KU LEUVEN

STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

| | |
|---|---|
| **Citation/Reference** | Billiet L., Van Huffel S., Van Belle V., **Interval Coded Scoring Extensions for Larger Problems**, in *Proc. of the 2nd ICTS4eHealth workshop, 22nd IEEE symposium on Computers and Communications (ISCC17)*, Heraklion, Greece, Jul. 2017, pp. 198-203. |
| **Archived version** | Final publisher's version/pdf |
| **Published version** | http://ieeexplore.ieee.org/document/8024529/ |
| **Journal homepage** | http://icts4ehealth.icar.cnr.it/ICTS4eHealth2017/index.html |
| **Author contact** | your email lieven.billiet@esat.kuleuven.be<br>your phone number *+ 32 (0)16 327685* |
| **Abstract** | In many medical problems, clinicians are suffering from information overload. Nowadays, clinical decision support systems (CDSS) are widely available to alleviate this issue. However, they are often not transparent, in contrast to medical scoring systems originating in the clinical world itself. This work presents an extension of Interval Coded Scoring System (ICS), an approach for semi-automatic data-driven extraction of such scoring systems from clinical data. It focuses on two ICS improvements for large or high-dimensional datasets. Firstly, it offers an alternative elastic net implementation. This can be solved efficiently due to equivalence with Support Vector Machines and the existing efficient solver for its primal formulation. Secondly, an informed preselection of the variables allows to lower ICS' computational burden. ICS is applied to problems as diverse as arrhythmia diagnosis, breast cancer prognosis, functional capacity assessment and diagnosis of spinal diseases and obtains good results. In particular, a comparison on the arrhythmia database shows the importance of using the extensions which yield similar performance as the original ICS while shortening the execution time with a factor of ten. |
| **IR** | NA |

*(article begins on next page)*

# Interval Coded Scoring Extensions for Larger Problems

Lieven Billiet*†, Sabine Van Huffel*†, Vanya Van Belle*
*KU Leuven
Department of Electrical Engineering (ESAT)
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
Kasteelpark Arenberg 10 bus 2446
3001 Leuven, Belgium
Email: {lieven.billiet, sabine.vanhuffel, vanya.vanbelle}@kuleuven.be
† imec Leuven

***Abstract*—In many medical problems, clinicians are suffering from information overload. Nowadays, clinical decision support systems (CDSS) are widely available to alleviate this issue. However, they are often not transparent, in contrast to medical scoring systems originating in the clinical world itself. This work presents an extension of Interval Coded Scoring System (ICS), an approach for semi-automatic data-driven extraction of such scoring systems from clinical data. It focuses on two ICS improvements for large or high-dimensional datasets. Firstly, it offers an alternative elastic net implementation. This can be solved efficiently due to equivalence with Support Vector Machines and the existing efficient solver for its primal formulation. Secondly, an informed preselection of the variables allows to lower ICS' computational burden. ICS is applied to problems as diverse as arrhythmia diagnosis, breast cancer prognosis, functional capacity assessment and diagnosis of spinal diseases and obtains good results. In particular, a comparison on the arrhythmia database shows the importance of using the extensions which yield similar performance as the original ICS while shortening the execution time with a factor of ten.**

## I. Introduction

Over the last decades, Machine Learning (ML) and Data Mining techniques have become commonplace in biomedical applications and the clinical environment. Ever more data is available and clinicians need to make decisions based on it. However, an individual cannot easily process such amounts of information anymore. This is called information overload [1]. In such cases, ML serves as a powerful toolset to aid clinicians in diagnosis or prognosis. The ML field has solid foundations and many techniques can be readily applied to new applications. For example, Support Vector Machines and Bayesian classifiers have been used in a medical context [2]. An important drawback of such techniques is their non-transparency: they essentially behave as black boxes. However, clinicians tend to favour interpretable models to assess the relation between the logic of the model and their own insights. Therefore other approaches, such as medical scoring systems, have been used for decades. They simply list risk factors, attributing a number of points to each factor. Summing the points corresponding to each factor yields a final score related to an empirical risk. Examples of such approaches include $CHA_2DS_2$-VASc for atrial fibrillation [3], Glasgow for pancreatitis [4] and CURB-65 for pneumonia [5]. Some of these have been designed based on a statistical analysis or have been verified after construction. Often however, they are powerful rules of thumb, a crude summary of existing medical knowledge on the matter.

