STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

Citation/Reference	Billiet L., Van Huffel S., Van Belle V. (2016), **Interval Coded Scoring System with Interaction Effects: a Sensitivity Study** 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM2016)
Archived version	Author manuscript/pdf
Published version	NA
Journal homepage	http://www.icpram.org/?y=2016
Author contact	lieven.billiet@esat.kuleuven.be your phone number *+ 32 (0)16 32 76 85*
IR	NA

(article begins on next page)

Interval Coded Scoring Index with Interaction Effects
A Sensitivity Study

Lieven Billiet[1,2], Sabine Van Huffel[1,2] and Vanya Van Belle[1,2]
[1]*STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Department of Electrical Engineering, KU Leuven, Leuven, Belgium*
[2]*iMinds Medical Information Technologies, Leuven, Belgium*
lieven.billiet@esat.kuleuven.be

Keywords: Sparse Optimization, Interpretability, Scoring Systems.

Abstract: Scoring systems have been used since long in medical practice, but often they are based on experience rather than a structural approach. In literature, the interval coded scoring index (ICS) has been introduced as an alternative. It derives a scoring system from data using optimization techniques. This work discusses an extension, ICS*, that takes variable interactions into account. Furthermore, a study is performed to give insight into the new model's sensitivity to noise, the size of the data set and the number of non-informative variables. The study shows interactions can mostly be discovered robustly, even in the presence of noise and spurious variables. A final validation on two UCI data sets further indicates the quality of the approach.

1 INTRODUCTION

When working in the medical field, one notices that applying standard Machine Learning approaches faces difficult challenges. Generic techniques such as Support Vector Machines (SVM) and Bayesian classifiers have been used (Chowriappa et al., 2014), but they most often offer a black-box solution of a problem. In order to accept the support of a system, a medical expert should understand and trust its recommendations. Therefore, interpretability is important. Looking back at medical practice since the early days, one can see that one kind of interpretable models has frequently been used in the medical world itself: scoring systems. Examples include APACHE-II, SIRS, Glasgow (pancreatitis) (Mounzer et al., 2012), PSI and CURB-65 (pneumonia) (Jeong et al., 2013). They are powerful methods, often based on clinical experience or mathematical models, but their discriminative power is limited due to their simplicity. Furthermore, most systems developed so far are not the result of a standardized or well-founded learning approach. Yet, studies to validate or compare commonly used scores have been conducted (Mounzer et al., 2012; Jeong et al., 2013). There have also been attempts to construct scoring systems with statistical methods (Yang et al., 2011) or directly from data (Van Belle et al., 2012), but the proposed models are restricted to the main effects or involve tuning ad hoc parameters.

Generating a scoring system from data involves finding a sparse model. This approach is well-known in fields such as compressed sensing, where ℓ_0 or ℓ_1 minimization is used to induce this property. Some example methods include the LASSO or basic pursuit (Davenport et al., 2012). Similar approaches have been used to generate scoring systems (Ustun et al., 2013) recently, but focus on giving integer coefficients to previously defined features in general. Yet, other approaches, such as the Interval Coded Scoring Index (ICS) (Van Belle et al., 2012), focus rather on intervals, but are limited to main effects.

The remainder of this paper is structured as follows: the next section introduces the extension of ICS that allows for interaction effects: ICS*. Section 3 discusses the sensitivity study carried out on synthetic data. Finally, the framework is applied on two UCI data sets after which we conclude with a discussion and a preview to future work.

2 THE ICS* ALGORITHM

The model can be best explained by expressing it in an SVM framework as a binary classification problem. The primal formulation of a (non-linear) SVM is given by (Vapnik, 1995):

$$\min_{w,b,\varepsilon} \quad \frac{1}{2}w^T w + \gamma \sum_{i=1}^{N} \varepsilon_i \tag{1}$$

$$\text{subject to: } \begin{cases} y_i(w^T \varphi(x_i) + b) \geq 1 - \varepsilon_i, & \forall i = 1..N \\ \varepsilon_i \geq 0, & \forall i = 1..N \end{cases}$$

In this formulation, (x_i, y_i) are pairs of observations and labels, w and b the coefficients and bias of the

model, respectively, and ε_i slack variables used for the regularization controlled by γ.

