28	61.82	96.53
LRb	0.9423	0.25	85.26	82.67	86.08	65.26	94.01
	±0.0191	0.15	83.65	92.00	81.01	60.53	96.97
RVMb_{Lin}	0.9434	0.26	86.22	84.00	86.92	67.02	94.50
	±0.0180	0.16	**83.33**	**93.33**	**80.17**	59.83	97.44
RVMb_{RBF}	0.9426	0.25	86.22	86.67	86.08	66.33	95.33
	±0.0191	0.15	82.05	89.33	79.75	58.26	95.94
RVMb_{RBFadd}	0.9447	0.25	**86.86**	**84.00**	**87.76**	68.48	94.55
	±0.0187	0.15	83.01	92.00	80.17	59.48	96.94
LS-SVMb_{Lin}	0.9417	0.22	84.94	85.33	84.81	64.00	94.81
	±0.0192	0.14	83.65	90.67	81.43	60.71	96.50
LS-SVMb_{RBF}	0.9350	0.18	86.22	82.67	87.34	67.39	94.09
	±0.0202	0.10	83.01	89.33	81.01	58.82	96.00
LS-SVMb_{RBFadd}	0.9369	0.22	85.90	85.33	86.08	65.98	94.88
	±0.0199	0.11	83.33	90.67	81.01	60.18	96.48

Note: the first part of the results were obtained by the models using MODELa, and the second part were obtained by the models using MODELb. The letter "a" or "b" is inserted to the name of model type, in order to indicate the use of subset MODELa or MODELb, resepctively. The 'best' results obtained at a certain decision level are indicated in bold. The two cutoff values for each variables are determined based on the performance on the training set.

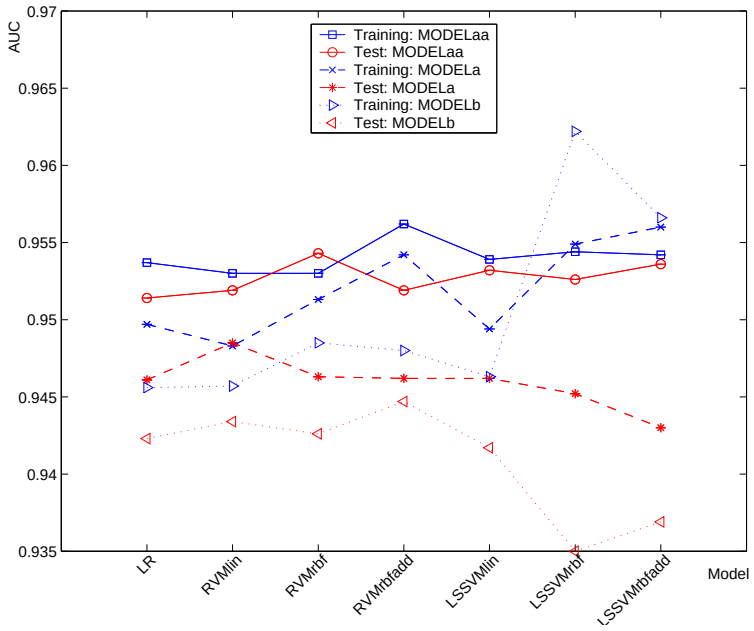


Figure 4.12: Training and test AUC for different models with different variable subsets. MODELaa contains 18 variables automatically selected by forward Bayesian linear LS-SVM. MODELa consists of the 12 variables obtained by pruning some highly correlated variables or clinically unreliable variables from MODELaa. MODELb contains 12 variables selected via a semi-automatic procedure involving stepwise logistic regression.

For some centers in which only a small number of patient data (say less than 20) are available, a considerable gap between the best and worst test set performance can be observed, this is due to the large variances associated with the AUC estimation and validation. However, the two centers MSW (from Sweden) and LBE (from Leuven, Belgium) which have contributed the largest amount of patient data in this IOTA project, both obtained a very high performance on the test set with relatively smaller variance over the different models. The best performing models for center MSW using MODELa and MODELb gave AUCs of 0.981 and 0.962, respectively. The best performing models for center LBE using MODELa and MODELb gave AUCs of 0.961 and 0.965, respectively. This time, all the models obtained a perfect performance with an AUC of 1 in center BFR.

Note that since center NIT has no malignant tumors in the test set, the evaluation of the performance on this center in AUC is not available. However, by checking the posterior probability output of the models on the 8 test data

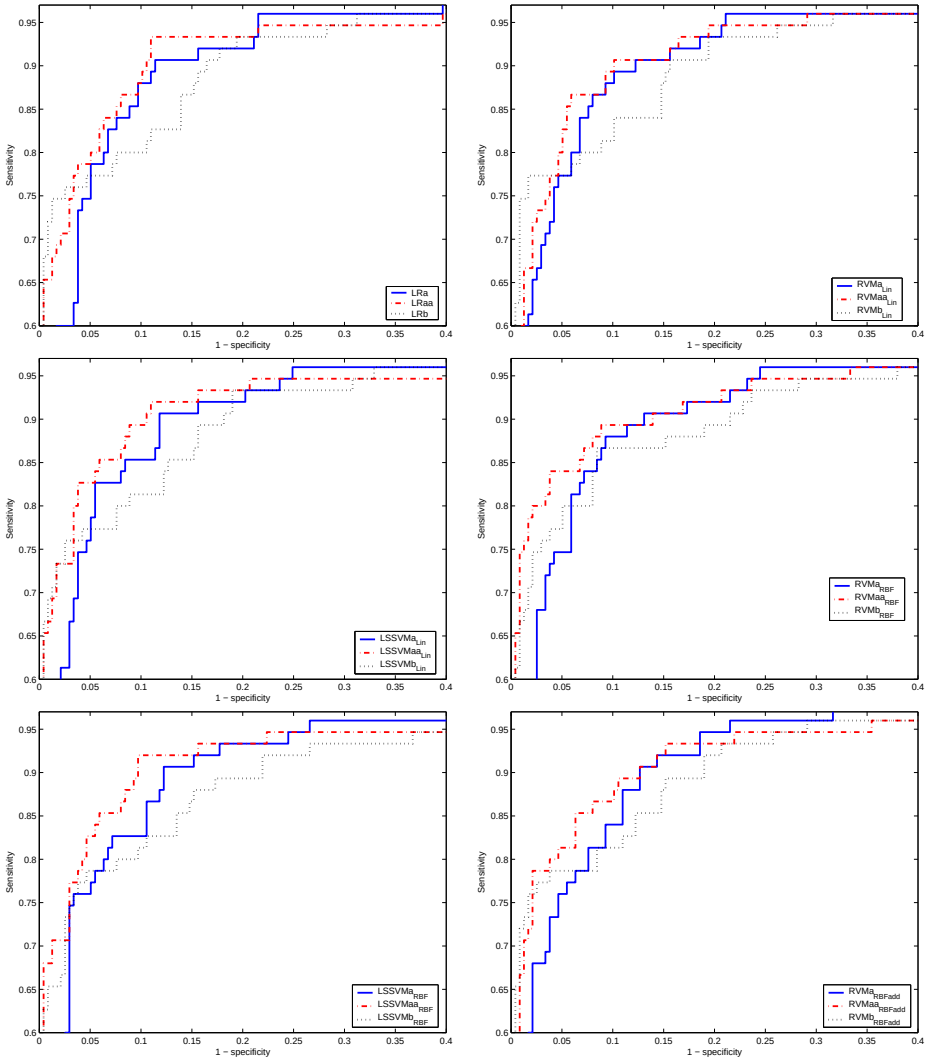


Figure 4.13: ROC curves of different models using various variable subsets on the 312 test data from the IOTA project. The variable subset used is indicated by adding the letters "a", "aa" or "b" to the model types in the legend, which correspond to the use of subset MODELa, MODELaa and MODELb, resepctively.

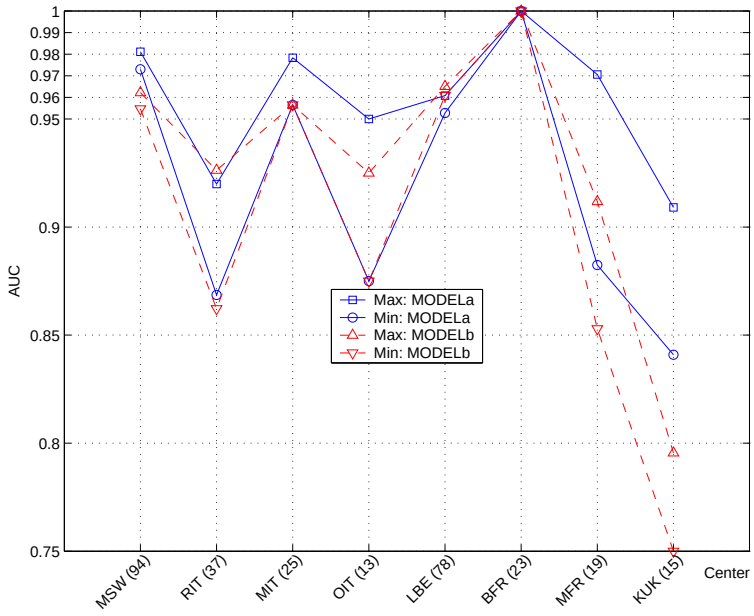


Figure 4.14: AUC for different models with different variable subsets when tested in different centers. The max/min AUC refers to the maximum/minimum test AUC among the 7 compared models using a certain variable subset. The number of data points of the test set for each center is indicated by the number besides the center label on X-axis.

for center NIT, we could know that the accuracy for this center is around 0.75 for every model when the cutoff value is set as 0.30, i.e. 2 out of 8 cases were misclassified as malignant ones.

Model visualization and interpretation

As done in Section 4.4.4, transformation of the kernel-based classifier with linear kernels have been performed in order to understand the behavior of these models. We could also visualize the additive relationship of the input variables and the partial latent outcome of the kernel-based classifier with additive kernels. Here several kernel-based classifiers with the highest test AUCs have been given a visualization. Figure 4.15 visualizes the RVM classifier with linear kernels using MODELa, and Figure 4.16 shows the impact of the individual variables in the model outcome for the linear Bayesian LS-SVM using MODELa. It can be seen that these two linear kernel-based models are very similar to each other in the relationship between the individual variable and the partial latent outcome except for the scale of latent outcome (which are related to the scale of the model parameters). Both models consider variable `SolidDmax`,

Table 4.12: Test AUC (SE) for different models in different centers. The models were built with variable subset MODELa based on 754 IOTA data and tested on 312 IOTA data.

Center	MSW	RIT	MIT	OIT	LBE	BFR	MFR	KUK
#data	94	37	25	13	78	23	19	15
(%malignant)	(21.3)	(35.1)	(8.0)	(38.5)	(34.6)	(8.7)	(10.5)	(26.7)
Model Type								
LR	0.9797	0.9167	0.9783	0.9000	0.9528	1.0000	0.8824	0.8409
	(0.0224)	(0.0562)	(0.0742)	(0.1026)	(0.0295)		(0.1621)	(0.1348)
RVM lin	0.9784	0.9199	0.9783	0.9500	0.9535	1.0000	0.9412	0.8409
	(0.0232)	(0.0552)	(0.0742)	(0.0739)	(0.0292)		(0.1198)	(0.1348)
RVM rbf	0.9797	0.9038	0.9783	0.9000	0.9535	1.0000	0.9412	0.8409
	(0.0224)	(0.0601)	(0.0742)	(0.1026)	(0.0292)		(0.1198)	(0.1348)
RVM rbf add	0.9791	0.8846	0.9565	0.9000	0.9564	1.0000	0.9706	0.8636
	(0.0228)	(0.0653)	(0.1033)	(0.1026)	(0.0283)		(0.0865)	(0.1267)
LSSVM lin	0.9804	0.9071	0.9783	0.9000	0.9542	1.0000	0.8824	0.8409
	(0.0221)	(0.0592)	(0.0742)	(0.1026)	(0.0290)		(0.1621)	(0.1348)
LSSVM rbf	0.9811	0.8974	0.9783	0.8750	0.9608	1.0000	0.8824	0.8409
	(0.0217)	(0.0619)	(0.0742)	(0.1135)	(0.0269)		(0.1621)	(0.1348)
LSSVM rbf add	0.9730	0.8686	0.9565	0.9250	0.9557	1.0000	0.9412	0.9091
	(0.0258)	(0.0692)	(0.1033)	(0.0897)	(0.0286)		(0.1198)	(0.1063)

Table 4.13: Test AUC (SE) for different models in different centers. The models were built with variable subset MODELb based on 754 IOTA data and tested on 312 IOTA data.

Center	MSW	RIT	MIT	OIT	LBE	BFR	MFR	KUK
#data	94	37	25	13	78	23	19	15
(%malignant)	(21.3)	(35.1)	(8.0)	(38.5)	(34.6)	(8.7)	(10.5)	(26.7)
Model Type								
LR	0.9547	0.9263	0.9565	0.9000	0.9651	1.0000	0.8529	0.7500
	(0.0331)	(0.0531)	(0.1033)	(0.1026)	(0.0254)		(0.1770)	(0.1584)
RVM lin	0.9561	0.9231	0.9565	0.9000	0.9637	1.0000	0.8824	0.7727
	(0.0326)	(0.0542)	(0.1033)	(0.1026)	(0.0259)		(0.1621)	(0.1536)
RVM rbf	0.9561	0.9199	0.9565	0.9000	0.9622	1.0000	0.9118	0.7955
	(0.0326)	(0.0552)	(0.1033)	(0.1026)	(0.0264)		(0.1436)	(0.1482)
RVM rbf add	0.9601	0.9231	0.9565	0.9000	0.9644	1.0000	0.8824	0.7727
	(0.0311)	(0.0542)	(0.1033)	(0.1026)	(0.0257)		(0.1621)	(0.1536)
LSSVM lin	0.9568	0.9231	0.9565	0.8750	0.9644	1.0000	0.8529	0.7500
	(0.0324)	(0.0542)	(0.1033)	(0.1135)	(0.0257)		(0.1770)	(0.1584)
LSSVM rbf	0.9622	0.8814	0.9565	0.8750	0.9608	1.0000	0.8824	0.7955
	(0.0304)	(0.0661)	(0.1033)	(0.1229)	(0.0269)		(0.1621)	(0.1482)
LSSVM rbf add	0.9561	0.8622	0.9565	0.9250	0.9622	1.0000	0.9118	0.7955
	(0.0326)	(0.0706)	(0.1033)	(0.0897)	(0.0264)		(0.1436)	(0.1482)

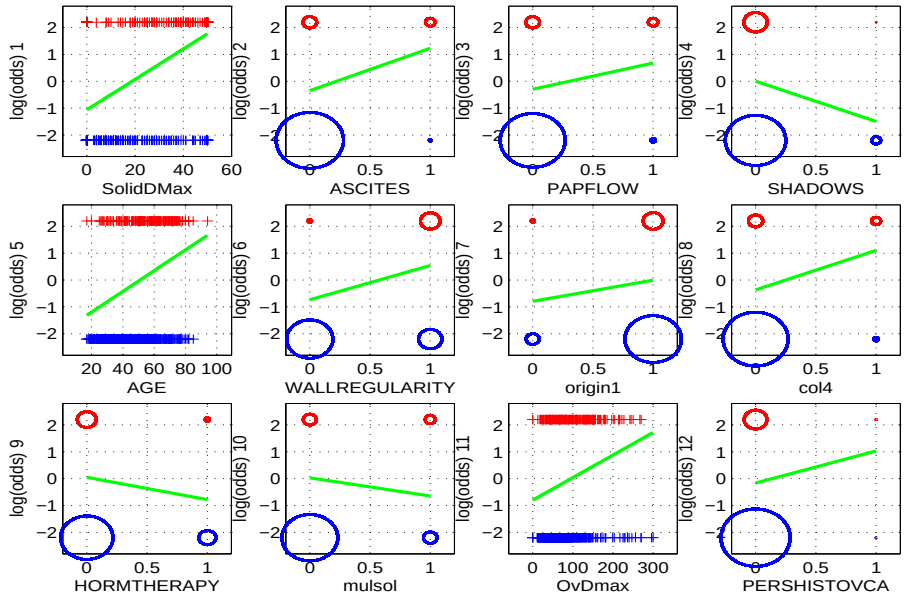


Figure 4.15: Visualization of the RVM classifier with linear kernels using 12 variables in MODELa, which was developed based on the training data set (N=754) from the IOTA project.