Recently, we proposed an approach that combines the strength of the aforementioned techniques: Interval Coded Scoring (ICS) [6], [7]. It uses sparse optimization to extract a scoring system from clinical data. In contrast to the typical existing scoring systems, it also supports variable interactions. It uses the same rationale as the work of Ustun et al. [8], but ICS goes beyond a linear model. As will be explained in Section II, it performs feature selection and rejection at the level of variable intervals, not just at the level of variables. Furthermore, it connects to the general literature on sparse optimization including LASSO [9], elastic nets [10] and compressed sensing [11].

The aim of this work is to introduce two extensions of ICS to address its relatively high computational burden. These allow to tackle larger problems. Depending on the specific problem and the settings, the current version is in practice restricted to about 50 variables and about 10,000 observations if one wants a result in less than a few hours.

The remainder of the work is structured as follows. Section II provides a short summary of the main ICS framework, followed by a description of the extensions. Then, Section III shows a comparison between the current and extended version of ICS on synthetic data and a clinical application. Both execution time and the obtained model will be considered.

This work received funding from Bijzonder Onderzoeksfonds KU Leuven (BOF): Center of Excellence (CoE) # PFV/10/002 (OPTEC), SPARKLE Sensor-based Platform for the Accurate and Remote monitoring of Kinematics Linked to E-health #IDO-13-0358; Belgian Federal Science Policy Office: IUAP #P7/19/ (DYSCO, 'Dynamical systems, control and optimization', 2012-2017); European Research Council: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Advanced Grant: BIOTENSORS (nr 339804). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

978-1-5386-1629-1/17/$31.00 ©2017 IEEE

Section IV consists of a discussion. Finally, Section V summarizes the most important findings of this work.

## II. Methods

In a general classification problem, one has a data matrix $X \in \mathbb{R}^{N \times N_v}$, consisting of $N$ observations and $N_v$ variables, and a label vector $\mathbf{y} \in \mathbb{R}^N$. ICS is a way to obtain a model to predict the label for new observations. Its current approach will be sketched in part II-A of this Section. Part II-B introduces preselection, a selection technique aiming at an informed reduction of the number of variables before the core of ICS is executed. Finally, part II-C presents Support Vector Elastic Net. It reformulates the ICS core to reduce its computational burden.

### A. ICS summary

The original ICS formulation is based on the soft-margin Support Vector Machine (SVM) framework [12]. For details of its implementation, see [6]. Similar to SVM, it is a binary classification framework trying to strike a balance between the classification error and a second objective. Also similar to SVM, it uses a mapping $Z = \varphi(X)$ to convert the $N_v$ variables $x^p$ ($p \in 1..N_v$) into $N_f$ features $z^l$ ($l \in 1..N_f$). Taking this into account, the core of ICS can be expressed as:

$$\begin{array}{lll} \min\limits_{w,b,\varepsilon} & \|D\mathbf{w}\|_1 + \gamma\varepsilon^T\mathbf{1}, & D \in \mathbb{R}^{N_{df} \times N_f}, \mathbf{w} \in \mathbb{R}^{N_f} \\ s.t. & Y(Z\mathbf{w} + b) \geq \mathbf{1} - \varepsilon, & Y \in \mathbb{R}^{N \times N}, Z \in \mathbb{B}^{N \times N_f} \\ & \varepsilon \geq \mathbf{0}, & \varepsilon \in \mathbb{R}^N \end{array} \tag{1}$$

Here, $\mathbb{B}$ is the set of binary numbers, $N$ denotes the number of observations and $N_f$ the number of features. $Y$ is a diagonal matrix of class labels $y_{ii} \in [-1, 1]$. $Z = \varphi(X)$ is the binary data matrix, a transformation of the original data matrix $X$. $\mathbf{w}$ is the weight vector, $b \in \mathbb{R}$ the bias term and $\varepsilon$ the vector of (positive) error values with respect to the classification margin, as in the original SVM formulation. Finally, $D$ is a difference matrix (discussed further) defining $N_{df}$ feature differences and $\gamma$ is a regularization parameter. Although very similar to SVM, ICS introduces some adaptations. Firstly, it uses a specific feature map $\varphi$. Secondly, it changes the objective function. Thirdly, it presents the model in an interpretable way. These aspects will be introduced one by one, followed by a discussion of the overall ICS flow.