The original ICS approach restricts the feature map $\varphi(x)$ (Van Belle et al., 2012). Instead of the original input vectors x_i with variables x_i^p, it considers binary variables $z_{i,l}^p$ indicating whether the x_i^p are within predefined intervals $\left[\tau_{l-1}^p, \tau_l^p\right]$, $l = 1 : k_p + 1$, $\tau_0 = -\infty$, $\tau_{k_p+1} = \infty$. Furthermore, the total variation of the coefficient vector is minimized instead of its norm. As a result, a sparse scoring system is automatically obtained. To further improve sparsity, the coefficients are iteratively reweighted. To allow the inclusion of interaction, ICS* further expands the binary feature space as follows.

Mapping to a Binary Feature Space Assume an observation $x_i \in \mathbb{R}^d$. The proposed feature map is $(\mathbb{R}^d \to \mathbb{R}^{N_f}) : z_i = \varphi(x_i) = \left[\varphi_{gr_1}(x^{gr_1}) ... \varphi_{gr_p}(x^{gr_p})\right]$, in which $gr \subset \{1, ..., d\}$ can be any subset of the original variables. Hence, z_i is the concatenation of feature maps for every variable and the groups of variables among which interactions should be considered. The feature submaps φ_{gr} expand the space spanned by the variables involved to a multidimensional binary space. The submap for a group gr involving variables $\{p_1, p_2..p_q\}$ contains the following binary features:

$$\begin{aligned} f_{i,l_{p_1}..l_{pq}}^{p_1..p_q} = \quad & I(\tau_{l_{p_1}-1}^{p_1} \leq x_i^{p_1} < \tau_{l_{p_1}}^{p_1}) \qquad (2) \\ \& \quad & ... \\ \& \quad & I(\tau_{l_{pq}-1}^{p_q} \leq x_i^{p_q} < \tau_{l_{pq}}^{p_q}) \\ \text{with} \quad & l_{p_1} \in \{1, .., k_{p_1} + 1\}, ... \\ & l_{p_q} \in \{1, .., k_{p_q} + 1\} \end{aligned}$$

f_i is a multidimensional array indexed by $l_1, .., l_q$. I is a binary indicator using thresholds τ to split the range of each variable. In effect, the space spanned by the original variables is divided into bins based on the thresholds τ. These are initially inferred from the distribution of the data, but in simplification steps included in ICS*, adjacent regions will be merged if possible. Finally, the multidimensional array can be vectorized to yield the feature vector $\varphi_{gr_p}(x^{gr_p})$. These feature vectors are then concatenated to yield the full feature vector z_i.

The resulting optimization problem can be expressed in matrix formulation as:

$$\min_{w,b,\varepsilon} \sum | Dw | + \gamma \varepsilon^T \mathbf{1}, \quad D \in \mathbb{R}^{N_{df} \text{x} N_f}, w \in \mathbb{R}^{N_f} \qquad (3)$$

$$\text{s.t.:} \begin{cases} Y(Zw + b) \geq \mathbf{1} - \varepsilon, & Y \in \mathbb{R}^{N \text{x} N}, Z \in \mathbb{R}^{N \text{x} N_f} \\ \varepsilon \geq \mathbf{0}, & \varepsilon \in \mathbb{R}^N \end{cases}$$

w is a vector containing the coefficients that will be added to the score when the corresponding binary feature in z_i equals 1. D is a matrix defining coefficient differences, Z is the data matrix made up of rows z_i in the binary feature space and Y is a diagonal matrix of class labels. N, N_f and N_{df} are the number of data observations, binary features and coefficient differences, respectively.