`AGE` and `OvDmax` very important in predicting the malignancy of ovarian tumors. The best performing classifier using MODELb is the RVM classifier with additive RBF kernels, which has been visualized in Figure 4.17. In this nonlinear additive model, only slightly nonlinear relationship can be observed for variables of `AGE`, LesDmax and `COLSCORE`.

We need to be careful not to over interpret the additive models when interactions between variables exist. For example, for the additive models using MODELa, from Figure 4.15, by observing the solid line for variable mulsol, we would be tend to say that the odds of having ovarian cancer is lower for a patient with presence of multilocule solid lesions (`mulsol`=1) than a patient absent of such lesions (`mulsol`=0). However, this is seemingly contradictory to the findings from the univariate descriptive statistics of variable `mulsol`, which says a higher risk to malignancy is associated with a tumor that is multilocule solid. This implies that interactions exist between `mulsol` and the other variables in the model. And it is found that the odds ratio of a malignant outcome for variable `col4` [2] is much lower (with value of 3.4) in case of

[2]The odds ratio of a malignant outcome for variable col4 is given by: the ratio of the odds of a malignant outcome in the group with presence of high blood flow (`col4`=1), to the odds of a malignant outcome in the group without (`col4`=0).

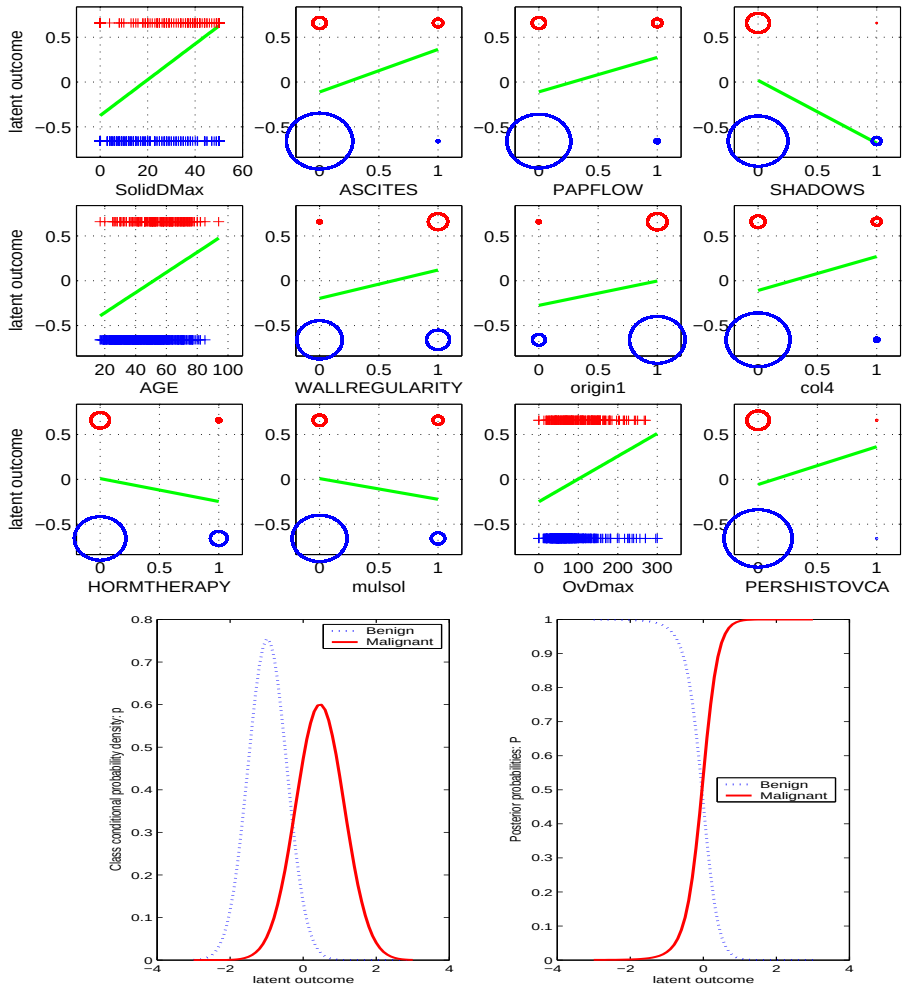


Figure 4.16: Visualization of the Bayesian LS-SVM classifier with linear kernels using 12 variables in MODELa, which was developed based on the training data set (N=754) from the IOTA project.

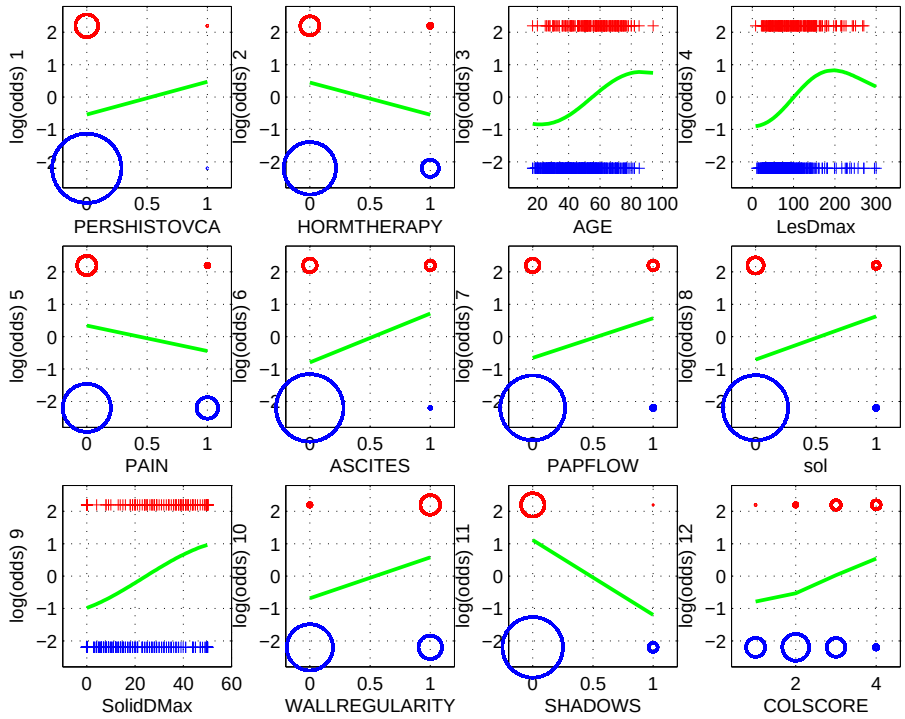


Figure 4.17: Visualization of RVM classifier with additive RBF kernels using 12 variables in MODELb, which was developed based on the training data set (N=754) from the IOTA project.

a multilocule solid lesion (`mulsol`=1) than the odds ratio (with value of 12.7) in other cases (`mulsol`=0). This interaction explains partially why the models have a negative coefficient for variable `mulsol` with the latent outcome of the model, though `mulsol` has a positive correlation with the tumor outcome in a univariate setting.

4.5 Discussion

In this section, we would like to discuss several issues related to the development and evaluation of our diagnostic models in clinical practice. Let us start with discussing the potential steps that might help in improving the performance of the mathematical models.

Incorporation of prior knowledge

One possible step is to incorporate the domain knowledge, which is abundantly owned by the experts, into the learning of the blackbox models. The quality of a purely data driven model depends on the quality and quantity of the training data. The representativeness of the training data is critical for the learning and generalization performance. For this ovarian tumor classification problem, a hybrid approach has been proposed and applied, which exploits the domain knowledge (represented in a belief network) in the learning of MLPs [5][28]. However, in case of a sufficiently large training data set, the advantage of incorporating the prior domain knowledge become less prominent. For example, as reported in [28], the Bayesian MLP classifiers trained on the IOTA data set obtained an average test AUC of 0.9448 by use of informative priors compared to an average AUC of 0.9444 by use of non-informative priors only[3]. And the informative priors were generated via using the pseudo-samples from the expert Bayesian belief network. That is, no significant improvement has been achieved so far by using the prior domain knowledge in this large IOTA data set. Further investigation of this hybrid approach and with application to other types of blackbox models such as LS-SVMs and RVMs are still needed.

On Bayesian modelling

Additionally, in Bayesian modelling itself, there are still some ingredients that might bring the improvement in performance. For example, other types of priors for the model parameters can be used. The posterior probability of the model (the marginalization) can be computed in a more accurate way, using for example MCMC sampling and variational methods [69, 8].

On ensembles of classifiers

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in a certain way, typically weighted or unweighted voting, to classify new examples. It has been discovered that, ensembles are often much more accurate than the individual classifiers that makes them up [23][14]. An ensemble can be more accurate than its component classifiers only if the individual classifiers disagree with one another, and the errors made from the individual classifiers should be somehow less correlated. In our case, we have already developed a couple of models based on the same large training set using the same or similar subsets of variables. Furthermore, the algorithms we employed are considered quite stable, unlike some unstable algorithms such as C4.5 for

[3] In those experiments, only data with CA125 values were used, and 11 variables including CA125 were selected in accordance of the expert Bayesian network, which was structured and parameterized by a medical expert. The data set was split randomly into two parts, 70% for training and 30% for test. And this hold out cross validation was repeated 1000 times to obtain an estimate of the final average performance.

decision tree learning. Hence, we did not expect a great improvement in performance by combining these individual models. Anyway, we still try to see whether there is any difference by use of ensembles. Several methods have been tried to combine the models using the IOTA data set. One is to simply take the unweighted average of the posterior probability from various models using variable subsets of either MODELa or MODELb. The combined model did not perform as well as the the best performing single model - the linear RVM using MODELa, resulting a test AUC of 0.9481 compared to 0.9485 from the single model. We have also tried to combined the latent outcome from various models by using a linear RVM (trained from a transformed data set using the model latent outcomes as input variables). This ensemble led to a test AUC of 0.9489, but only slightly better than the single best model. Therefore, we would conclude that combining the series of model built in this way (i.e., using similar subsets of variables and with the same training data) would not give a significant better performance over the best individual classifiers. However, other ways to generate individual models could be tried, for example, individual models can be created using data from different centers.

On interpretation of models

We have shown that the kernel-based models with additive kernels could be visualized and interpreted simply by isolating the impact of each variable to the latent outcome of the models. However, for models without additive effect for individual variables, how can we interpret them in one way or another? As shown in Chapter 3, for LS-SVM classifiers, we could make use of the primal-dual interpretations so as to see some insights in their blackbox decisions. The idea is to visualize the data and the decision boundary (obtained from the kPCR) in the coordinate system composed of pairs of PCs (in the feature space). To superimpose the explanatory variables on the graph, we could perform a kind of sensitivity analysis (i.e. to track the change of the latent outcome by varying only one variable while fixing the others), and plot the variable trajectories from a certain starting point. When nonadditive kernels are used in the model, it would be more interesting to see the trajectories obtained from various starting points.

Here we only show two biplots for Bayesian LS-SVM classifiers created from the 425 training data in the K.U. Leuven pilot project. Again, the variable trajectories (dashed lines) concurrent at the starting point (circle), which is the projection of the median input values in the training patterns. For more details on this type of biplot, please refer to the explanation for Figure 4.18. The 12-dimensional training data are projected to pairs of PCs in the kernel-induced feature space. The solid lines are the approximated decision boundary using only two PCs. The pseudosamples for a variable trajectory are bounded by the minimum and maximum value of the corresponding variable in the training set. And the end of the trajectory with the maximum value for a

variable is indicated by a positive sign together with a triangle. For the linear LS-SVMs, the variable trajectories can be viewed as variable axes. Due to the additive effect on the variables, one can also read out the position of a given new input pattern in the projected space, if the scales of the variable trajectories are accurately indicated. In the biplot for LS-SVM with RBF kernels, e.g., variable 9 (`col4`) and 1 (`l_ca125`) exhibit to be the most discriminative, while variable 8 (`SHADOWS`) seems contribute little to the distinction as it is almost parallel to the decision boundary. In general, interpretation of the Biplot based on PCR should be done cautiously since projections onto a few PCs do not show the complete picture of the data set, we need to make sure whether the variable could be expressed well by these PCs or not, and whether the PCs are discriminative or not, and more trajectories should be used from other starting points in order to discover the effects of the variable under different situations.

Dealing with missing values

Furthermore, one should be aware of the missing value problem in medical classification which occurs often in reality. For example the value of CA125 level is not always available, though it is the best known tumor marker till now. The CA125 blood test is not conducted for every patient due to various reasons in for example cost (the CA125 blood test is costly and time consuming). Thus we could build a series of models without considering this variable for such occasions. A more general solution to the incomplete data problem is *imputation*, i.e. to replace the missing value by a plausible value using, for example, the EM algorithm [21]. Multiple imputation [84] could also be applied. According to our preliminary test results, this could be a valuable tool to handle such problems in the cancer diagnosis.

Uncertainty in input variables

Another important issue in medical decision making is again related to the reasoning with the presence of uncertainty. This might influence the variable selection and model selection in general. Some variables (such as the level of CA125) are quite objective, while some are subjective or semi-subjective (such as color score), and some might be noisy. There is a confidence and reliability measure associated with each variable or measurement. The Bayesian approach we considered so far has taken into account the uncertainty in the model parameters and the output, however, it relies on the assumption that the input is known exactly. Further investigation is still needed on how to integrate the uncertainty being present in the input variables into the variable selection and prediction.