*1) Feature map:* ICS transforms categorical, integer or continuous data values to a binary representation. Every variable $x^p$ in the data set $X \in \mathbb{R}^{N \times N_v}$ is divided into a number $n_p$ of bins or variable intervals based on thresholds $\tau_1^p$,$\tau_2^p$, …, $\tau_{n_p-1}^p$ depending on the data distribution. Each of these bins is represented by a binary feature. Hence, every observation $x_i^p$ of a variable can be represented by a binary vector $z_i^p$ with only a single 1 corresponding to the bin of the observed value. The binary vectors corresponding to all $N_v$ variables in X can be concatenated into a single vector for every observation. Summarizing:

$$\begin{array}{c} z_i = [z_i^1, z_i^2, \ldots z_i^{N_v}] \quad \text{with} \quad z_i^p = \varphi^p(x_i^p) \\ \varphi^p(x_i^p) = [x_i^p < \tau_1^p, \tau_1^p \leq x_i^p < \tau_2^p, \ldots \tau_{n_p-1}^p \leq x_i^p] \end{array} \tag{2}$$

The mapping has been extended in a straightforward way for variable interactions [7], but this work will be limited to main effects as described in (2).

*2) Objective function:* In SVMs, one maximizes the margin between the given classes. This corresponds to minimizing $\|\mathbf{w}\|_2^2$, balanced by the classification error. ICS minimizes $\|D\mathbf{w}\|_1$ instead. The matrix $D$ encodes the differences in weights corresponding to adjacent bins of one variable and this for all variables. It is a sparse matrix consisting of rows with a single 1 and a single -1. The $\ell_1$ norm imposes sparsity, in this case on the differences between the bin weights. Its effect is twofold. Firstly, if all differences of the weights $w$ corresponding to the bins of a variable $x^p$ are zero, this variable can be discarded. Hence, it implements feature selection. Secondly, it favours equal weights for adjacent bins, allowing them to be joined. Only if necessary, bin weights will be different. As a result, the approach yields simple models. In the literature, the problem is also known as total variation minimization (TV) [13]. The model is simplified even more by repeating the above procedure iteratively with weighted differences $D\mathbf{w}$ to smooth them out even further. The reweighting coefficients $\chi$ are based on the absolute value of the differences. The obtained level of smoothing is a tradeoff between model performance and simplicity. It can be decided automatically with a threshold, but in general, the user should make this decision.

*3) Visualization:* The obtained model can be shown in an interpretable way that makes it easily applicable to new data. It highlights the important variable intervals and their weight in the final decision. Examples of the visualization will be shown in Sections III.

*4) Algorithm:* After listing the main components, the ICS flow can be summarized as in Figure 1. The hyperparameter $\gamma$ is selected by simulated annealing. Lines 6 to 13 display the iterative reweighting. At every iteration, the variables with only zero weights are removed, as explained earlier. $\mathbf{w}$, $Z$ and $D$ are updated by removing the columns corresponding to the bins of the removed original variables $x^p$. The parameter $a$ defines the level of smoothing of the solution. Once the final $\mathbf{w}$ has been obtained, it can be scaled and rounded to yield the final scoring system, ready for visualization. Additionally, logistic regression is used to map the occurring scores to a risk estimation.

### B. Preselection

The approach as described above in (1) can be converted to a linear programming problem and can be solved by any available solver. Therefore, we will refer to it as `lpICS`. The size of the problem is dominated by the number of binary features in $Z$. A very high-dimensional dataset and many bins per variable can lead to huge problems. Although many solvers can solve it efficiently by exploiting sparsity, large problems still pose a heavy computational and memory load.