The matrix D is necessary to minimize the total variation of the coefficient vector w. Multiplication of D with w yields differences between adjacent bins in the multidimensional representation f_i defined in Equation 2. For example, for the bin $f_{i,l_1..l_q}^{p_1..p_q}$, the matrix D defines q coefficient differences:

$$w_{i,l_1..l_q}^{p_1..p_q} - w_{i,l_1-1..l_q}^{p_1..p_q}, .., w_{i,l_1..l_q}^{p_1..p_q} - w_{i,l_1..l_q-1}^{p_1..p_q}$$

To make sure that the first coefficient of each group equals zero, an additional row with only a single 1 corresponding to the first binary feature of that group is included in D.

Despite sparsity, one can still end up with small steps $w_{i,l_1..l_q}^{p_1..p_q} - w_{i,l_1-1..l_q}^{p_1..p_q}$. From the point of interpretability, less and larger steps are preferred. For this reason, one tries to strike a balance between accuracy, induced by small steps corresponding to local behavior, and interpretability, which benefits from less steps. This trade-off can be achieved by iterative reweighting of the model.

Scoring and Prediction of Probability To convert the model to a scoring system, the coefficients w are normalized and rounded to obtain integer point values s. The score can then be obtained by summing across the binary feature space: score $= s^T z$. Finally, a mapping from scores to probabilities is obtained through application of logistic regression with the scores as only predictor.

Some Remarks on Solving the ICS* Formulation Although the formulation in Equation (3) contains an absolute value, it can be reformulated as a linear programming problem and solved by dedicated solvers. Yet, one should be aware that the size of the system grows with the number of variables in the data set, the number of thresholds τ of each variable and, particularly, the number of required interactions. Solving it is possible because of its inherent sparsity both in the data and in the constraints. This property not only allows storing the system, but it can also be exploited by dedicated solvers.

3 SENSITIVITY STUDY

The sensitivity study performed in this paper is carried out on synthetic data. This allows to know the

correct solution and to insert specific effects. The model can be expressed as

$$p = S(7x_1x_2 + 4x_3^2 + 3x_4 - 3) \quad (4)$$

in which S is the standard logistic (sigmoid) function and p the risk or probability of the data point x belonging to the target class. The model includes two main effects, quadratic in x_3 and linear in x_4, and one interaction involving x_1 and x_2. Data generation is done by randomly generating a pool of independent normally distributed data. Apart from the four required, additional non-informative variables can be added. The basic data set that will be used for the study consists of 250 observations for each class, involving seven variables (four required and three additional). This set will be used in the remainder, unless mentioned otherwise.

The results of applying ICS* on the basic data set are presented in Figure 1. Two third of the data was used for training, one third for testing. The three top parts of the Figure represent the detected effects. The τ values are shown at the borders. The top effect involves x_1 and x_2. Notice the influence of the multiplication. The quadratic and linear effect were correctly detected as well, whereas the three spurious variables were correctly rejected as effects. The bottom part of the Figure is the Risk Profile. It maps the final score obtained by summing over all effects to the probability of belonging to the target class. With this model, ICS* is able to classify the test data with an accuracy of 86.5%, or an Area Under the ROC Curve (AUC) of 0.94.

The sensitivity study consists of five parts. The first part is a simple resampling by cross-validation (cv) of the model data, with and without interactions to investigate the stability of the feature selection. Secondly, the influence of additive white noise will be investigated. Furthermore, the influence of the number of non-informative variables and the training set size are discussed. These last two in particular have an effect on the execution time, the last topic of the study. Unless stated otherwise, the model will be trained on two third of the data whereas one third will be used for testing.