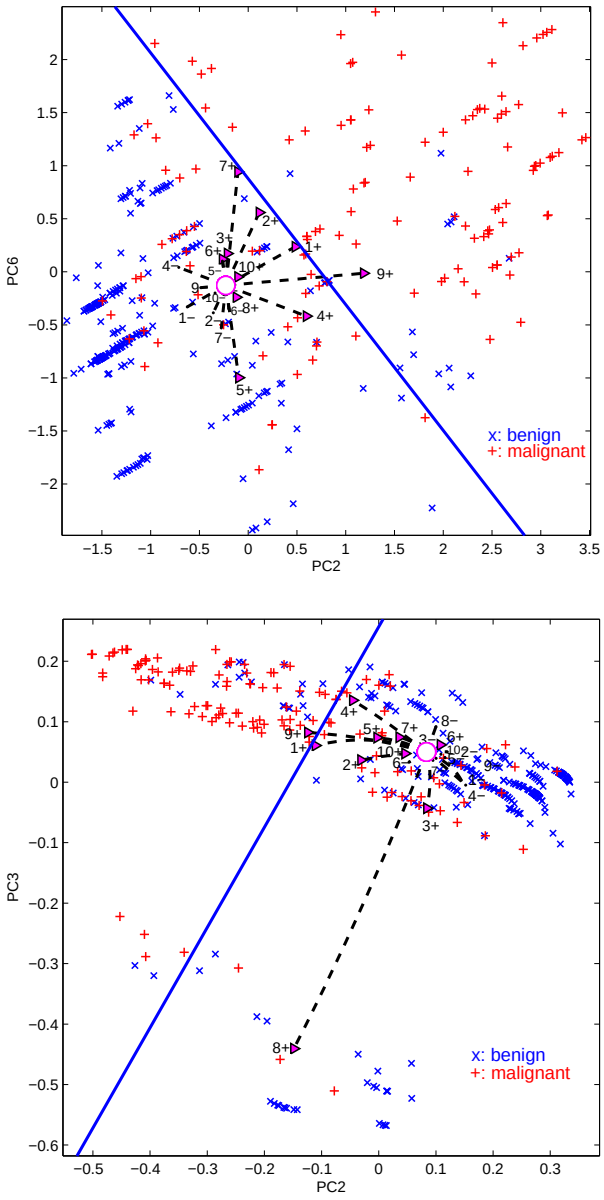


Figure 4.18: Visualization of LS-SVM with linear (upper) and RBF (bottom) kernels via kPCR on classification of ovarian tumors based on the 425 training data from the K.U. Leuven pilot project. The variable trajectories are distinguished by a number at the ends of the trajectories.

Data splitting in model evaluation

Also worth discussing is the way of data splitting for validating the diagnostic models. The splitting of the data set into a training and test set according to the time scale is more natural in clinical practice. There is a danger for changes in the patient population over time, however. For instance, as is the case in our initial study for the K.U. Leuven pilot project, the more experienced the expert, the more difficult cases are being referred to him for diagnosis, implying that the test set includes a higher number of harder cases (e.g., with borderline malignancy) to diagnose.

A homogeneity analysis of the group difference reveals that significant differences exist in age between the patient group examined from 1994 to 1997 and the new patient group examined after 1997. The patients of the old group are elder in average (mean age of 52.4) than the patients of the new group (mean age of 48.6). And not surprisingly the old group has a higher proportion of post-menopausal patients than in the new group. It is well known that the level of tumor marker CA125 can better predict the presence of cancer in post-menopausal patients, compared to that in pre-menopausal patients. This also implies that it is harder to predict correctly the malignancy of the tumors in the new patient group compared to that in the old patient group.

A random splitting of test and training set leads to a more equilibrated distribution of the patient data over both sets other than for a random variation, and is thus a weak procedure and less stringent [3]. This splitting is not representative for the way the models are used in clinical practice, where a prospective evaluation is normally needed. As is the case for our models developed from the IOTA data sets, a random splitting is utilized, therefore, a more strict prospective validation is still a must before the models can be applied in the clinical practice.

Multi-center study

As we see from the results in the initial study on the K.U. Leuven pilot project, the temporal validation performance of our Bayesian models is quite encouraging. The prospective test results for the single-center models are also very promising when tested on the multi-center IOTA data set. However, the validation performance is not yet perfect. Since the underlying mechanism of the disease is not yet clear, a statistically reliable model is only possible based on a sufficiently large amount of training examples. This is one motivation for the IOTA project, which has collected more than 1000 patient data from different centers in different countries, providing us a unique opportunity to develop more reliable models for ovarian cancer diagnosis. It can be shown that, the costly CA125 measurement is not a must for building a more accurate classifier. And linear models generally perform already sufficiently well in discriminating

between malignant and benign tumors. The models built on the multi-center data set performed quite well in 6 out of the 8 centers with test AUC above 0.95 in using variable subset MODELa. However, investigation is still needed in order to find out the reasons why the models performed worse in some centers than in the others.

Based on this large data set, mathematical models can be developed for not only preoperative classification of benign and malignant ovarian tumors, but also subclassification of the tumors between functional, non-functional benign, primary invasive and metastatic tumors. The variation between centers in outcomes of histology and the performance of the models will also be assessed. This will be the topics for the further research. Then another large amount of patient data, say around 1000, will be collected for future prospective validations.

4.6 Conclusions: towards robust ovarian cancer diagnostic models

In this chapter, we have discussed the application of various probabilistic models to preoperatively predict the malignancy of ovarian tumors. Particularly the LS-SVM and RVM classifiers with Bayesian learning have been advocated. Within the Bayesian evidence framework, the hyperparameter tuning, input variable selection and computation of posterior class probability can be conducted in a unified way. More flexibility has been given to the modelling by use of (either linear or nonlinear) kernels. Compared with RVMs, Bayesian LS-SVMs have the advantages in training speed and global optimization in its conventional training step. Though the model of RVM is usually very sparse and the relevant vectors are the prototypical data points instead of the ones close to the decision boundary. Unlike the two types of kernel based models, extra hyperparameters need to be tuned for the MLP modelling such as the structure of the neural network. Our results demonstrate that both the Bayesian LS-SVM models and RVM models have the potential to obtain a reliable preoperative distinction between benign and malignant ovarian tumors, and to assist the clinicians for making a correct diagnosis.

Now, we summarize the model development procedure for the ovarian cancer diagnosis, and spotlight some important issues that should be emphasized during the data modelling.

First of all, medical classification is an integral part of the medical decision making process, where one should consider not only the classification accuracy, but also many other factors, such as the uncertainty in model selection and

parameter estimation, different misclassification cost and the variance in the distribution of the patient data with time and environment. Remember that the reasoning is done with the presence of uncertainty and the risk should be minimized when making a decision. In working towards a more robust medical diagnostic model, the following issues are worth examining.

First, instead of emphasizing the predicting accuracy only, other medical diagnostic measures should be considered, such as the use of ROC curves, sensitivity, specificity, positive predictive value, etc. Instead of just presenting the black and white classification output, probabilistic output values should be provided.

Second, there should be adequate preprocessing of the patient data. Data screening, explorative data analysis with some visualization is useful in getting insights of the data and choosing the modelling techniques. Selection of input variables could help in enhancement of the model performance and better understanding of the underlying medical problems.

Third, a control system should be used to decide on the best model to use for a certain case. We also cannot claim that just one of the models is the best diagnostic model for all the cases. But we may conclude that a certain class of models should be used for a certain type of patients. The different models can be combined into committee machines, e.g. by using the additional layer that combines the individual models [95].

As an art, model building and model selection deserve a lot of attention. The model structure, the hyperparameters such as the amount of regularization, and parameters for the kernels need to be tuned. The Bayesian approach, which handles the uncertainty in modelling and prediction, is advocated for this purpose.

It is remarkable that different criteria exist for the academic world versus the clinical world. Also, to build a good system we need sufficient data and experiments. A multi-center study is essential for collecting enough data and getting diversity in the patients. Clinically, more confidence can be gained if we test the models in a multi-center environment.

Chapter 5

Bagging linear sparse Bayesian learning models for variable selection in cancer diagnosis

In this chapter, we investigate variable selection and classification for biomedical data sets with a small sample size and a very high input dimension. The sequential sparse Bayesian learning methods with linear bases were used as the basic variable selection algorithm. The selected variables were then fed to the kernel-based probabilistic classifiers: Bayesian least squares support vector machines (LS-SVMs) and relevance vector machines (RVMs). We employed the bagging techniques both for variable selection and model building in order to improve the reliability of the selected variables and the predictive performance. This modelling strategy has been applied to real-life medical classification problems including two binary cancer diagnosis problems based on microarray data and a brain tumor multiclass classification problem using spectra acquired via magnetic resonance spectroscopy (MRS). Other variable selection methods exploiting support vector machine (SVM) based recursive feature elimination (RFE) variable ranking or Fisher's criterion have been compared as well. It is shown that the use of bagging can improve the reliability and stability of the variable selection and model prediction.

5.1 Introduction

Recent advances in technologies such as microarrays and magnetic resonance (MR) have facilitated the collection of genomic, proteomic and metabolic data that can be used for medical decision support. For example, DNA microarrays enable us to simultaneously monitor the expression of thousands of genes [33][2]. It is then possible to compare the overall differences in gene expression between normal and diseased cells. Magnetic resonance spectroscopy (MRS) [25] is able to provide detailed chemical information about the metabolites present in living tissue. In particular, *in vivo* proton MRS offers considerable potential for clinical applications, e.g. for brain tumor diagnosis [25][70].

A great deal of attention has been paid to class prediction in the context of such new diagnostic tools, particularly for cancer diagnosis. The challenge of classification using microarrays and MR spectra lies in: (1) the large number of input variables and a relatively small number of samples, (2) the presence of noise and artefacts.

Kernel-based methods are of particular interest for this task since they can deal with high dimensional data in nature and have been supported by statistical learning theory and good empirical results [37][116][32]. Despite the early success, the presence of a significant amount of irrelevant variables or measurement noise might hamper the performance and interpretation of predictive models. Variable selection (VS) is used to identify the variables (features) most relevant for classification. This is important for medical classification, it can have an impact not only on the accuracy and complexity of the classifiers, but also on the economics of data acquisition. Moreover it is helpful for understanding the underlying mechanisms of the disease.

As already briefly mentioned in Chapter 2, several statistical and computational approaches to variable selection exist for classifying such data. The first approach is variables ranking followed by a selection (or filtering) step, usually accompanied with cross-validation (CV), so as to determine the number of variables to use in the classifier. The ranking criteria could be based on some univariate methods such as correlation, t-statistics, and some goodness measures of the corresponding univariate models. Variables could also be ranked in a multivariate manner taking into account the interactions among variables which are not accessed by the univariate methods [49]. One popular method of this type is support vector machine (SVM) based recursive feature elimination (RFE) [37][32]. The second approach is the so-called wrapper approach, which searches for the optimal combination of variables according to some performance measures of the models [117][74][93]. The third approach is the embedded approach. It combines the two tasks of variable selection and model fitting into one optimization procedure. This approach is considered to be computationally more efficient than ranking methods and wrapper meth-

ods. The embedded SVM based algorithms typically reformulate the standard SVM optimization problem in order to select only a fixed number of variables. This can be done via imposing additional constraints and adopting another objective function such as a generalization bound [116]. Nevertheless, these methods usually require an additional step (typically of cross-validation) for choosing the predefined number of variables. In [50] the Bayesian automatic relevance determination (ARD) algorithms were exploited for linear models. This is another type of embedded methods, in which the number of selected variables can be determined automatically. However, these algorithms seem to be sensitive to a small permutation of the training set, rendering their results less reliable from a biological point of view. In this thesis, we explore a similar type of Bayesian ARD methods, and show how the reliability of the selected variables and classification performance can be improved by using bagging and feeding various bootstrap variables to various types of probabilistic models. Furthermore, by utilizing the sparse Bayesian learning with logistic functions, we do not need to tune nuisance hyperparameters such as the regularization parameter used in SVMs, or a priori known noise variance in regression models as used in [50].

5.2 Methods

We have utilized the fast sequential sparse Bayesian learning algorithm [105] based on linear logit model without use of kernels (i.e. the basis functions are taken as the input variables). Details about this sequential Bayesian learning algorithm has already been described in Section 3.3.2 of Chapter 3. However, due to the nature of the sequential (and somehow greedy) sparse Bayesian learning algorithm, multiple solutions exist, and the algorithm is sensitive to small perturbations of experimental conditions (e.g. the solutions may vary with even very small change in the training data). The problem of instability for such types of algorithms become more serious when applied to these metabolic and genomic data sets, which are typically characterized by the high-dimensionality and relatively small sample size. In order to stabilize the procedure, and reduce the bias resulted from the use of small training set, the very simple model ensemble method, bagging, has been exploited.

The basic probabilistic models we used to fit the training data are kernel-based models including relevance vector machines and Bayesian least squares support vector machines. The standard support vector machine classifiers have also been used for comparison. Since linear models (i.e. models with linear kernels) suffice for our problems at hand, we utilized only linear kernels here.

How can we stabilize the variable selection procedure, and reduce the bias resulted from the use of small training data set? Next we describe one approach

to accounting for the uncertainty in variable selection and improving the model performance, that is, the simple model ensemble method, bagging, over our model selection, model building and outcome prediction.

5.2.1 Bagging strategy

Bagging the selected variables and models

Bagging is a "bootstrap" ensemble method that generates individuals for its ensemble by training each classifier on a random redistribution of the training set [14]. Each classifier's training set is generated by randomly drawing, with replacement, the same number of examples as in the original training set. It is shown that the bootstrap mean is approximately a posterior average of a quantity of interest [40]. Suppose a model is fitted to our training set $\mathcal{D}$, obtaining the prediction $f(\mathbf{x})$ at input $\mathbf{x}$. This prediction could be the latent outcome of e.g. a standard SVM model, or the predicted class probability of a probabilistic model. Bootstrap aggregation or bagging averages this prediction over a collection of bootstrap samples, thereby reducing its variance. For each bootstrap sample $\mathcal{D}^{*b}, b = 1, 2, \cdots, B$, we fit the model, giving prediction $f^{*b}(\mathbf{x})$. The bagging estimate is defined by

$$f_{\text{bag}}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} f^{*b}(\mathbf{x}). \tag{5.1}$$

The final class label will be decided by thresholding the bootstrap estimate of the class probability or the latent outcome. Bagging can push a good but unstable procedure a significant step towards optimality, which has been evidenced both experimentally and theoretically [14][40].

Note that, the class proportion in the bootstrap samples was not controlled in the experiments. However, the prior class probabilities have been fixed for each bagged model based on the training set class distribution.

An alternative bagging strategy is to bag only the predicted class labels and the final prediction will be given by voting. However, a reliable estimate of the class probability is essential for medical diagnosis. The prediction averaging strategy tends to produce bagged estimates with lower variance, especially for small B. Therefore, the prediction averaging strategy is preferred and advocated here.