A first way to address this is to diminish the dimensionality of the data. Many variable selection techniques exist. However,

1: Construct $\tau^p \quad \forall x_p, \quad p \in [1, N_v]$
2: Construct $D$
3: $Z \leftarrow \varphi(X, \tau)$ as in (2)
4: $\mathbf{w}, b \leftarrow$ solve (1)
5: $\mathbf{w}, Z, D \leftarrow$ nonEmptyVars($\mathbf{w}, Z, D$)
6: **repeat**
7: $\quad n_{old} \leftarrow$ #vars($\mathbf{w}$)
8: $\quad$ select $a$
9: $\quad \chi$ = 1./($\epsilon + a|D\mathbf{w}|$)
10: $\quad \chi_m$ = diag($\chi$)
11: $\quad \mathbf{w}, b \leftarrow$ solve (1), with $\|D\mathbf{w}\|_1$ replaced by $\|\chi_m D\mathbf{w}\|_1$
12: $\quad \mathbf{w}, Z, D \leftarrow$ nonEmptyVars($\mathbf{w}, Z, D$)
13: **until** $n_{old}$ == #vars($\mathbf{w}$)

Fig. 1. The original ICS core algorithm (`lpICS`)

the aim is to still take into account the importance of the bins. Hence, the proposed preselection is a four-step procedure.

1) The data transformation in (2) is performed on the initial dataset $X$. Using the computed $Z$ and the known labels, one can solve a standard Support Vector Machine classification problem yielding weights $\omega$. The result is a crude scoring system without the sparsity induced by the weight differences.
2) A new data set $X_2$ is constructed by replacing the variable values in $X$ by their weights $\omega$ obtained through the crude scoring. As a result, $X_2 \in \mathbb{R}^{N \times N_f}$, the same size as $X$. This approach can be considered as a data-driven discrete non-linear transformation. $X_2$ encodes data trends for each variable.
3) The goal remains to eliminate certain variables. To obtain this, any feature selection method could be applied on the trend-informed data $X_2$. Elastic net was chosen due to its further use in other optimization steps (see further). The selection is semi-automatic: the user can decide on the sparsity based on cross-validated performance estimations.
4) Finally, the set $S$ of variables selected from $X_2$ is taken from $X$ to define the data used after preselection: $X = X(:, S)$. This reduced data set can subsequently be used for the main ICS flow in Figure (1).

### C. Support Vector Elastic Net (SVEN)

Another way to address high-dimensionality is to solve (1) using an approach that is insensitive to it, such as the dual approach for SVMs. However, due to the $\ell_1$ constraint it is impossible to apply the kernel trick in the same efficient way since the dual is no longer quadratic. Yet, there is a way around this by reformulating both the optimization problem and the feature map. With these new building blocks, the ICS flow using elastic net (`enICS`) can be formulated.

*1) Elastic net formulation:* We can restate the ICS problem as a (generalized) elastic net [10] with a sparsity-inducing regularization penalty on $D\mathbf{w}$:

$$\min_{\mathbf{w},b} \|Z\mathbf{w} + b - \mathbf{y}\|_2^2 + \epsilon\|D\mathbf{w}\|_2^2 \quad \text{s.t.} \quad \|D\mathbf{w}\|_1 \leq t \qquad (3)$$

Here, we balance the squared error with regard to the labels by both an $\ell_2$ and $\ell_1$ penalty. Without the $\ell_2$ penalty, this reduces to the LASSO [9]. It is included for the stability of the problem since it imposes strict convexity. Since we are mainly interested in the sparsity obtained by the $\ell_1$ penalty, the $\ell_2$ weight is reduced. In our applications, typically $\epsilon = 0.05$.