Resampling Resampling is performed to assess the basic stability of the model. If slightly different data is used, to what extent do the detected effects change? The resampling is performed in the structured cross-validation framework (10 folds) in which each part of the data is used for testing in one fold, whereas it is used for training in all other folds. The same 10-fold cross-validation was carried out twice: once with the additional restriction that no interactions should

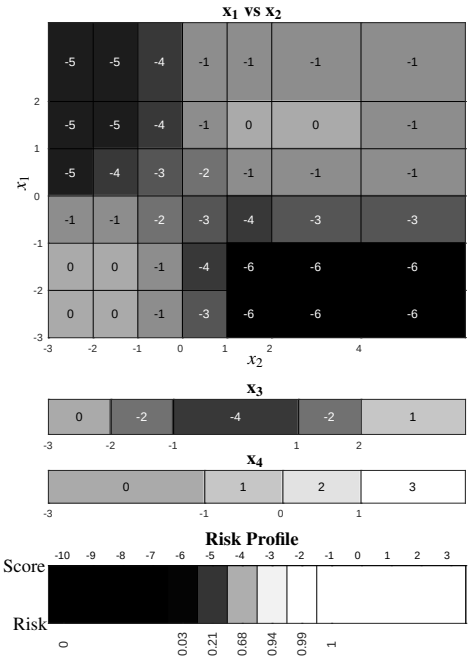


Figure 1: The results of the application of ICS* to the basic synthetic data set.

be investigated (basic ICS), and once including the option for interactions (ICS*).

The discovered effects for the ten folds and corresponding test AUCs are shown in Table 1. Both ICS and ICS* mostly succeed at keeping the relevant effects included. Of course, the interaction effect cannot be discovered by ICS since these effects are not considered in this method. Secondly, ICS has less tendency to include non-informative effects. This was expected, since the number of effects to be considered and the average number of coefficients per effect are much larger for ICS*. For both settings, some overfitting occurs (fold 8 for ICS, folds 5 and 6 for ICS*).

Table 1: Resampling results for ICS|ICS*. For each fold, the detected effects and test AUC (%) are indicated.

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

Non-informative variables in ICS* often include effect 1 and 2. In conjunction with the interaction between 1 and 2, these variables can actually be informative. In other words, these effects could be incorporated into the interaction effect. The same holds for some other effects, e.g. an interaction between 1 and 4 is considered relevant a few times. This still yields a good model. This non-uniqueness of the solution will be further discussed in Section 5. Table 1 shows ICS* obtains a better classifier than ICS. Even when overfitting occurs, the variable coefficients are such that the model still yields good performance. Of course, this comes at the cost of a more complex model.

In conclusion, one could say that resampling can be used to improve robustness of model selection. For the final model, trained on all the (training) data, only the effects that occurred in more than 7 out of 10 folds will be included. Applying this principle for the data presented in Table 1 leads to inclusion of all correct effects. No spurious detected effects are included in ICS and one main effect for variable 2 is included for ICS*. The analyses that follow will only be reported for ICS*. The resampling scheme presented here will be used in the remainder of the experiments.

Influence of Noise The amount of additive white noise can be characterized by the Signal-to-Noise Ratio (SNR), defined as $\text{SNR} = \frac{\sigma_s^2}{\sigma_n^2}$, the ratio of the variances of the signal and the noise. The influence of noise can be shown by comparing the models found for various SNRs. In this study, SNR $\in \{\infty, 50, 25, 10, 5, 4, 3, 2, 1.5\}$ will be considered. A Signal-to-Noise Ratio of ∞ corresponds to the noiseless case. Noise is added to x_i after setting y_i using the model described in (4).

The sensitivity to noise is illustrated in Table 2. For high SNR, all relevant effects are detected, whereas for lower SNR, the interaction effect is lost. This is logical, since both variables are affected by the noise. Although the interaction effect is masked by the noise, ICS* does not model the noise itself. The additional spurious detections for high SNR may seem surprising, but they involve variable 1 and 2, the variables also involved in the interaction. As such, they do contribute to the solution. The last column of the Table highlights the drop in performance when the noise level increase, particularly steep when going from an SNR of 4 to an SNR of 3 due to the loss of the interaction term.