We now sketch our bagging strategy for variable selection and modelling. For a given training set, B bootstrap data are randomly generated with replacement. For each bootstrap training set, one subset of variables is selected via the SBL logit models with linear basis functions, followed by feeding these variables to a model of interest such as Bayesian LS-SVM. Then B subsets of variables are selected and B corresponding models are built based upon the B bootstrap

training data. Given input data $\mathbf{x}$, the class probability or the latent outcome for the bagged binary classifier will be the average of the B class probabilities or latent outcomes. We took $B = 30$ in our experiments.

Strategy for the multiclass classification problems

For multiclass classification problems, we reduce the K-class classification problem into $K(K-1)/2$ pairwise binary classification problems, which yield the conditional pairwise probability estimates for a given input $\mathbf{x}$. The conditional probabilities are then coupled to obtain the joint posterior probability for each class by using Hastie's method [41], as introduced in section 2.5.2. The final prediction of the class will be the one for which the highest joint probability is achieved. Accordingly the variables used should be the union of the $K(K-1)/2$ sets of variables that are used in the same number of binary classifiers. Bagging is applied to each binary classification individually. Only the mean predicted probabilities from the bagged binary classifiers are coupled in order to get the final joint posterior probability for the multiclass classification problems.

5.2.2 Compared methods

In order to see the potential performance gain of using our proposed methods, we also assessed the performance of some reference methods. We denote the proposed variable selection approach as "LinSBL+Bag", which bags the variables selected from the linear SBL logit models. Accordingly the model fitting and prediction will be "bagged" as well. Its counterpart method "LinSBL" forms a classifier using only one subset of variables selected from a single linear SBL logit model, which is based on the whole training set without bootstrap repetition. Other variable selection methods used for comparison include two typical variable ranking methods, in which the models are built on the N_{v} variables with the highest ranks.

One of these is the popular SVM based recursive feature elimination (RFE) method [37]. The idea of this method is to recursively eliminate the variable which contributes the least in the SVM model, and then rank the variables based on the reverse order of their elimination. The contribution of the mth variable is evaluated by means of the change in the cost function $\Delta\mathcal{J}_m$ caused by removing the mth variable. The definitions of the cost functions have been previously given in Section 2.4.3. When a linear kernel is used, $\Delta\mathcal{J}_m = w_m^2$ with w_m the corresponding weight in the linear SVM model: $w_m = \Sigma_{i=1}^{N} a_i x_{im} y_i$.

The variables can also be ranked based on Fisher's criterion [10], which is a measure of the correlation between the variables and the class labels. For a binary classification, the Fisher discriminant criterion for an individual variable is given by $(\mu_{m,+} - \mu_{m,-})^2/(\sigma_{m,-}^2 + \sigma_{m,+}^2)$, where $\mu_{m,+}$ and $\mu_{m,-}$ are the means of variable m within the positive and negative class, respectively, $\sigma_{m,+}$ and

$\sigma_{m,-}$ the standard deviations of the variable within each class. The larger the Fisher's criterion, the higher the ranking of the variable.

The number of selected variables N_{v} was tuned by means of 10-fold cross-validation using SVMs. A coarse-to-fine strategy was utilized to search for N_{v} within a range of possible values. The N_{v} with the lowest 10-fold CV error rate were selected, and the tie breaking rule is to choose the smallest number of variables. The ranking and the number of variables were determined by using the whole training set in each run of cross-validation. These two methods are denoted by "RFE+CV" and "Fisher+CV", respectively.

Note that no bagging was applied for these reference methods. Our preliminary experiments show that the effect in bagging the models is more prominent when the variables vary among different bootstrap data. However, it becomes too time consuming to bootstrap variable selection with the methods of "ranking + CV".

Concerning the modelling techniques, besides the advocated probabilistic models, we use the standard linear SVM classifier as baseline model. In this work, we fixed the regularization hyperparameter of SVM to 10^6, which is high enough to keep the training error low. Unlike the other probabilistic models, the final SVM classifiers do not generate naturally the probability output. Therefore, for the multiclass classification problem, the final predicted class labels were decided by voting using the pairwise binary SVM classification results.

5.3 Experiments

5.3.1 Experimental settings

The generalization performance of the models in conjunction with variable selection was evaluated by 30 runs of randomized cross-validation. We applied a full cross-validation, where the variable selection was conducted prior to each model fitting process for each realization of the training data. An incomplete cross-validation, i.e. a cross-validation after variable selection might lead to a serious underestimation of the prediction error [89].

The data set was divided into two parts by stratified random splitting. The training set was used for model building including variable selection and model fitting, while the remaining data were used for test purposes. The performance is measured by the mean accuracy (Acc) and the mean area under the ROC curve (AUC) [38] and the corresponding standard error (SE) of the mean. Our matlab programs used for these experiments were built upon several toolboxes, including the SparseBayes V1.0 [107] (with modifications) for the sequential sparse Bayesian learning, the Spider [115] for RFE and SVM modelling, and

LS-SVMlab [78] for Bayesian LS-SVM modelling.

5.3.2 Binary cancer classification based on microarray data

Two benchmark binary cancer classification problems based on DNA microarray data have been considered. The first problem aims to discriminate between two types of leukemia: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The dataset includes 72 samples (47 of AML and 25 of ALL) with 7129 gene expression values [1] [33][37]. The second problem aims to differentiate tumors from normal tissues using colon cancer data. The dataset contains information of 40 tumor and 22 normal colon tissues. Each tissue is associated with 2000 gene expression values [2] [2][37]. All microarray data have been normalized to zero mean and unit variance. Each realization of the training set contains 50 data points, the test set includes the rest of the data of size 22 and 12 in leukemia and colon cancer data, respectively.

The accuracy and the AUC for the leukemia and colon cancer classification problems are reported in Table 5.1 and Table 5.2, respectively. The mean number of selected variables ($\overline{N_{\mathrm{v}}}$) for each variable selection (VS) method within one trial is also given.

Notice that the accuracy measure varies with the change of class priors or the probability cutoff values (which was fixed to 0.5 in our binary classifications). The most prominent changes are the following. When the priors P_+ and P_- in leukemia cancer prediction were both set to 0.5 instead of their default values 0.34 and 0.66, the accuracy for BayLSSVM using all variables increased to $95 \pm 0.78\%$. In the colon cancer diagnosis, bagged BayLSSVMs and RVMs reached a higher accuracy of $86.11 \pm 1.68\%$ and $82.78 \pm 1.92\%$, if P_+ and P_- were set to 0.2 and 0.8 (instead of their default values 0.36 and 0.64), respectively. On the contrary, the AUC measure is independent from the prior class distribution. Although AUCs for the bagged models might change slightly due to the bagging effect, e.g. the mean AUC for the bagged BayLSSVM changed from 89.06% to 89.38% for colon data.

Additionally, the LinSBL logit model is itself a probabilistic model with an automatic variable selection mechanism. It achieved a similar performance as the probabilistic kernel models. For prediction of leukemia cancer, a single LinSBL model and a bagged LinSBL model yielded a test performance in terms of mean AUC of $94.55 \pm 1.05\%$ and $98.51 \pm 0.46\%$, respectively. Using the colon data, the test performance for a single LinSBL and a bagged LinSBL reached a mean AUC of $82.08 \pm 1.94\%$ and $89.27 \pm 1.72\%$, respectively.

[1] available online at www.genome.wi.mit.edu/MPR/data_set_ALL_AML.html
[2] available online at microarray.princeton.edu/oncology/affydata/index.html

Table 5.1: Test results for leukemia cancer classification, reported as mean±SE of the accuracy and AUC from 30 runs of cross-validation. All classifiers were tested with the same series of variable selection techniques, and utilized only linear kernels. The highest value of accuracy or AUC for each type of classifier (in row) is indicated in bold.

VS method	All		Fisher+CV		RFE+CV		LinSBL		LinSBL+Bag	
$\overline{N}_{\mathrm{v}}$	7129		43.30		5.10		3.50		49.10	
Classifier	Acc(%)	AUC(%)	Acc(%)	AUC(%)	Acc(%)	AUC(%)	Acc(%)	AUC	Acc(%)	AUC(%)
SVM	**95.61**	**99.29**	92.42	97.29	90.30	94.88	90.45	94.43	92.42	98.01
	±0.73	±0.23	±0.89	±0.64	±1.19	±1.17	±0.92	±0.94	±0.92	±0.46
BayLSSVM	89.85	**98.78**	92.73	97.62	90.00	95.01	88.94	93.50	**93.79**	98.48
	±1.05	±0.34	±1.04	±0.59	±1.29	±1.11	±1.02	±1.40	±1.02	±0.41
RVM	90.15	96.22	93.03	97.26	90.45	94.93	89.85	93.72	**93.18**	**98.24**
	±1.34	±0.77	±0.85	±0.61	±1.24	±1.10	±1.17	±1.17	±1.00	±0.47

Table 5.2: Test results for colon cancer classification, reported as mean±SE of the accuracy and AUC from 30 runs of cross-validation. All classifiers were tested with the same series of variable selection techniques, and utilized only linear kernels. The highest value of accuracy or AUC for each type of classifier (in row) is indicated in bold.

VS method	All		Fisher+CV		RFE+CV		LinSBL		LinSBL+Bag	
$\overline{N}_{\mathrm{v}}$	2000		114.30		9.73		6.50		107.17	
Classifier	Acc(%)	AUC(%)	Acc(%)	AUC(%)	Acc(%)	AUC(%)	Acc(%)	AUC(%)	Acc(%)	AUC(%)
SVM	81.94	85.31	80.28	85.83	81.36	83.54	76.39	81.67	**86.11**	**87.71**
	±1.80	±2.22	±2.10	±2.26	±2.18	±3.10	±2.03	±2.23	±1.73	±1.93
BayLSSVM	85.00	88.65	**85.28**	**89.27**	81.11	86.15	76.67	84.48	84.44	89.06
	±1.73	±1.71	±1.56	±1.60	±2.11	±2.79	±2.08	±2.09	±1.65	±1.78
RVM	83.61	87.71	**83.61**	86.98	81.67	86.46	73.06	82.40	79.17	**87.71**
	±1.82	±2.16	±1.73	±2.17	±2.30	±2.75	±2.41	±2.23	±1.99	±1.88

5.3.3 Classification of brain tumors based on MRS data

The method has also been applied to a multiclass classification problem of brain tumors using short echo time ^{1}H MRS data. The dataset consists of 205 spectra in the frequency domain. The full spectrum (a row vector of magnitude values) has been normalized to unit norm. Only the frequency region of interest from 4.17 to 0 ppm (a measure of the chemical shift in a field independent frequency scale) was used in this study, corresponding to 138 input variables. The dataset contains the records from four major types of brain tumors: meningiomas (Class 1, 57 spectra), astrocytomas grade II (Class 2, 22 spectra), glioblastomas (87 spectra) and metastases (39 spectra) [22]. However, the last two types of tumors are very difficult to distinguish. Our experience on this dataset is that, the trained models did not perform better than a majority classifier, which assigns the majority class in the training set to all the test cases. Therefore, we merged the two tumor types - glioblastomas and metastases - into one class of aggressive tumors (Class 3), and only dealt with the three-class classification problem. For details of the data acquisition and preprocessing procedure for this dataset, the readers are referred to [22][52].

Since the data are unbalanced, the model using the default priors will lead to a relatively low sensitivity for astrocytomas grade II. Therefore, we decided to use equal priors for all binary classifiers, which resulted into a "satisfactory" sensitivity and specificity for all three classes. The average test AUC for each pairwise binary classification is reported in Table 5.3. Table 5.4 presents both the training and test accuracy of the brain tumor classification problems using equal class priors.

Again the LinSBL model can be used to classify brain tumors. Using equal priors, the test performance for 3-class prediction reached a mean accuracy of $86.03 \pm 0.63\%$ by using a single LinSBL model, and $89.46 \pm 0.52\%$ by using a bagged LinSBL model.

Notice that, the mean number of variables selected by LinSBL+Bag for the 3-class classification problem is 98.73, which means that about 72% of the variables were involved in the classification. However, checking the number of variables selected for each pairwise binary classification provides more insight. For example the comparison of Class 1 vs. 2 used only an average of 31.17 variables.

5.3.4 Biological relevance of the selected variables

We examined the most frequently selected variables from the LinSBL+Bag and their biological relevance for each dataset. This was done by first calculating the number of occurrences o_{vm} of variable m being selected on the vth run of cross-validation, i.e. on the 30 bootstrap data of one training set realization.

Table 5.3: Test AUC (%) for pairwise binary classification of brain tumors, reported as mean±SE from 30 runs of cross-validation. All classifiers were tested with the same series of variable selection techniques, and utilized only linear kernels. The highest AUC of each binary classification for each type of model (in row) is indicated in bold.

VS method	All			Fisher+CV			RFE+CV			LinSBL			LinSBL+Bag		
Class pair	1vs.2	1vs.3	2vs.3	1vs.2	1vs.3	2vs.3	1vs.2	1vs.3	2vs.3	1vs.2	1vs.3	2vs.3	1vs.2	1vs.3	2vs.3
Classifier $\overline{N}_{\mathrm{v}}$	138	138	138	39.17	114.57	9.03	6.10	23.10	14.07	4.30	9.50	6.20	31.17	59.67	55.40
SVM	**99.82**	97.52	92.14	98.75	96.17	95.06	98.18	96.56	92.23	96.99	96.51	90.78	98.17	**97.76**	**96.01**
	±0.08	±0.26	±0.77	±0.41	±0.89	±0.69	±0.56	±0.37	±0.95	±0.62	±0.37	±1.07	±0.88	±0.28	±0.44
BayLSSVM	99.62	97.33	95.15	98.67	96.85	95.43	97.44	97.04	94.23	96.94	96.88	93.38	**99.65**	**97.95**	**96.44**
	±0.16	±0.27	±0.56	±0.39	±0.39	±0.63	±0.57	±0.30	±0.70	±0.57	±0.34	±0.82	±0.15	±0.26	±0.40
RVM	98.47	97.55	**96.87**	98.60	96.82	95.80	97.52	97.18	95.57	96.99	97.02	94.52	**99.70**	**97.94**	96.19
	±0.37	±0.27	±0.37	±0.38	±0.47	±0.65	±0.74	±0.32	±0.67	±0.72	±0.35	±0.70	±0.13	±0.26	±0.42

Table 5.4: Training and test accuracy for brain tumor three-class classification, reported as mean±SE from 30 runs of cross-validation. All classifiers were tested with the same series of variable selection techniques, and utilized only linear kernels. The highest value of training accuracy or test accuracy for each type of classifier (in row) is indicated in bold.