*2) Data transformation:* The formulation in (3) cannot easily be solved by standard elastic net solvers. On the one hand due to the presence of $D$ in the penalties, but also, as before, due to the sheer size of the problem. The former can be addressed by a change of the original feature map to incorporate the differences of the bins in the data itself. Afterwards, the $\mathbf{w}$ values can be reconstructed from the differences by simply taking the cumulative sum. Therefore, we propose as new feature map $A = \phi(X) \in \mathbb{B}^{N \times N_{df}}$:

$$\begin{aligned} a_i &= [a_i^1, a_i^2, \ldots a_i^{N_v}] \quad \text{with} \quad a_i^p = \phi^p(x_i^p) \\ \phi^p(x_i^p) &= [1, x_i^p \geq \tau_1^p, x_i^p \geq \tau_2^p, \ldots, x_i^p \geq \tau_{n_p-1}^p] \end{aligned} \qquad (4)$$

Notice the changes in the second line. Part of the sparsity is lost, since every variable value is now encoded with a 1 for all variable bins whose left threshold is less than or equal to the value under consideration. The left threshold of the first bin of a variable is considered $-\infty$. Using the new data notation $A$, one can rewrite 3 as:

$$\min_{\mathbf{q},b} \|A\mathbf{q} + b - \mathbf{y}\|_2^2 + \epsilon\|\mathbf{q}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{q}\|_1 \leq t \qquad (5)$$

with $q \in \mathbb{R}^{N_{df}}$ the new vector of unknowns. $\mathbf{w}$ can be reconstructed from it by a binary per-variable summing matrix $D_2 \in \mathbb{B}^{N_f \times N_{df}}$ as $\mathbf{w} = D_2\mathbf{q}$.

*3) Solving the problem:* Zhou et al. [14] show that that the constrained quadratic problem expressed in (5) is equivalent to a dual SVM notation, giving rise to the Support Vector Elastic Net method (SVEN). The corresponding primal SVM has only $N$ parameters, that is, as many as the original number of observations in $X$. Hence, the problem should be solved in the primal to reduce its complexity, e.g. using an iterative conjugated gradient approach as suggested by Chapelle [15]. In case of data sets with a large number of observations $N$, fixed-size methods can be used [16].

Bringing it all together, the algorithm based on elastic net (`enICS`) is expressed in Figure 2. The eventual outcome $\mathbf{w}$ needs only be reconstructed in the end (line 14). The selection of variables to be discarded can easily be performed based on $\mathbf{q}$. Yet, $D_2$ still needs to be updated in lines 5 and 12. Also, note the changes in the iterative reweighting. In particular, line 9 in Figure 1 and Figure 2: since in `enICS` the data matrix is being adapted instead of the penalty, the weights should be the inverse compared to their `lpICS` counterparts.

## III. Experiments

The impact of replacing `lpICS` by `enICS` will be demonstrated based on a synthetic dataset. Both the timing and the obtained model will be compared. Furthermore, a test on a clinical dataset on arrhythmia will highlight the impact of the preselection. In what follows, *weight* will refer to the number

1: Construct $\tau^p \quad \forall x_p, \quad p \in [1, N_f]$
2: Construct $D_2$
3: $A \leftarrow \phi(X, \tau)$ as in (4)
4: $\mathbf{q}, b \leftarrow$ solve (5) using SVEN
5: $\mathbf{q}, A, D_2 \leftarrow$ nonEmptyVars($\mathbf{q}, Z, D$)
6: **repeat**
7: $\quad n_{old} \leftarrow$ #vars($\mathbf{q}$)
8: $\quad$ select $a$
9: $\quad \chi = a|\mathbf{q}|$
10: $\quad \chi_m = \text{diag}(\chi)$
11: $\quad \mathbf{q}, b \leftarrow$ solve (5), with $A$ replaced by $A\chi_m$
12: $\quad \mathbf{q}, A, D_2 \leftarrow$ nonEmptyVars($\mathbf{q}, Z, D$)
13: **until** $n_{old}$ == #vars($\mathbf{q}$)
14: Derive $\mathbf{w} = D_2\mathbf{q}$

Fig. 2. ICS core using SVEN (`enICS`)

of score points corresponding to a bin, whereas the *score* is the summation of the weights by applying the scoring system. A (main) *effect* refers to the weight distribution over all bins of a single variable e.g. a linear effect.

### A. Experiments on synthetic data

*1) The synthetic dataset:* A binary classification problem was created according to the following formula:

$$y = (x_1 + x_2^2 + 2\text{sign}(x_3)) > 0 \tag{6}$$

It involves a linear effect, a quadratic effect and a step function. $x_1$, $x_2$ and $x_3$ are sampled from a standard normal distribution $N(0, 1)$. Additional confounding variables, also drawn from $N(0, 1)$, are added during the experiments.