Influence of Non-Informative Variables ICS* is able to exclude non-informative variables from the model. However, a variable can only be excluded if all of its bins in the extended binary feature space have zero coefficients. To quantify the influence of having a higher number of variables, ICS* was applied for an additional number of non-informative variables going from one to ten. The experiments yielded a correct rejection of all non-informative variables whilst keeping the test AUC around 0.9.

Table 2: Influence of noise on the detected effects and test AUC for ICS*.

SNR	Eff [1,2]	Eff 3	Eff 4	#Other Eff	AUC
Inf	✓	✓	✓	2	0.91
50	✓	✓	✓	1	0.86
25	✓	✓	✓	2	0.86
10	✓	✓			0.80
5	✓	✓	✓		0.79
4	✓	✓	✓		0.76
3		✓	✓		0.63
2			✓		0.63
1.5		✓	✓		0.58

Influence of Training Set Size One would expect an improvement in the ability of ICS* to infer a model from the data when the training set size grows. To study this, an independent test set of 150 observations of each class will be considered. The training set is enlarged gradually. Set sizes of 100, 200, 500, 1000, 1500, 2000, 3000 and 5000 with equal contribution of the two classes will be considered.

The influence of the training set size is presented in Table 3. Even when only 100 data points are available, the three effects can be discovered. The one additional effect for a set size of 500 is related to variable 1, which is indeed involved in the model. When one looks in more depth at the generated models for each case, one observes that when the set size grows, the number of binary features for some effects increases, particularly for the interaction. This signifies that although the correct effects are already discovered with less data, the scoring system becomes more refined when more data is added. This is due to the choice of τ (when more data are available, more and small intervals are considered). The impact on per-

Table 3: Influence of the training set size on the detected effects and the test AUC for ICS*.

Size	Eff [1,2]	Eff 3	Eff 4	#Other Eff	AUC
100	✓	✓	✓		0.85
200	✓	✓	✓		0.87
500	✓	✓	✓	1	0.95
1k	✓	✓	✓		0.94
1.5k	✓	✓	✓		0.94
2k	✓	✓	✓		0.94
3k	✓	✓	✓		0.94
5k	✓	✓	✓		0.94

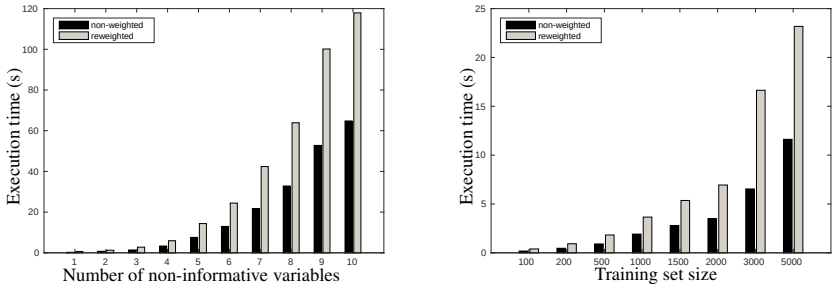


Figure 2: Execution time as a function of the number of non-informative variables (left) and training set size (right).

formance is shown in the last column of the Table: the AUC improves with a growing data set, though even the coarsest model already has an AUC of 0.85. With the data set used for this study, one notices AUC saturation for a set size larger than 500 data points. No information is gained by having a larger data set. Note that these results depend on the complexity of the underlying model and the predefined thresholds τ.

Execution Time The set size and the number of non-informative variables both influence not only the performance of the model in terms of accuracy, but also the problem size. Depending on the method used to solve (3), it can have an impact on execution time. To quantify this, 100 executions of the optimization problem (3) were executed with the training sizes and additional variables as described in the previous subsections. Furthermore, the evolution for the non-weighted and the reweighted case is compared.