VS method	All		Fisher+CV		RFE+CV		LinSBL		LinSBL+Bag	
$\overline{N}_v$	138		115.97		37.37		17.9		98.73	
Classifier	Train(%)	Test(%)	Train(%)	Test(%)	Train(%)	Test(%)	Train(%)	Test(%)	Train(%)	Test(%)
SVM	**100.00**	85.25	95.72	85.54	**100.00**	85.05	99.95	83.92	96.40	**86.91**
	±0.00	±0.69	±1.07	±1.10	±0.00	±0.70	±0.05	±0.43	±0.30	±0.71
BayLSSVM	**99.25**	86.37	94.57	86.47	98.54	85.15	96.20	86.37	96.84	**89.51**
	±0.12	±0.75	±0.52	±0.66	±0.18	±0.66	±0.25	±0.79	±0.20	±0.55
RVM	89.17	87.72	89.49	87.11	94.55	87.11	94.60	86.72	**94.67**	**89.95**
	±0.31	±0.76	±0.34	±0.61	±0.43	±0.70	±0.38	±0.76	±0.19	±0.56

This number of occurrences of a variable can be used to rank the variables. The higher the number, the higher the rank for that variable. The variables that were selected only by chance, e.g. once or twice, should not be considered as important. For each binary classification, an occurrence matrix $\mathbf{O}$ was computed with element of $o_{vm}, v = 1, \cdots, 30$ and $m = 1, \cdots, d$. By plotting this matrix, we can have a visual assessment of the variables.

To facilitate the interpretation of the occurrence matrix, we ordered the variables and the occurrence matrix accordingly in descending order of the total number of selection occurrences ($\sum_{v=1}^{30} o_{mv}$) among the 30 trials. Then we plot the submatrix of $\mathbf{O}$ containing only the variables that had been selected at least 30 times in total. The results are shown in Figure 5.1 for the leukemia data, and in Figure 5.2 for the colon cancer data.

It is noteworthy that, the genes that were selected by the proposed bagged LinSBL logit models are mostly biologically interpretable. In the leukemia cancer classification, the three top ranked genes identified by our algorithm are all among the informative genes according to [33]. The highest ranked gene is zyxin (gene 4847 according to its order in the original dataset, accession no. X95735), which encodes a LIM domain protein localized at focal contacts in

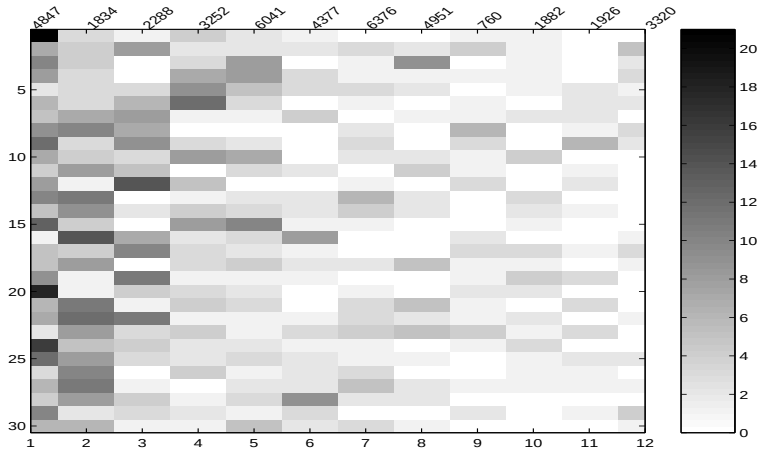


Figure 5.1: Genes selected by LinSBL+Bag from the 30 realizations of the training sets for leukemia cancer microarray data. The x-axis labels in the bottom the rank of the gene, and on the top the index of the gene in the original microarray data matrix. The y-axis refers to the number of runs in the 30 randomized cross-validations. Only the genes that were selected more than 30 times in all the 30×30=900 linear SBL models are listed in the plot. The gray level in each cell corresponds to the number of occurrences that a gene was selected in bootstrapping for one realization of the training set.

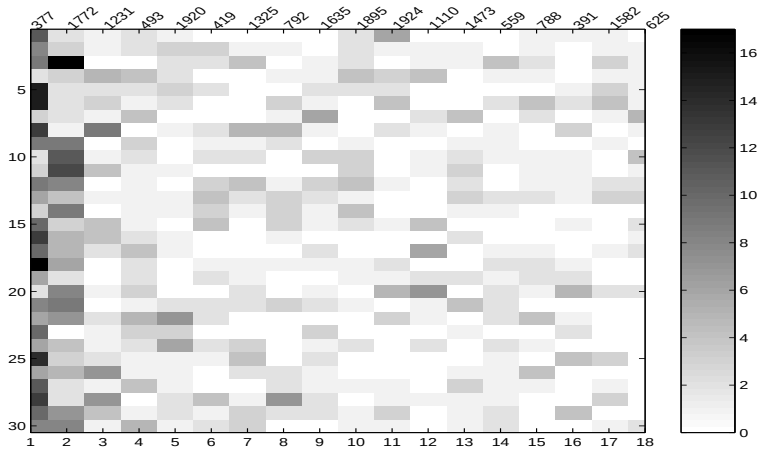


Figure 5.2: Genes selected by LinSBL+Bag from the 30 realizations of the training set for colon cancer microarray data. See Figure 5.1 for details.

adherent erythroleukemia cells. CD33 (gene 1834, M23197) is the differentiation antigen encoding cell surface protein, for which monoclonal antibodies have been demonstrated to be useful in distinguishing lymphoid from myeloid lineage cells.

In the colon cancer data, the most important gene which is identified by our method corresponds to mRNA for uroguanylin precursor (gene 377, accession no. Z50753). Guanylin and uroguanylin have been recently found to be linked to colon cancer, and treatment with uroguanylin was found to have possible therapeutic significance [50][88]. The gene with the second highest rank (Gene 1772, H08393) is a collagen alpha2 (XI) chain which is involved in cell adhesion, and collagen degrading activity is part of the metastatic process for colon carcinoma cells [37][47].

For the class prediction of brain tumors, we examined the corresponding metabolites of which the magnitude values appeared to be significant in the pairwise binary classification problems. Figure 5.3 depicts the mean spectra of the three classes, as well as the metabolites associated with their resonance frequencies [34]. The two horizontal bars, below each mean spectrum, represent the selection rate of each variable (corresponding to a frequency value) in pairwise discrimination with another tumor class. The selection rate for variable m was computed by dividing the total number of selection occurrences by $30 \times 30 = 900$. We can take the selection rate as a kind of importance measure for the variables.

By examination of the figure and incorporating the domain knowledge, we were able to point out the metabolites that are important or useful for the classifica-

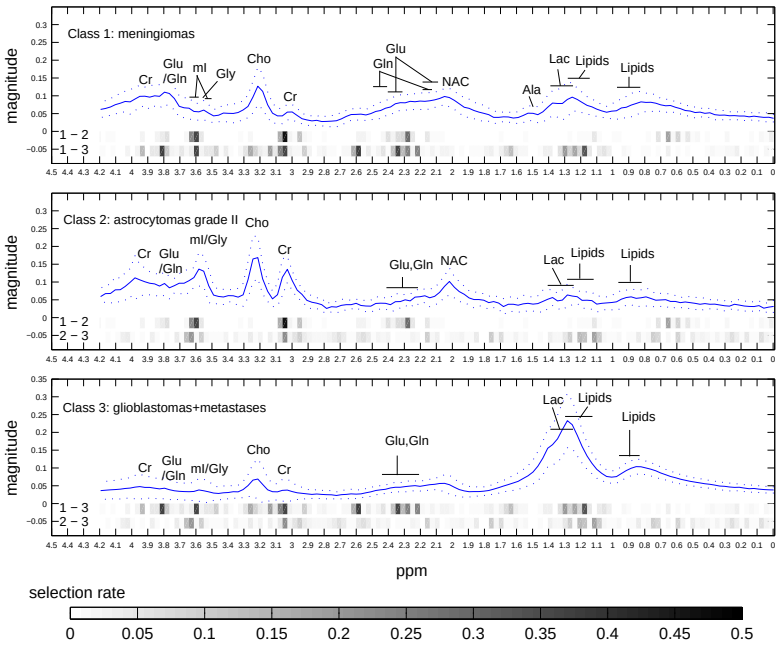


Figure 5.3: Mean spectrum of the brain tumors and the selection rate of the variables using LinSBL+Bag from the 30 runs of cross-validation for the pairwise binary classification. The dotted line is the mean ± SD (standard deviation) spectrum.

tion. For example, in contrast to the other two classes, the astrocytomas grade II have a relatively high level[3] in the frequency regions of both total creatine (Cr) and myo-inositol (mI)/ glycine (Gly). These variables were also selected most frequently in all three pairwise binary classification problems, particularly for differentiating Class 1 from 2. Indeed, in these regions the selection rate has the darkest color and reaches a value close to 0.5. To discriminate meningiomas from the aggressive class, more frequency regions are used: not only Cr and mI/Gly, but also Glutamate (Glu), Glutamine (Gln), Lipids, N-acetyl containing macromolecules (NAC). Interestingly, Cr does play a role in the maintenance of energy metabolism. While NAC resonances at the usual NAA (N-acetyl aspartate) chemical shift may appear in the solid or cystic areas of brain tumors. However one must be cautious when interpreting the selected variables for such MR spectra: the resonances at the same position may originate in different compounds depending on the tumor type. For example, the 2.03 ppm peak originates mostly from Lipids in Class 3 tumors, while it is safe

[3] The "level" here refers to relative intensity in the spectra as they are scaled to unit norm.

to label it NAC for Class 1 and 2. It may have NAA contribution, but other N-acetyl compounds are contributing varying amounts [15]. The whole region at 2-2.6 ppm may have variable contribution from macromolecules (mostly proteins).

5.4 Discussion

From both the binary and multiclass examples, we can clearly see how bagging improves the performance of the single model generated by simply selecting one subset of variables. There is a significant gap in predictive performance between the methods of LinSBL and LinSBL+Bag in our experiments. These two sets of test performance were compared via paired t-tests for each classifier. The p-values of comparison on AUC are all $< 10^{-4}$ for the leukemia data, and all < 0.015 for the colon data. The t-tests on accuracy for the 3-class brain tumor diagnosis got a p-value $< 10^{-4}$ for each model type. This gap might result from: on the one hand the large uncertainty because of the small sample size of the training data, and on the other hand the sensitivity of SBL itself. Even for the reference methods which are considered to be more stable than SBL, unreliability of the selected variables can still be observed from the big difference between the training and test performance.

As to the modelling techniques, the Bayesian probabilistic models performed somehow better than the standard SVMs. This might be partially due to the fact that the hyperparameters for SVMs (C, fixed to a high value of 10^6 to keep the training error low) were not optimized. Our main focus was on the models with probabilistic output which is important in biomedical diagnosis and without the burden of cross-validation for hyperparameter tuning. We have also tested the single model using the aggregated variables from the LinSBL+Bag selection, which led again to a worse performance compared to that of the bagged models with different variables.

The class priors or the cutoff values for probabilistic classification should be determined based upon the prevalence of the class in the target environment and the different misclassification costs. Practically, the uncertain cases with a posterior probability close to the cutoff values should be rejected by the classifier and referred to further examination.

To get an idea of the computational efficiency of our VS methods, we computed the average CPU time in a CV trial consumed by LinSBL+Bag on 30 bootstrap training samples. The simulations were conducted on cluster machines with Pentium processors (1GHz). Around 7 minutes and 3 minutes were needed for the leukemia data and the colon data, respectively. For prediction of brain tumors, in total 24 minutes were used for all 3 pairs of binary classification problems.

One limitation of the bagging strategy is that a single structure in the model is lost. To deal with this problem, one can adapt a similar approach as described in [93], in which the linear discriminant analysis (LDA) was bootstrapped, to generate an "average" classifier using the weighted average of the B sets of model parameters. Note that, the parameters of kernel models such as BayLSSVM and RVM are not directly ready for averaging. Whereas for linear kernel models, which is the case in this work, we could transform the model representation from kernel based to variable based, for which the parameters can be averaged and become easy to interpret. Also be aware that, unlike in [93] our bagging strategy uses different variables on different bootstrap samples, the weights for the less selected variables should be lower as well. However, the output class probability of the single "average" classifier will not be the same as the average class probability from the bagged models, which is usually more reliable and accurate than the outcome of a single model. Nevertheless, evaluation of such average models is still needed. One direction for future investigation could also be in finding a good way to transform the model aggregation into a single structure model, which is easy to explain clinically.

5.5 Conclusions

The largest potential problem for classification described here lies in the use of datasets with a small sample size and a huge dimensionality. The populations are usually underrepresented in this situation, which might result into a serious bias towards the training set, i.e. with a high training performance for a single model after variable selection and possibly a much lower generalization performance on the unseen data. This motivates the use of a bagging strategy in order to improve the reliability and lower the uncertainty in both variable selection and modelling. Experimental results confirm the advantages of the bagging strategy.

We have demonstrated that the use of the proposed variable selection can enhance the reliability of the models and thus increase the generalization performance of the models in these experiments. Furthermore, unlike the popular variable ranking methods such as RFE and the Fisher's criterion methods, the proposed method requires no additional step in order to decide on the number of variables to be used in the models. The variables are selected within a Bayesian framework, and the procedure is shown to be computationally efficient if the sample size is small. The number of occurrences of a variable being selected can serve as an importance measure for the variable. Our results imply that the linear sparse Bayesian learning plus bagging should play some role in variable selection for biomedical classification tasks.

Chapter 6

Conclusions and future research

In this thesis, we have given an overview of the classical and state-of-the-art techniques for linear and nonlinear classification, which have been applied to a several medical diagnostic problems. To conclude, we first give an outline of the key topics and major conclusions from model development and evaluations on our real-life medical classification problems. Interesting topics and directions for further research are pointed out in the second part of this chapter.

6.1 Summary

With the recent advancement of computational technologies along with the technologies in health care and biomedical field, the research community in medical intelligent systems and in machine learning is facing new challenges and opportunities to contribute substantially to clinical medicine and general health care practice.

This thesis examines the use of intelligent computational methods in the development of predictive models for medical classification based on patient data. Challenges of the problem not only lie in general issues in statistical learning and machine learning, but also in the rising difficulties due to the characteristics of data collected in medical environment. In addition to most importantly the accuracy in prediction, other desired properties of predictive models intended for clinical decision support include: (1) to take into account the uncertainty involved in the modelling and prediction, (2) interpretability.

For these reasons, our applied research was focused on the study of probabilistic machine learning approaches applied to classification, particularly the Bayesian kernel based algorithms including Bayesian least squares support vector machines and the sparse Bayesian learning models. We also proposed and analyzed feature selection schemes within the Bayesian framework. For high dimensional data sets, special attention has been given to take into account the uncertainty in model (feature) selection. The methodologies have been successfully applied to several real world medical diagnosis problems.