*2) Timing:* This first experiment directly compares the speed of (1) and (5), that is, the core of both methods. To this end, 2000 data points, 1000 from each class, are generated. The problem is solved for an increasing number of confounding variables ranging from 0 to 45. For each added variable the median time of 50 executions is reported to account for the (small) variance in the results.

Figure 3 shows the impact of using `enICS` instead of `lpICS`. The latter's execution time increases exponentially, as indicated by the dashed exponential fit. It starts at a fraction of a second if only the three informative variables are used and grows to several seconds if more than 25 confounders are added. In contrast, `enICS` grows slowly, linearly, and remains under 100ms, starting around 3ms. This shows its ability to deal with high-dimensional data.

*3) Model comparison:* The second experiment looks at the models obtained by `lpICS` and `enICS` in the full ICS workflow without preselection, as in the algorithms shown in Figures 1 and 2. 20 bins have been used per variable. Furthermore, 10 confounding variables have been added. Again, 2000 data samples are generated for training. Additionally, 500 other data samples are used to assess the methods' performance on independent data.

The obtained models with `lpICS` and `enICS` are shown in Figures 4 and 5, respectively. Each row represents a selected

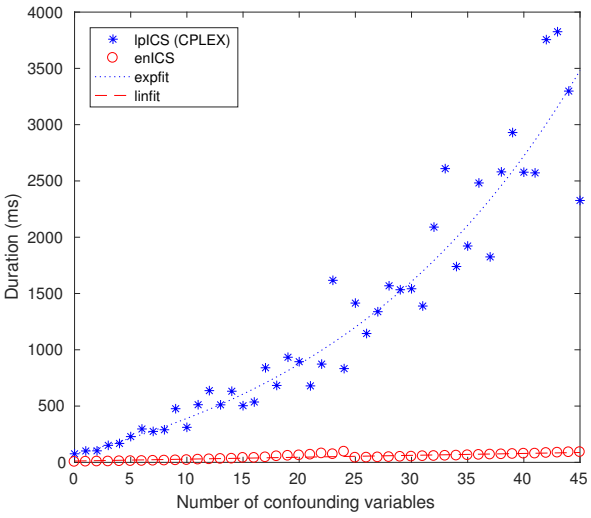



Fig. 3. Comparison of `lpICS` and `enICS` execution time.

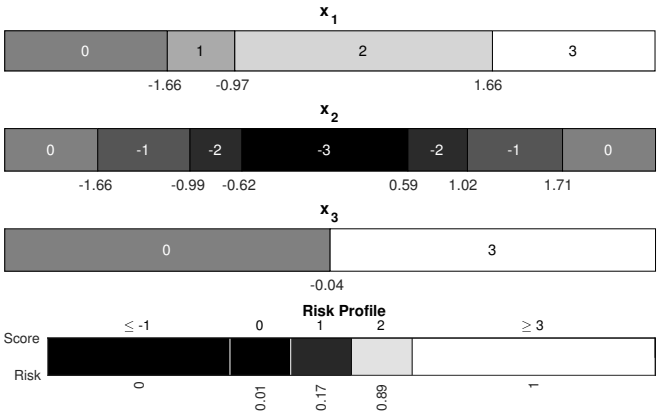



Fig. 4. Model for the synthetic dataset resulting from `lpICS`.

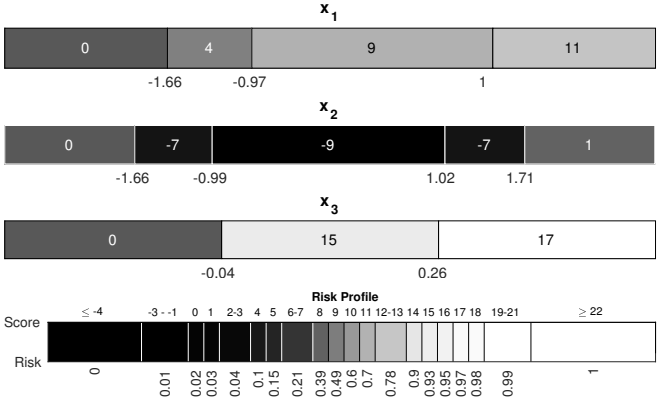



Fig. 5. Model for the synthetic dataset resulting from `enICS`.