Figure 2 shows an exponential increase in the execution time for the weighted and unweighted case for an increasing number of spurious variables, as compared to only a nearly linear increase for the size of the data set. Hence, the impact of the number of spurious variables is dominant over the impact of the training set size. This is due to the combinatorial expansion of the feature space implied by the mapping defined in (2), whereas the linear increase is related to the number of constraints. The issue will be covered in more depth in Section 5. As mentioned, for application purposes this is not a crucial drawback, as long as the problem still fits in memory.

4 APPLICATIONS

Two data sets from the UCI repository (Lichman, 2013) will be used to validate ICS*. The first one is the Mushroom Data Set, the second one is the Vertebral Column Data Set. For the Mushroom set, 90% of the data was used for training and 10% for testing, divided by random sampling. For the Vertebral Column data set, two third of the data was used for training, also randomly sampled. In both cases, important effects were selected by 10-fold cv on the training set, after which the final model was trained on the entire training set.

Mushroom Data Set The mushroom set includes descriptions of 23 species of the Agaricus and Lepiota family (Duch et al., 1997). The aim is to classify them as either edible or poisonous based on 22 nominal attributes. 8124 samples were provided with a class distribution of 51.08% edible and 48.2% poisonous.

It should be noted that the cross-validation was unanimous in the choice of effects to be selected for the final model. This is due to the large number of samples. The obtained test AUC is 0.993. To validate the quality of the model even further, it can be compared to the optimal solution being offered with the data set (Duch et al., 1997). Perfect separation can be obtained using a set of four subsequent rules, given in Table 4. The solution obtained with ICS* corresponds exactly to the first two rules, which are responsible for 99.4% accurate classification on the set as a whole. The reason ICS* does not find all four rules is a limit imposed on its training AUC to avoid trivial overfitting. This could be avoided by interactively selecting this threshold based on the ROC characteristics instead of using an automatic procedure.

Vertebral Column Data Set This data set consists of 310 observations with 6 real-valued biomechanical

Table 4: The optimal solution rules for the mushroom data set.

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

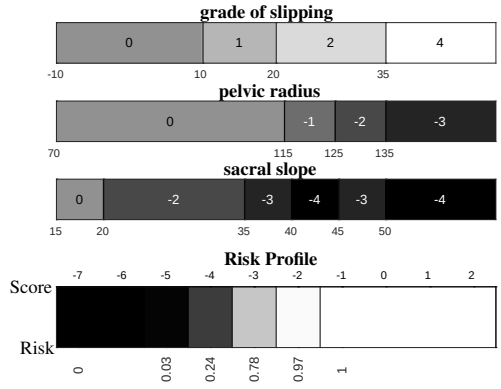


Figure 3: Model for the UCI Vertebral Column data set.

attributes (da Rocha Neto et al., 2011). Class labels distinguish 100 'Normal' from 210 'Abnormal' patients (disk hernia or spondylolisthesis).

ICS* succeeds in deriving a simple model with high performance. Three out of six variables are selected as main effects and no interactions are discovered. Two of the discovered effects were selected in all ten folds of the resampling. The third one was chosen in eight out of ten folds. The final model is visualized in Figure 3. Using ICS*, one obtains a test AUC of 0.89 and a test accuracy of 81.7%. Earlier work on this data set showed that performance can be increased using rejection of data (da Rocha Neto et al., 2011), up to a maximal accuracy of higher than 95%. However, when not taking data rejection into account, their result is only slightly higher than ours. They also report on the classical SVM obtaining an accuracy of 85%. The results are difficult to compare since they perform multiple evaluations on the dataset using resampling and average the result, whereas this paper uses only a single train-test split. A more fair comparison can be performed by applying established techniques directly on the specific train-test split being used here. For that reason, Least-Square Support Vector Machines (LS-SVM) with embedded hyperparameter selection were trained and evaluated (Suykens et al., 2002). LS-SVM with a linear kernel obtained a test AUC of 0.88, whereas the use of an RBF kernel resulted in a test AUC of 0.90. ICS* obtains a similar performance as the LS-SVM approaches, whilst at the same time offering a simple and interpretable model.