Some of the key issues addressed Chapter 3, 4 and 5 are highlighted as follows:

- **Bayesian methods** provide a unifying approach to conducting (hyper)parameter tuning, input variable selection, model comparison and outcome prediction for probabilistic models.
- **Kernel-based models** are flexible and powerful tools for supervised learning. Compared with mulitilayer perceptrons, kernel-based models have less nuisance (hyper)parameters to tune, e.g. no need to determine the structure (such as number of hidden layers and hidden neurons) of the model a priori. It is the kernels that determines the potentially higher dimensional feature space where the linear classification is to be applied. By integrating kernel-based modelling with the Bayesian framework, the tradeoff between the model fitness and model complexity can be controlled automatically.
- **Sparseness** is a good property for kernel-based models, which can not only save the time and memory for future model prediction but also lead to a better generalization performance as well. For relevance vector machines, sparseness is achieved by ARD within the Bayesian framework, i.e. to maximize the model evidence while choosing the appropriate priors for the model parameters. Support vector classifiers are in nature sparse as margin maximizers. Sparseness can also be imposed to least squares support vector classifiers by iterative pruning which mimic the SVM margin maximization procedure.
- **Regularization** is essential in modelling. Even for classical linear classifiers, such as logistic regression or linear discriminant analysis models, an appropriate control of the parameter complexity can usually lead to better generalization performance, as has been shown in the results of Chapter 4.
- Although the use of logistic likelihood might be conceptually more appropriate for classification problems, the use of **squared error based likelihood** (such as for LS-SVMs) could lead to more stable and much faster learning process. And these two different approaches to classification usually lead to comparable performance.

- **Variable selection** is important not only because it might improve the model performance, but also it can help in understanding the underlying mechanism of the disease and have economic impact on data acquisition. The level of the performance, for the linear or nonlinear classifiers we considered, is more related to the choice of the variable subset than to the choice of the model type. The forward selection using Bayesian LS-SVMs has been shown to be valuable in the ovarian tumor classification problem. Some domain knowledge is helpful for variable selection in order to find a more reliable model from a clinical point of view.
- **Sampling techniques and model ensemble** can help in stabilize and reduce the variance in variable selection, modelling and predictions, and thereby improve the performance over individual models. This is more prominent when the individual models disagree with each other. For data sets which have a high dimensionality and a small sample size, the bagged models using different variable subset and different bootstrap training samples appeared to be perform significantly better than the individual models.
- **Model interpretability** is also desirable for medical diagnostic models. Kernel-based models are usually considered as blackbox models which are difficult to understand. A compromise can be made by the model complexity and the interpretability by use of additive kernels. For LS-SVMs, we can also gain some insights to the decision of the classifiers by using kernel principal component regression and nonlinear biplots.
- **Linear** models performs sufficiently well for the ovarian cancer and for the other cancer diagnostic problems considered in this thesis. Nevertheless, the **nonlinear** modelling approaches provided by the kernel-based algorithms are still worth of trying for the other applications.

6.2 Future research

There are some critical open problems in applying probabilistic machine learning algorithms for medical classification problems. Further research is still needed in order to improve the models with respect to the following desired properties: 1) robustness of the models both in handling missing data and noisy/erroneous data; 2) cost sensitive modelling approach which can take into account the costs in a certain input measurements; 3) interpretability from the aspects of both understanding of the model behavior and abstracting the potential knowledge discovered by the model.

- The Bayesian framework we applied in this thesis (for both RVMs and Bayesian LS-SVMs) used type II maximum likelihood (i.e. evidence

framework), which is not a complete Bayesian approach. It exploits Gaussian (or Laplace) approximation for integration of the posterior probability for the parameters and the models. However, the approximated model evidence might be inaccurate if the underlying Gaussian assumptions are violated. Sampling method such Markov chain Monte Carlo (MCMC) might give more accurate results with the a much higher computational cost. Alternatively, variational methods can also be used for approximate Bayesian inference, where the lower bound of the marginal likelihood (evidence) is maximized [8].

- A robust modeling approach is desired for dealing with outliers in the data set. It might be interesting to apply similar techniques as in the robust LS-SVMs [95] and robust RVMs [29], which are both original for regression only, to our classification problems.

- We have utilized two Bayesian variable selection methods: the forward or stepwise selection of variables via Bayesian LS-SVMs, and the use of linear sparse Bayesian logit models. These methods gave quite satisfactory results in our applications. However, the computational cost become a burden when the number of variables or the number of training patterns increases. The question is: how can we jointly optimize variable selection and model fitting within the Bayesian framework for kernel-based models in an efficient way? Or we could consider a compromise by use of additive kernels with an backfitting like learning algorithm.

- We are also facing the problems dealing with uncertainty resulted from the input variables due to the effect of e.g. noise and subjective measurement. How can we quantify such uncertainty and employ them in the learning process and model prediction. The automatic variable selection should also be able to take into account such uncertainties in variables, so as to improve the reliability of the variable subset. This problem is maybe more application specific.

- As to model interpretation, besides visualizing the model output or decisions, can we extract some knowledge from the models by e.g. linguistic rule extraction? Many research has been already invested on this issue in order to open up the blackbox of neural networks [4][118]. The main concern here is a tradeoff between a faithful representation of the model and a good generalization performance using the inducted rules.

From the aspects of applications, in particular for predictive modelling for ovarian cancer diagnosis, there are also some topics that remain for further research:

- For the multi-center study, further investigation is still needed to analyze the difference among the centers, and to see whether different models

are needed for different centers. Model ensembles of individual models generated from different centers could be of interest.

- It could be useful to combine the kernel-based models with the existing Bayesian belief networks (either the expert Bayesian belief network or the Bayesian belief network learning from data). An ensemble of these two types of models might further improve the model performance, and moreover, the Bayesian belief networks can handle missing values in a natural way, which is not easy for kernel-based models.
- Further work is also needed to subclassify the benign and malignant tumors.
- High throughput technologies enable us to analyze the tumors using the genomic, protemic or metabolomic data. Recently, quite some projects have been undergone to characterize the gene expression patterns for ovarian tumors, especially for epithelial ovarian cancers [90][108][87]. This has posed many challenges on how to analyze these data and get deeper understanding of the diseases.

Appendix A

Medical background of the diseases

Ovarian cancer

Ovarian cancer is a disease produced by the rapid growth and division of cells within one or both ovaries-reproductive glands in which the ova, or eggs, and the female sex hormones are made [75]. When growth control is lost and cells divide too much and too fast, a cellular mass, or tumor, is formed. If the tumor is confined to a few cell layers, for example, surface cells, and it does not invade surrounding tissues or organs, it is considered benign. If the tumor spreads to surrounding tissues or organs, it is considered malignant, or cancerous. When cancerous cells break away from the original tumor, travel through the blood or lymphatic vessels, and grow within other parts of the body, the process is known as metastasis.

Brain tumors

A brain tumor is a mass of unnecessary and abnormal cells growing in the brain [75][43]. The cell of origin determines the type of the brain tumor. A tumor that starts in the brain is a primary brain tumor. Glioblastoma multiform, astrocytoma, medulloblastoma and ependymoma are examples of primary brain tumors. If cells have spread to the brain from a cancerous tumor in another part of the body, they are known as secondary brain tumors or metastatic brain tumors. Brain tumors are usually located in the posterior third of the brain in childhood and in the anterior two-thirds of the brain in adulthood.

Leukemia

Leukemia is a cancer of the blood in which blood cells mature abnormally or incompletely [75][27]. Leukemia occurs when there is an excess of abnormal white blood cells in the blood. These abnormal or immature cells accumulate in the body. They are not able to carry out the functions of normal blood cells. The most common types of leukemia include acute lymphocytic leukemia (ALL), acute myelogenous leukemia (AML), chronic lymphocytic leukemia (CLL), and chronic myelogenous leukemia (CML).

Colon cancer

Colon cancer is a disease in which malignant (cancer) cells form in the tissues of the colon [75][27]. The colon is part of the body's digestive system. The human colon is a muscular, tube-shaped organ measuring about 4 feet long. The colon has 3 main functions: to digest and absorb nutrients from food, to concentrate fecal material by absorbing fluid (and electrolytes) from it and to store and control evacuation of fecal material. Most colon cancers are adenocarcinomas, tumors that develop from the glands lining the colon's inner wall. These tumors are sometimes referred to as colorectal cancer, reflecting the fact that the rectum, the end portion of the colon, can also be affected.

Appendix B

Variable descriptions for the IOTA data set

Clinical information

Family history

`FAMHYSTOVCA` Number of first degree relatives with ovarian cancer $(0, 1, \cdots)$

`FAMHYSTBRCA` Number of first degree relatives with breast cancer $(0, 1, \cdots)$

`PERSHISTOVCA` Personal history of ovarian cancer

`PERSHISTBRCA` Personal history of breast cancer

Medical history

`AGE` Age of the patient (years).

`PARITY` Parity (number of deliveries).

`HYSTERECTOMY` Hysterectomy (1 - yes, 0- no)

`menopause` Menopausal status (0 - pre- or 1- postmenopausal). This is a transformed variable. In the transformed coding, the patients with hysterectomy or unknown status are assigned to the pre- and postmenopausal status as follows: if age < 50 then premenopausal, otherwise postmenopausal status.

`HORMTHERAPY` Hormonal therapy (1 - yes, 0 - no)

`PAIN` Pelvic pain during the scan. (1 - yes, 0 - no)

`PMB` Presence of postmenopausal bleeding within one month prior to the ultrasonography (1 - yes, 0 - no)

Serum tumor markers

`CA125` Measure of tumor marker in the blood (in international units/ml)

`l_ca125` The logarithm of CA125 level.

Color Doppler imaging and blood flow indices

`COLSCORE` A subjective semi-quantitative assessment of the amount of blood flow. Three dummy variable are created for this variable:

dummy variable	COLSCORE=1 No blood flow	COLSCORE=2 Weak blood flow	COLSCORE=3 Normal blood flow	COLSCORE=4 High blood flow
col2	1	1	0	0
col3	1	1	1	0
col4	1	1	1	1

`VENOUS` Presence of venous blood flow (0-No, 1-Yes).

`PI` Pulsatility index: PI=(S - D) / A. [1]

`RI` Resistence index: RI=(S - D) / S.

`PSV` The peak Doppler shifted frequency (cm/s).

`TAMX` The time-averaged maximum velocity (cm/s).

Ultrasonography

B-mode ultrasonography

`OvDmax` The size (measured in mm) of the ovary in two perpendicular planes. This is a transformed variable by taking the maximum value out of the three diameters of the ovary in two perpendicular planes.

[1] This is an index obtained with color Doppler imaging(`CDI`). S=the peak Doppler shifted frequency(PSV), D=the minimum Doppler shifted frequency (or end-diastolic velocity), A=the time-averaged maximum velocity (or TAMX).

`OvVol` The volume (ml) of the ovary is calculated from the three diameters in two perpendicular planes. This is a transformed variable from the three diameter measurements.

`LesDmax` [2] The size (measured in mm) of the lesion in two perpendicular planes. This is a transformed variable by taking the maximum value out of the three diameters of the lesion in two perpendicular planes.

`LesVol` The volume of the lesion is calculated from the three diameters in two perpendicular planes. This is a transformed variable from the three diameter measurements.

`ORIGIN` The origin of the tumor, it is coded as: 1 - ovarian, 2 - tubal, 3 - other, 4 - uncertain. This variable have been transformed to 3 dummy variables, they are coded as:

dummy variable	ORIGIN=1	ORIGIN=2	ORIGIN=3	ORIGIN=4
origin1	1	0	0	0
origin2	0	1	0	0
origin3	0	0	1	0

`ASCITES` The presence of ascites (i.e.fluid outside the pouch of Douglas) is noted (yes/no).

`FLUID` Fluid in the pouch of Douglas (mm) is measured in a sagittal plane (the largest anteroposterior diameter is given).

`uni` [3] Unilocular (a unilocular cyst without septa and without solid parts or papillary structures).

`unisol` Unilocular cyst with solid component (a unilocular cyst with a measurable solid component or at least one papillary structure).

`mul` Multilocular (a cyst with at least one septum but no measurable solid components or papillary projections).

`mulsol` Multilocular with solid component (a multilocular cyst with a measurable solid component or at least one papillary structure).

[2] The adnexal lesion is that part of an ovary or of an adnexal mass that is judged by ultrasonography to be not consistent with normal physiology. This can be a persistent unilocular cyst, surrounded by normal looking ovarian stroma with some follicles. In this case the whole ovary containing the cyst is the 'ovary', whereas the unilocular cyst is the 'lesion'. Both are measured and the cyst is described as being 'unilocular' and not 'unilocular-solid'. In other cases the lesion is separate from the ovary (e.g. hydrosalpinx). Again, both ovary and lesion are measured separately. In other cases no normal ovarian stroma is visualized. In these cases the lesion and the ovary are undistinguishable and the measurement of lesion and ovary will be the same.

[3] All lesions are qualitatively classified into one of 5 categories.

`sol` Solid (a tumor where the solid components comprise 80% or more of the tumor when assessed in a two-dimensional section).

`SolidDMax` [4] In cystic-solid tumors the largest solid component is measured separately (in three perpendicular planes). SolidDMax is the transformed variable by taking the maximum value of three measures of the solid components.

`SolidVol` Volume for the solid components is the transformed variable by calculating volume using three measures of the solid components.

`WALLREGULARITY` Whether the internal wall of the lesion is smooth (0) or irregular (1). If there is a solid papillary projection, then the wall is irregular by definition.

`NRLOCULES` The number of locules are counted (1, 2, 3, 4, 5-10, >10) (i.e. the real number of locules, as assessed in various planes).

Morphology

`PAPILLATION` Presence of solid papillary projections, which are defined as any solid projections into the cyst cavity from the cyst wall greater than or equal to 3 mm in height.

`PAPHEIGHT, PAPBASE1, PAPBASE2` The largest size (mm) of the solid papillary projections is measured in 3 planes: height, base, and base.

`PAPNR` The number of separate papillary projections (1/2/3/more).

`PAPFLOW` The presence of flow within some of these projections (yes/no).

`PAPSMOOTH` Whether the largest papillation is smooth (0) or irregular (1).

`BILATERAL` Masses involved in two ovaries (1 - yes, 0 - no).

`SHADOWS` The presence of acoustic shadows, defined as loss of acoustic echo behind a sound-absorbing structure (1 - yes, 0 - no).

`SEPTUM` Measure of the thickness (in mm) of the thickest septum where it appears to be at its widest. A septum is defined as a thin strand of tissue running across the cyst cavity from one internal surface to the contralateral side.

`septumg3` A dichotomized version of SEPTUM: if SEPTUM>3mm, septumg3=1; otherwise septumg3=0.

[4] Solid papillary projections are defined as any solid projections into the cyst cavity from the cyst wall greater than or equal to 3 mm in height. The largest is measured in 3 planes: height, base, and base.

`INCOMPLSEPTUM` An incomplete septum (as seen in hydrosalpinges) is defined as a thin strand of tissue running across the cyst cavity from one internal surface to the contralateral side, but which is not complete in some scanning planes.

Echogenicity

The dominant feature of the cystic contents is described as anechoic (black), low-level echogenic (homogeneous low level echogenic as seen in mucinous tumors).

`lucent` Anechoic.

`low_level` Low-level echogenic (homogeneous low level echogenic as seen in mucinous tumors)

`g_glass` Ground glass appearance (homogeneously dispersed echogenic cystic contents

`haem` Hemorrhagic (with internal thread-like structures, representing fibrin strands

`mixed` It is possible to describe the echogenicity as star-shaped, cobweb-like or jelly-like) or mixed echogenic (as often seen in teratomas).