variable, the last row shows the conversion from the score (top) to the associated risk (bottom). For selected variables, the representation shows their division in bins, the thresholds and the weight of each bin. These weights are also colour-coded: similar colours represent weights that are close to each other. As an example, consider an observation with $x_1 = 1$, $x_2 = 0.1$ and $x_3 = 0$. It can be classified by the `lpICS` model from Figure 1. The top bar represents $x_1$: a value of 1 is in between -0.97 and 1.66. Hence, the attributed number of points is 2. Similarly, $x_2 = 0.1$ yields -3 points due to the second bar and $x_3 = 0$ yields 3 points according to the third bar. Hence, the total score is 2 + -3 + 3 = 2. The lowest part, the risk profile, converts this to a probability of 89% of belonging to the positive class.

Both models correctly eliminate the confounders and detect $x_1$, $x_2$ and $x_3$ as the main effects. Both also show the defined linear trend, quadratic effect and the step. In case of `enICS`, the latter has been approximated with an intermediate interval. `lpICS` has a test AUC of 0.977 and a test accuracy of 95.6%. `enICS` has approximately the same performance: its test AUC is 0.981 and its accuracy 93.8%. It uses larger score values to increase contrast, yielding a more complicated risk profile. However, the total number of intervals, an indicator for the model complexity, is actually lower for `enICS`. Also, in some cases, the more detailed risk profile might be desirable as well.

### B. Arrhythmia dataset

This experiment makes use of the 12-lead ECG arrhythmia dataset of the UCI Machine Learning database [17], [18]. It contains records of 452 patients classified as either normal or one of several classes of arrhythmia. Here, the problem is considered as binary: normal vs any of the arrhythmia classes. 80% of the data is selected by stratified sampling for training. The remainder is used for testing. Each patient record contains 279 features, 206 continuous and 73 binary. 5 of them are removed due to missing values. Feature details can be found in [18]. This high-dimensional dataset is used to assess the impact of the proposed measures on the complete ICS workflow, as in the previous experiment: both the choice of type (en/lp) and the presence of preselection will be evaluated. Automatic tuning will be used during iterative reweighting since the semi-automatic mode would disrupt time measurement. The number of bins per variable was set to 7.

The test results are shown in Table I. Notice the long computation time for `lpICS`. It can be improved drastically by either switching to `enICS` or by using the preselection, which reduces the time by a factor of 53.7 and 19.1, respectively. The preselection reduced the number of variables from 254 to 28. One can run it for `enICS`, but this appears counterproductive. The execution time increases, because the preselection approach is slower than direct `enICS`. This might be due to the specific SVM-package used for preselection. The AUC of `enICS` can be improved by preselection, but its accuracy reduces.

Overall, `lpICS` tends to be geared towards simpler methods than `enICS` when used in automatic mode as can be deduced from the number of intervals. This can also be connected to the findings on the synthetic dataset e.g. in the detection of the step of variable $x_3$. Also, `enICS` tends to have a higher final weight variability, leading to a more complicated risk profile since all possible scores have to be listed. As mentioned before, this can be an advantage if one looks for a more detailed risk assessment.

TABLE I
TEST RESULTS OF THE APPROACHES ON THE ARRHYTHMIA DATABASE. *pre* INDICATES THAT PRESELECTION WAS USED.

| | **lpICS** | **enICS** | **lpICS-pre** | **enICS-pre** |
|---|---|---|---|---|
| **Duration (s)** | 852.14 | 15.88 | 44.71 | 24.73 |
| **#variables** | 4 | 4 | 4 | 3 |
| **#intervals** | 8 | 11 | 8 | 11 |
| **AUC** | 0.816 | 0.772 | 0.803 | 0.799 |
| **Acc (%)** | 82.2 | 77.8 | 81.1 | 74.4 |