5 DISCUSSION

The sensitivity study first showed that resampling can be applied as a method to increase the robustness of ICS and ICS*. Both detect the correct effects, including the interaction in the case of ICS*, but sometimes, other effects are included as well. The threshold for robustness is set arbitrarily for the moment. More elaborate techniques than thresholding should be used to give a statistical justification for the inclusion of an effect. Moreover, there is an additional factor which increases the complexity of the robustness problem. Several times during the presentation of the results, it was mentioned that a spuriously detected effect could be tolerated since it could be included in an intended effect, e.g. the interaction. This is due to the nature of the model. Due to additive formulation, (3) is not strictly convex, leading to a non-unique optimal solution. It might seem unsatisfactory from a programming point of view, but it leaves space for discussion with medical practice, where, in the end, the interpretation will take place. However, if one would aim at uniqueness e.g. for repeated runs and comparisons of the resulting scoring systems, additional steps should be taken. One possible approach works by transformation of the problem. According to literature (Sra, 2006), a problem as (3) can be rewritten such that the unique optimal solution will be the one among the solutions of the original problem with the smallest ℓ_2-norm.

This transformation might also prove useful to alleviate the problems with execution time. An increase in set size yields a same increase in data constraints in (3). On the other hand, adding extra variables yields a combinatorial increase in the dimensionality of the feature space. Currently, the linear programming problem is solved using a standard primal-dual approach. This explains the exponential and linear dependencies shown in Figure 2. Yet, when a dual algorithm could be applied, the dimension of the feature space would become irrelevant. The use of the ℓ_1 norm and the matrix D defining the differences lead to a more difficult entirely dual formulation of the problem. The transformation proposed in (Sra, 2006) yields a standard quadratic program, which is easier to consider in the dual space.

ICS* proved robust to noise. For low SNR, the interaction effect was lost since it was obscured by the noise. However, the noise itself did not influence the model in the sense that no spurious effects were introduced to try to include it.

The assessment of sensitivity with regard to set size and number of spurious variables was positive. Only in one case, an effect was missed. The correct detection of the effects for smaller data sets and the gradual improvement of the model until saturation when more data is available opens perspectives for large-size problems. As sometimes applied in other domains fixed-size approximations could be considered (Suykens et al., 2002).

Another aspect to be discussed is the selection of the thresholds τ defining the binary feature space. For this paper, an automated approach was used, selecting initial thresholds based on the quantiles of the data distribution. In later stages, adjacent binary features thus defined are merged if their coefficients are equal.

6 CONCLUSION

In this paper, ICS* was introduced as an extension of ICS. It allows to infer relevant effects, including interactions, from given data and construct a scoring system by solving a minimization problem. After introduction of the changes applied to ICS, ICS* was subjected to a sensitivity study on synthetic data. The study showed that resampling can be used to improve the robustness of the method. Furthermore, it also indicated robustness to noise, training set size and the number of additional non-informative variables. However, both set size and number of variables were shown to have a large impact on execution time. Finally, ICS* was applied to two UCI data sets with good results.

Future work will investigate the formulation of a more advanced approach to the initial estimation of the τ thresholds. A better estimation of the final thresholds from the beginning reduces the complexity of the problem to be solved, since it relates directly to the dimensionality of the expanded feature space.

Another goal is the formulation of the quadratic transformation of ICS*. This would ensure the uniqueness of the solution for a given data set. Furthermore, row-action methods could be applied to achieve a reduction of the execution time. More generally, approaches other than the LP, e.g. sparse integer solutions, could have interesting characteristics.

Finally, the problem to be solved is essentially a combination of variable selection (sparsity on the level of the original variables) and minimization of the number of steps within each effect (sparsity on the level of coefficient differences). Such a combined criterion can be tackled by methods as group sparse LASSO (Simon et al., 2013) for fast convergence to the optimal solution.