Bibliography

[1] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

[2] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *Proc. Natl. Acad. Sci.*, volume 96, pages 6745–6750, USA, 1999. National Academy of Science.

[3] D.G. Altman and P. Royston. What do we mean by validating a prognostic model? *Statistics in Medicine*, 19:453–473, 2000.

[4] R. Andrews, J. Diederich, and A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389, 1995.

[5] P. Antal, G. Fannes, D. Timmerman, B. De Moor, and Y. Moreau. Bayesian applications of belief networks and multilayer perceptrons for ovarian tumor classification with rejection. *Artificial Intelligence in Medicine*, 29:39–60, 2003.

[6] P. Antal, H. Verrelst, D. Timmerman, Y. Moreau, S. Van Huffel, B. De Moor, and I. Vergote. Bayesian networks in ovarian cancer diagnosis: potentials and limitations. In *Workshop on Fusion of Domain Knowledge with Data for Decision Support, 16th Uncertainty in Artificial Intelligence Conference*, pages 103–109, Standford, CA, 2000.

[7] J.H. Van Bemmel and M.A. Musen, editors. *Handbook of Medical Informatics*. Springer, 1998.

[8] C.M. Bishop and M.E. Tipping. Variational relevance vector machines. In C. Boutilier and M. Goldszmidt, editors, *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 46–53. Morgan Kaufmann, 2000.

[9] C.M. Bishop and M.E. Tipping. Bayesian regression and classification. In J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Models and Applications*, volume 190 of *NATO Science Series III: Computer and Systems Sciences*, pages 267–288. IOS Press, 2003.

[10] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[11] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proc. of 5th Annual ACM Workshop Computational Learning Theory*, pages 144–152, 1992.

[12] R. A. Bradley and M. E. Terry. The rank analysis of incomplete block design I: the method of paired comparisons. *Biometrika*, 39:324–345, 1952.

[13] I. Bratko, I. Mozetic, and N.Lavrac. *KARDIO: A study in Deep and Qualitative Knowledge for Expert Systems*. MIT Press, Cambridge MA., 1989.

[14] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[15] A.P. Candiota, C. Majós, A. bassols, M.E. Cabañas, J.J. Acebes, M.R. Quintero, and C. Arús. Assignment of the 2.03 ppm resonance in in vivo ^{1}H MRS of human brain tumour cystic fluid: contribution of macromolecules. *MAGMA 2004*, 2004. in press.

[16] E. Coiera. Artificial intelligence systems in routine clinical use. *Online*, 1999.

[17] E. Coiera. *A Guide to Health Informatics*. Hodder & Stoughton Educational, UK, 2nd edition, 2003.

[18] G. Condous, A. Khalid E. Okaro, D. Timmerman, C. Lu, Y. Zhou, S. Van Huffel, W. Collins, and T Bourne. The use of a new logistic regression model for predicting the outcome of pregnancies of unknown location. *Human Reproduction*, 19(8):1900–1910, 2004.

[19] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2002.

[20] E.R. DeLong, D.M. DeLong, and D.L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44:837–845, 1988.

[21] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38, November 1977.

[22] A. Devos, L. Lukas, J.A.K. Suykens, L. Vanhamme, A.R. Tate, F.A. Howe, C. Majós, A. Moreno-Torres, M. Van der Graaf, C. Arús, and S. Van Huffel. Classification of brain tumours using short echo time ^{1}H MRS spectra. *Journal of Magnetic Resonance*, 170:164–175, 2004.

[23] T.G. Dietterich. Machine-learning research - four current directions. *AI Magazine*, 18:97–136, 1997.

[24] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* A Wiley-Interscience Publication, John Wiley & Sons, Inc, New York, second edition, 2001.

[25] S.K. Mukherji (eds.). *Clinical Applications of Magnetic Resonance Spectroscopy.* Wiley-Liss, 1998.

[26] B. Efron and R.J. Tibshirami. *An introduction to the bootstrap.* Chapman & Hall, New York, 1993.

[27] eMedicineHealth. http://www.emedicinehealth.com/.

[28] G. Fannes. *Bayesian learning with expert knowledge: Transforming informative priors between Bayesian networks and multilayer perceptrons.* PhD thesis, Faculty of Engineering, K.U.Levuen, Leuven, Belgium, 2004.

[29] A. Faul and M.E. Tipping. A variational approach to robust regression. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proc. of the International Conference on Artificial Neural Networks (ICANN'01)*, pages 95–102. Springer, 2001.

[30] J. Friedman. Another approach to polychotomous classification. *Technical report, Standford University*, 1996.

[31] EU funded INTERPRET project. IST-1999-10310, http://carbon.uab.es /INTERPRET.

[32] L.M. Fu and E.S. Youn. Improving reliability of gene selection from microarray functional genomics data. *IEEE Transations on Information Technology in Biomedicine*, 7(3):191–196, 2003.

[33] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M. A. Caliguiri, C. D. Bloomeld, and E.S. Lander. olecular classication of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[34] V. Govindaraju, K. Young, and A. A. Maudsley. Proton nmr chemical shifts and coupling constants for brain metabolites. *NMR in Biomedicine*, 13:129–153, 2003.

[35] J.C. Gower and D.J. Hand. *Biplots.* Chapman & Hall, London, 1996.

[36] Y. Grandvalet and S. Canu. Outcome of the equivalence of adaptive ridge with least absolute shrinkage. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 11, 1998.

[37] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:382–422, 2002.

[38] J.A. Hanley and B. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.

[39] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*. Mongraphs on statistics and applied probability. Chapman & Hall/CRC, Boca Raton, Florida, 1999.

[40] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning - data mining, inference, and prediction*. Springer, New York, 2001.

[41] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998.

[42] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 1999.

[43] National Cancer Institute. http://www.nci.nih.gov/cancertopics.

[44] I. Jacobs, D. Oram, J. Fairbanks, J. Turner, C. Frost, and J.G. Grudzinskas. A risk of malignancy index incorporating CA 125, ultrasound and menopausal status for the accurate preoperative diagnosis of ovarian cancer. *Br J Obstet Gynaecol*, 97:9220–929, 1990.

[45] A. Jain, A. Jain, S. Jain, and L. Jain. *Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis*, volume 39 of *Series in Machine Perception & Artificial Intelligence*. World Scientific, 2000.

[46] H. Jeffreys. *Theory of Probability*. Oxford University Press, New York, 1961.

[47] G. Karakiulakis, C. Papanikolaou, S.M. Jankovic, A. Aletras, E. Papakonstantinou, E. Vretou, and V. Mirtsou-Fidani. Increased type iv collagen-degrading activity in metastases originaing from priorary tumors of human colon. *Invasion and metastasis*, 17(3):158–168, 1997.

[48] R. Kohavi and G. John. Wrapper for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[49] P.J.G. Lisboa, S.P.J. Kirby, A. Vellido, Y.Y.B. Lee, and W. El-Deredy. Assessment of statistical and neural networks methods in NMR spectral classification and metabolite selection. *NMR in Biomedicine*, 11:225–234, 1998.

[50] Y. Li, C. Campbell, and M.E. Tipping. Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics*, 18:1332–1339, 2002.

[51] W.J. Long. Medical informatics: reasoning methods. *Artificial Intelligence in Medicine*, 23:71–87, 2001.

[52] L. Lukas, A. Devos, J.A.K. Suykens, L. Vanhamme, F.A. Howe, C. Majós, A. Moreno-Torres, M. Van der Graaf, A.R. Tate, C. Arús, and S. Van Huffel. Brain tumour classification based on long echo proton MRS signals. *Artificial Intelligence in Medicine*, 31:73–89, 2004.

[53] C. Lu, J. De Brabanter, S. Van Huffel, I. Vergote, and D. Timmerman. Logistic regression and artificial neural networks for classification of ovarian tumors. Internal Report 01-137, ESAT-SISTA, K.U.Leuven, Leuven, Belgium, 2004.

[54] C. Lu, A. Devos, J.A.K. Suykens, C. Arús, and S. Van Huffel. Bagging linear sparse bayesian learning models for variable selection in cancer diagnosis. Internal Report 04-176, ESAT-SISTA, K.U.Leuven, Leuven, Belgium, July 2004.

[55] C. Lu, J. De Brabanter, S. Van Huffel, I. Vergote, and D. Timmerman D. Using artificial neural networks to predict malignancy of ovarian tumors. In *Proceeding of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2001)*, pages CD–ROM, 4.2.2–6, 2001. Istanbul, Istanbul.

[56] C. Lu, T. Van Gestel, J.A.K. Suykens, S. Van Huffel, I. Vergote, and D. Timmerman. Bayesian least squares support vector machines for classification of ovarian tumors. In *Proc. of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTN2004)*, Leuven, Belgium, July 2004.

[57] C. Lu, J. Suykens, S. Van Huffel, I. Vergote, and D. Timmerman. Chapter 11: Linear and nonlinear preoperative classification of ovarian tumors. In T. Ichimura and K. Yoshida, editors, *Knowledge Based Intelligent System for Health Care*, volume 7 of *International Series on Advavnced Intelligence*, pages 343–382, Australia, 2004. Advanced Knowledge International.

[58] C. Lu, T. Van Gestel, J. Suykens, S. Van Huffel, I. Vergote, and D. Timmerman. Classification of ovarian tumor using bayesian least squares

support vector machines. In M. Dojat, E. Keravnou, and P. Barahona, editors, *Proceedings of the 9th Conference on Artificial Intelligence in Medicine Europe (AIME 03)*, volume 2780, pages 219–228. Springer-Verlag, October 2003. Protaras, Cyprus.

[59] C. Lu, T. Van Gestel, J. Suykens, S. Van Huffel, I. Vergote, and D. Timmerman. Preoperative prediction of malignancy of ovariun tumor using least squares support vector machines. *Artificial Intelligence in Medicine*, 28(3):281–306, July 2003.

[60] D.J.C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):698–741, 1992.

[61] D.J.C. MacKay. Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. *Computation in Neural Systems*, 6:469–505, 1995.

[62] N. Mani and B. Lithgow. Medical and healthcare applications of intelligent information systems: An overview. In Brian Lithgow and Irena COsic, editors, *Proceedings of the Inaugural Conference of the Victorian Chapter of the IEEE Engineering in Medicine and Biology Society*, Australia, January 1999.

[63] G. L. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley, New York, 1992.

[64] J. Mercer. Function of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society London*, A 209:415–446, 1909.

[65] T.M. Michell. *Machine Learning*. McGraw-Hill, 1997.

[66] R.A. Miller. Medical diagnostic decision support systems–past, present, and future: a threaded bibliography and brief commentary. *J Am Med Inform Assoc*, 1(1):8–27, January 1994.

[67] N.M. Morel, J. M. Holland, J. Van Der Greef, E.W. Marple, C. Clish, J. Loscalzo, and S. Naylor. Primer on medical genomics part xiv: introduction to systems biology - a new approach to understanding disease and treatment. In *Mayo Clin Proc.*, volume 79, pages 651–658. Mayo Foundation for Medical Education and Research, May 2004.

[68] I. Nabney and C. Bishop. Netlab neural network software (release 3). http://www.ncrg.aston.ac.uk/netlab/.

[69] R.M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York, 1996.

[70] S.J. Nelson. Multivoxel magnetic resonance spectroscopy of brain tumors. *Molecular Cancer Therapeutics*, 2:497–507, 2003.

[71] J. Neter, M.H. Kutner, C.J. Nachtsheim, and W. Wasserman. *Applied Linear Statistical Models*. McGraw-Hill/Irwin, Chicago, Ill, fourth edition, 1996.

[72] J. Neves, L. Nelas V. Alves, A. Romeu, and S. Basto. An information system that supports knowledge discovery and data mining in medical imaging. In *ECCAI Advanced course in Artificial Intelligence (ACAI)*, Chania, Greece., 1999. Workshop(W13) on Machine Learning in Medical Applications.

[73] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naïve bayes. In S. Becker T. Dietterich and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 14, 2002.

[74] A.E. Nikulin, B. Dolenko, T. Bezabeh, and R.L. Somorjai. Near-optimal region selection for feature space reduction: novel preprocessing methods for classifying mr spectra. *NMR Biomed*, 11:209–216, 1998.

[75] Oncologychannel. http://www.oncologychannel.com/.

[76] R.F. Ozols, S.C. Rubin, G.M. Thomas, and S.J. Robboy. Epithelial ovarian cancer. In W.J. Hoskins, C.A. Perez, and R.C. Young, editors, *Principles and practice of gynecologic oncology*, pages 981–1058, Philadelphia, 2000. Lippincott Williams and Wilkins.

[77] K. Pelckmans, I. Goethals, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Componentwise least squares support vector machines. In L. Wang, editor, *Support Vector Machines: Theory and Applications*. Springer, 2005.

[78] K. Pelckmans, J.A.K. Suykens, T. Van Gestel, J. De Brabanter, L. Lukas, B. Hamers, B. De Moor, and J. Vandewalle. LSSVMlab: a matlab/c toolbox for least squares support vector machines. http://www.esat.kuleuven.ac.be/sista/lssvmlab/.

[79] J.C. Platt, N. Christianini, and J. Shawe-Taylor. Large margin dags for multiclass classification. In *Advances in Neural Information Processing System*, volume 12, pages 547–553. MIT Press, 2000.

[80] F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In J. Shavlik, editor, *Proceedings of the 15th International Conference on Machine Learning (IMLC-98)*, pages 445–453, San Francisco, CA, 1998. Morgan Kaufmann.

[81] Y. Reisman. clinical decision aids. a review of methods and assessment of systems. *Med. Inform.*, 21(3):179–197, 1996.

[82] B.D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, New York, 1996.

[83] Y.D. Rubinstein and T. Hastie. Discriminative vs informative learning. In *Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 49–53. AAAI Press, 1997.

[84] D.B. Rubin. *Multiple Imputation for Nonresponse in Surveys.* John Wiley & Sons, Inc, NewYork, 1987.

[85] B. Schölkopf, A. J. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10:1299–1319, 1998.

[86] B. Schölkopf and A. Smola. *Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond.* MIT Press, Cambridge, MA, 2002.

[87] D.R. Schwartz, S.L.R. Kardia, K.A. Shedden, R. Kuick, G. Michailidis, J.M.G. Taylor, D.E. Misek, R. Wu, Y. Zhai, D.M. Darrah, H. Reed, L.H. Ellenson, T.J. Giordano, E.R. Fearon, S.M. Hanash, and K.R. Cho. Gene expression in ovarian cancer reflects both morphology and biological behavior, distinguishing clear cell from other poor-prognosis ovarian carcinomas. *Cancer Research*, 62:4722–4729, August 2002.

[88] K. Shailubhai, H.H. Yu, K. Karunanandaa, J.Y. Wang, S.L. Eber, Y. Wang, N.S. Joo, H.D. Kim, B.W. Miedema, and S.Z. Abbas *et al.* Uroguanylin treatment suppress polyp formation in the apc$^{Min/+}$ mouse and induces apoptosis in human colon adenocarcinoma cells via cyclic gmp. *Cancer Res.*, 60:5151–5163, 2000.

[89] R. Simon, M. Radmacher, K. Dobbin, and L. McShane. Pitfalls in the use of dna microarray data for diagnostic and prognostic classification. *J Natl Cancer Inst*, 95:14–18, 2003.

[90] F. De Smet. *Microarrays : algorithms for knowledge discovery in oncology and molecular biology.* PhD thesis, Faculty of Engineering, K.U.Leuven, Leuven, Belgium, May 2004.

[91] A.E. Smith, C.D. Nugent, and S.I. McClean. Evaluation of inherent performance of intelligent medical decision support systems: utilising neural networks as an example. *Artificial Intelligence in Medicine*, 27:1–27, January 2003.

[92] R.L. Somorjai, B. Dolenko, and R. Baumgartner. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics*, 19:1484–1491, 2003.

[93] R.L. Somorjai, B. Dolenko, A. Nikulina, P. Nickersonb, D. Rushb, A. Shawa, M. Glogowskia, J. Rendella, and R. Deslauriers. Distinguishing normal from rejecting renal allografts: application of a three-stage classification strategy. *Vibr. Spectrosc*, 28:97–102, 2002.

[94] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[95] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines.* World Scientific, Singapore, 2002.

[96] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. Support vector machines: least squares approaches and extensions. In J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Models and Applications*, volume 190 of *NATO Science Series III: Computer and Systems Sciences*, pages 155–177. IOS Press, 2003.

[97] J.A.K. Suykens, T. Van Gestel, J. Vandewalle, and B. De Moor. A support vector machine formulation to PCA analysis and its kernel version. *IEEE Transactions on Neural Networks*, 14(2):447–450, 2003.

[98] J.A. Swets and R.M. Pickett. *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory.* Acdemic Press, New York, 1982.

[99] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58(1):267–288, 1996.

[100] D. Timmerman, T.H. Bourne, A. Tailor, W.P. Collins, H. Verrelst, K. Vandenberghe, and I. Vergote. A comparison of methods for preoperative discrimination between malignant and benign adnexal masses: The development of a new logistic regression model. *Am J Obstet Gynecol*, 181:57–65, 1999.

[101] D. Timmerman, P. Schwärzler, W.P. Collins, F. Claerhout, M. Coenen, F. Amant, I. Vergote, and T.H. Bourne. Subjective assesment of adnexal masses with the use of ultrasonography: an analysis of interobserver variability and experience. *Ultrasound Obstet Gynecol*, 13:11–16, 1999.

[102] D. Timmerman, A.C. Testa, T. Bourne, E. Ferrazzi, L. Ameye, M.L. Konstantinovic, W.P. Collins, I. Vergote, S. Van Huffel, and L. Valentin. A new logistic regression model to distinguish between the benign and malignant adnexal mass before surgery: a multicentre study by the international ovarian tumour analysis (iota) group. Technical report, 2004. submitted.

[103] D. Timmerman, L. Valentin, T.H. Bourne, W.P Collins, H. Verrelst, and I. Vergote. Terms, definitions and measurements to describe the ultrasonographic features of adnexal tumors: a consensus opinion from the international ovarian tumor analysis (iota) group. *Ultrasound Obstet Gynecol*, 16:500–505, 2000.

[104] D. Timmerman, H. Verrelst, T.H. Bourne, B. De Moor, W.P. Collins, I. Vergote, and J. Vandewalle. Artificial neural network models for the preoperative discrimination between malignant and benign adnexal masses. *Ultrasound Obstet Gynecol*, 13:17–25, 1999.

[105] M.E. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In *Proceedings of Artificial Intelligence and Statistics '03*, 2003.

[106] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[107] M.E. Tipping. Sparsebayes v1.0: a matlab implementation. http://research.microsoft.com/mlp/RVM/SparseBayesV1.00.tar.gz.

[108] P. Tonin, T.J. Hudson, F. Rodier, M. Bossolasco, P. Lee, J. Novak, E. Manderson, D. Provencher, and A.M. Mes-Masson. Microarray analysis of gene expression mirrors the biology of an ovarian cancer model. *Oncogene*, 20(45):6617–26, 2001.

[109] Lil Valentin. Gray scale sonography, subjective evaluation of the color doppler image and measurement of blood flow velocity for distinguishing benign and malignant tumors of suspected adnexal origin. *European Journal of Obstetrics & Gynecolog) and Reproductive Biology*, 72:63–72, 1997.

[110] T. Van Gestel, J.A.K. Suykens, D.E. Baestaens, A. Lamrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle. Financial time series prediction using least squares support vector machines within the evidence framework. *IEEE Transactions on Neural Networks (Special Issue on Financial Engineering)*, 12(4):809–821, 2001.

[111] T. Van Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vandewalle. A bayesian framework for least squares support vector machine classifiers. *Neural Computation*, 15(5):1115–1148, 2002.

[112] T. Van Gestel. *From linear to kernel based methods in classification, modelling and prediction*. PhD thesis, ESAT-SISTA, Katholike Unieversity Leuven, Leuven, Belgium, 2002.

[113] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[114] I. Vergote, J. De Brabanter, A. Fyles, K. Bertelsen, N. Einhorn, P. Sevelda, M.E. Gore, J. Kaern, H. Verrelst, K. Sjovall, D. Timmerman, J. Vandewalle, M. Van Gramberen, and C.G. Trope. Prognostic importance of degree of differentiation and cyst rupture in stage I invasive epithelial ovarian carcinoma. *Lancet*, 357(9251):176–182, 2001.

[115] J. Weston, A. Elisseeff, G. BakIr, and F. Sinz. Spider: a library of objects in matlab (version 1.4). http://www.kyb.tuebingen.mpg.de/bs/people/spider/index.html.

[116] J. Weston, A. Elisseeff, M.E. Tipping, and B. Schölkopf. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2002.

[117] M. Xiong, X. Fang, and J. Zhao. Biomarker identification by feature wrappers. *Genome Res.*, 11:1878–1887, 2001.

[118] Z.H. Zhou and Y. Jiang. Medical diagnosis with c4.5 rule preceded by artificial neural network ensemble. *IEEE Transactions on Information Technology in Biomedicine*, 7(1):37–42, 2003.

Publication list

Articles in international journals

- T. Van Den Bosch, D. Timmerman, D. Van Schoubroeck, C. Lu, J. De Brabanter and S. Van Huffel, Color doppler and gray-scale ultrasound evaluation of the postpartum uterus. *Ultrasound Obstet. Gynecol.*, vol. 20, no. 6, Dec 2002, pp. 586-591.
- C. Lu, T. Van Gestel, J.A.K. Suykens, S. Van Huffel, I. Vergote and D. Timmerman, Preoperative prediction of malignancy of ovarian tumors using least squares support vector machines. *Artificial Intelligence in Medicine*, vol 28, no. 3, Jul 2003, pp. 281-306.
- D. Van Schoubroeck, T. Van Den Bosch, K. Scharpe, C. Lu, S. Van Huffel, D. Timmerman, Prospective evaluation of blood flow in the myometrium and in the uterine arteries in the puerperium, *Ultrasound Obstet. Gynecol.*, vol. 23, no. 4, Apr 2004, pp. 378-381.
- G. Condous, E. Okaro, A. Khalid, D. Timmerman, C. Lu, Y. Zhou, S. Van Huffel, W. Collins and T Bourne, The use of a new logistic regression model for predicting the outcome of pregnancies of unknown location. *Human Reproduction*, vol. 19, no. 8, Aug 2004, pp. 1900-1910.
- G. Condous, C. Lu, S. Van Huffel, D.Timmerman, T. Bourne, Human chorionic gonadotrophin and progesterone levels in pregnancies of unknown location *International Journal of Gynecology and Obstetrics*, vol. 86, 2004, pp. 351-357.
- G. Condous, E. Okaro, A. Khalid, C. Lu, S. Van Huffel, T. Bourne, Do we need to follow up complete miscarriages with serum human chorionic gonadotrophin levels? Accepted for publication in *British Journal of Obstetrics and Gynaecology*.
- G. Condous, E. Okaro, A. Khalid, C. Lu, S. Van Huffel, D. Timmerman and T. Bourne, A prospective evaluation of a single visit strategy to manage pregnancies of unknown location (PUL). Accepted for publication in *Human Reproduction*, 2004.

Chapters in books

- C. Lu, J.A.K. Suykens, D. Timmerman, I. Vergote, S. Van Huffel, Linear and nonlinear preoperative classification of ovarian tumors, in T. Ichimura and K. Yoshida (eds.), *Knowledge Based Intelligent System for Health Care*, International Series on Advavnced Intelligence, vol. 7, Chapter 11, pp. 343-382, Advanced Knowledge International, Australia.

Articles in international conference proceedings

- C. Lu, J. De Brabanter, S. Van Huffel, I. Vergote, D. Timmerman, Using artificial neural networks to predict malignancy of ovarian tumors, in *Proc. of 23rd Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBS'01)*, Istanbul, Turkey, October 25-28, 2001, paper 4.2.2-6, 4 pp.

- L. Ameye, C. Lu, L. Lukas, J. De Brabanter, J.A.K. Suykens, S. Van Huffel, H. Daniels, G. Naulaers and H. Devlieger, Prediction of mental development of preterm newborns at birth time using LS-SVM, in *Proc. of the European Symposium for Artifical Neural Networks (ESANN)*, Bruges, Belgium, April 24-26 2002, pp. 167-172.

- G. Cirrincione, M. Cirrincione, C. Lu, S. Van Huffel, A novel neural approach to inverse problems with discontinuities (the GMR neural network), in *Proc. of 2003 International Joint Conference on Neural Networks (IJCNN'03)*, Portland, Oregon, Jul. 2003, pp. 3106-3111.

- G. Cirrincione, M. Cirrincione, C. Lu, M. Pucci, Direct Torque control of induction motors by use of the GMR Neural network, in *Proc. of 2003 International Joint Conference on Neural Networks (IJCNN'03)*, Portland, Oregon, July 20-24, 2003, pp. 2106-2111.

- C. Lu, T. Van Gestel, J.A.K. Suykens, S. Van Huffel, I. Vergote and D. Timmerman, Classification of ovarian tumor using Bayesian least squares support vector machines, in *Proc. of 9th conference on Artificial Intelligence in Medicine Europe (AIME 2003)*, Protaras, Cyprus, October 18-22, 2003, M. Dojat, E. Keravnou and P. Barahona (Eds.), published as: Lecture notes in Artificial Intelligence, Vol. 2780, pp. 219-228, Springer-Verlag, Berlin Heidelberg, 2003.

- C. Lu, T. Van Gestel, J.A.K. Suykens, S. Van Huffel, I. Vergote and D. Timmerman, Bayesian Least Squares Support Vector Machines for Classification of Ovarian Tumors, in *Proc. of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTN2004)*, Leuven, Belgium, July 5-9, 2004.

Abstracts in conference proceedings

- C. Lu, T. Van Gestel, J. Suykens, S. Van Huffel, I. Vergote and D. Timmerman, Blackbox classifiers for preoperative discrimination between malignant and benign ovarian tumors. Abstract at the Advanced NATO Study Institute on Learning Theory and Practice (NATO-ASI LTP 2002), Arenberg castle, Leuven, Belgium, July 8-19, 2002.

- C. Lu, G. Cirrincione and S. Van Huffel, The Generalised Mapping Regressor (GMR) neural network for inverse discontinuous problems. Poster presented at the PAI-IUAP 4/2 Study day, Katholieke Universiteit Leuven, Arenberg Castle, Leuven, Belgium, December 5, 2000.

- C. Lu, J. De Brabanter, S. Van Huffel, I. Vergote and D. Timmerman, Prediction of malignancy of ovarian tumors with logistic regression and artificial neural network models. Abstract at 9th annual meeting of the Belgian statistical society (BSS 2001), Oostende, Belgium, October 12-13, 2001.

- C. Lu, T. Van Gestel, J.A.K. Suykens and S. Van Huffel, Comparative study on variable selection for nonlinear classifiers. Abstract at 10th annual meeting of the Belgian statistical society (BSS 2002), Congress Center Rolduc, Kerkrade, The Netherlands, 2002.

- C. Lu, J.A.K. Suykens and S. Van Huffel, Variable selection using linear sparse Bayesian models for medical classification problems, Abstract at Belgian Day of Biomedical Engineering, Oct 17, 2003, Brussels.

- C. Lu, S. Van Huffel and J.A.K. Suykens, Intelligent medical diagnosis via machine learning, Abstract at First Flanders Engineering Ph.D. Symposium, Brussels, Belgium, 2003.

Internal reports (including submitted papers)

- C. Lu, J. De Brabanter, S. Van Huffel, I. Vergote and D. Timmerman, Logistic Regression and Artificial Neural Networks for Classification of Ovarian Tumors, Internal Report 01-137, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2001.

- C. Lu, A. Devos, J.A.K. Suykens, C. Arús, S. Van Huffel, Bagging linear sparse Bayesian learning models for variable selection in cancer diagnosis, IEEE Transactions on Information technology in BioMedicine, 2004, submitted.

- G. Condous, E. Okaro, A. Khalid, D. Timmerman, C. Lu, S. Van Huffel, W. Collins and T. Bourne, Can ectopic pregnancies be predicted using serum hormone levels and the human chorionic gonadotrophin ratio? British Journal of Obstetrics and Gynaecology, 2004, in revision.

- G. Condous, E. Okaro, A. Khalid, C. Lu, S. Van Huffel, T. Bourne, The accuracy of transvaginal ultrasonography for the diagnosis of ectopic pregnancy prior to surgery, revision submitted to Human Reproduction, 2004.
- G. Condous, E. Okaro, A. Khalid, C. Lu, S. Van Huffel, T. Bourne, The limitations of the use of discriminatory zones of serum human chorionic gonadotrophin for the diagnosis of ectopic pregnancy. In revision, 2004.

Chuan LU was born in GuangXi, People's Republic of China in December 1973. She received her bachelor's degree in Engineering (major in Management Information System) from Tianjin University, China in July 1995. She worked as a programmer/analyst at Samsung Electro-Mechanics Tianjin between May 1995 and May 1998 and at Motorola Tianjin from June till December in 1998. She came to Leuven afterwards to continue her study at the Katholieke Universiteit Leuven, Belgium. She obtained her master's degree in Artificial Intelligence (with option in Engineering and Computer Science) from the K.U. Leuven in July 2000. Since August 2000, she has been pursuing her doctorate at the SCD-SISTA laboratories, the Department of Electrical Engineering (ESAT) of the K.U. Leuven, under the supervision of Prof. Sabine Van Huffel and Prof. Johan Suykens.

Her research focuses on predictive modelling for medical diagnosis, and was funded by the projects of Belgium Federal Government, of the K.U. Leuven, and of the Fund for Scientific Research Flanders (FWO). Chuan LU was also supported by the scholarship for advanced foreign doctoral students of the Research Council of the K.U. Leuven from January 2003 to July 2004.