ACKNOWLEDGEMENTS

This research was supported by: Bijzonder Onderzoeksfonds KU Leuven (BOF), Center of Excellence (CoE): PFV/10/002 (OPTEC); KULeuven IDO funding: #3E140722 Sensor-based Platform for the Accurate and Remote monitoring of Kine(ma)tics Linked to E-health (SPARKLE); Belgian Federal Science Policy Office: IUAP #P7/19/ (DYSCO, 'Dynamical systems, control and optimization', 2012-2017). VVB is a postdoctoral fellow of the Research Foundation - Flanders (FWO).

REFERENCES

Chowriappa, P., Dua, S., and Todorov, Y. (2014). Introduction to machine learning in healthcare informatics. In *Machine Learning in Healthcare Informatics*, pages 1–23. Springer.

da Rocha Neto, A. R., Sousa, R., de A. Barreto, G., and Cardoso, J. S. (2011). Diagnostic of pathology on the vertebral column with embedded reject option. In Vitri, J., Sanches, J., and Hernndez, M., editors, *Pattern Recognition and Image Analysis*, volume 6669 of *Lecture Notes in Computer Science*, pages 588–595. Springer Berlin Heidelberg.

Davenport, M., Duarte, M., Eldar, Y., Kutyniok, G., et al. (2012). *Compressed sensing: theory and applications*. Cambridge University Press Cambridge.

Duch, W., Adamczak, R., Grabczewski, K., Ishikawa, M., and Ueda, H. (1997). Extraction of crisp logical rules using constrained backpropagation networks. In *Proc. of the European Symposium on Artificial Neural Networks (ESANN)*.

Jeong, B.-H., Koh, W.-J., Yoo, H., Um, S.-W., Suh, G. Y., Chung, M. P., Kim, H., Kwon, O. J., and Jeon, K. (2013). Performances of prognostic scoring systems in patients with healthcare-associated pneumonia. *Clinical Infectious Diseases*, 56(5):625–632.

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml. last accessed 20/5/2015.

Mounzer, R., Langmead, C. J., Wu, B. U., Evans, A. C., Bishehsari, F., Muddana, V., Singh, V. K., Slivka, A., Whitcomb, D. C., Yadav, D., Banks, P. A., and Papachristou, G. I. (2012). Comparison of existing clinical scoring systems to predict persistent organ failure in patients with acute pancreatitis. *Gastroenterology*, 142(7):1476 – 1482.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*.

Sra, S. (2006). Efficient large scale linear programming support vector machines. In *ECML 2006*, pages 767–774, Berlin, Germany. Max-Planck-Gesellschaft, Springer.

Suykens, J. A., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J., Suykens, J., and Van Gestel, T. (2002). *Least squares support vector machines*, volume 4. World Scientific.

Ustun, B., Trac, S., and Rudin, C. (2013). Supersparse linear integer models for predictive scoring systems. In *Proceeding of the 27th AAAI Conference on Artificial Intelligence (AAAI-13)*, pages 128–130.

Van Belle, V., Van Calster, B., Timmerman, D., Bourne, T., Bottomley, C., Valentin, L., Neven, P., Van Huf-

fel, S., Suykens, J. A. K., and Boyd, S. (2012). A mathematical model for interpretable clinical decision support with applications in gynecology. *PLoS ONE*, 7(3):e34312.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc.

Yang, H.-I., Yuen, M.-F., Chan, H. L.-Y., Han, K.-H., Chen, P.-J., Kim, D.-Y., Ahn, S.-H., Chen, C.-J., Wong, V. W.-S., and Seto, W.-K. (2011). Risk estimation for hepatocellular carcinoma in chronic hepatitis b (reach-b): development and validation of a predictive score. *The Lancet Oncology*, 12(6):568 – 574.