TABLE II
RESULTS OF THE APPROACHES ON THE ARRHYTHMIA DATABASE. *pre* INDICATES THAT PRESELECTION WAS USED.

| **lpICS** | **enICS** | **lpICS-pre** | **enICS-pre** |
|---|---|---|---|
| R' width (V1) | R' width (V1) | R' width (V1) | R' width (V1) |
| T ampl (V6) | T ampl (V6) | T ampl (V6) | T ampl (V6) |
| S ampl (V3) | Q ampl (V3) | S width (V3) | QRS duration |
| R' width (V2) | Q width (V2) | Age | |

Another observation is that the models change when the problem changes either in formulation or in the variables involved. For example, after preselection, `lpICS` finds a different model than before although its complexity is identical and its performance similar. This can partly be explained by the fact that the problem solution is not unique to begin with. However, another contributing factor is the use of the automatic mode. In this case, a cutoff factor (default 0.95) defines the sacrifice with respect to the maximal AUC to obtain a simpler model. Alternatively, the user can intervene and steer the algorithm for both preselection and iterative reweighting to the desired tradeoff (semi-automatic mode).

We can also understand the change in model when looking more in detail at the selected variables for all investigated methods in Table II. The details of the features can be found in [18]. They mostly refer to elements of the ECG waveforms with the ECG lead added between brackets. It can easily be seen that two of the effects have been selected in all cases. The performance is increased by adding some other variables. However, inspection shows that these are removed in the preselection phase. The same variance can probably be explained by other minor variables as well, hence their selection. A larger number of samples might alleviate this issue.

## IV. DISCUSSION

One can debate whether computational speed is of much importance while solving ICS problems. Indeed, once the model has been constructed, it can be evaluated without the need for a computer. However, when aiming for knowledge discovery, it might need to be evaluated many times. Further-

more, the LP results above were obtained using IBM's CPLEX solver which is already an order of magnitude faster than e.g. Matlab's built-in function *linprog*. Finally, the arrhythmia problem shown above is still only average-sized. A study with over 60,000 samples and hundreds of variables typically takes hours to complete when solved with `enICS`. Using `lpICS` without preselection is simply not an option as it would take days. From this point of view, the improvement is of vital importance.

The arrhythmia example also shows the importance of the semi-automatic intervention when building a model. Many models of similar performance can be obtained. Therefore, it is important to critically guide the process towards a stable model with an acceptable performance-complexity tradeoff for the application at hand. It further indicates the role of dataset size in this tradeoff: the smaller the set size, the more careful one has to consider discovered effects. Stability under changing conditions or data subsets is a more reliable indication of the true impact of a discovered effect. If the model is more stable, it is more likely that detected effects are indeed clinically relevant.

The summary of ICS mentioned that interaction effects would not be considered. Nevertheless, ICS does support them. Preselection of variables can and has been extended to include their use. However, the second extension, `enICS`, does not support them. The transformation in (4) cannot be uniquely defined for interactions. In other words, $D$ in (3) cannot be inverted. This is a known difficulty in total variation minimization e.g. for image denoising. We are currently investigating ways to extend `enICS` to include interactions as well.

Taking all experiments into account, it is recommended to choose one of the available optimizations presented in this paper. It is not advantageous to combine them. If interactions are needed, `lpICS-pre` should be selected. Otherwise, `enICS` is also an option, possibly leading to a more complicated risk profile due to a larger range in the detected weights.

This paper showcases the use of ICS on synthetic data and a clinical problem using a public dataset. During its development, its use has been demonstrated on several other problems with a particular interest in medical applications. Examples include breast cancer prognosis, ovarian cancer diagnosis, prediction of viable pregnancies [19], vertebral column disease diagnosis [7] and activity capacity assessment in spondyloarthritis patients [20].

## V. Conclusion

Interval Coded Scoring was introduced as an approach to derive transparent scoring systems to use as Decision Support System, primarily aiming at a clinical setting. This paper presented two technical additions to decrease the framework's computational burden. Both `enICS` and preselection were proven to provide a major improvement in computational effort, whilst still deriving equally performant models on both synthetic data and a clinical database. Other applications were listed as well. In the future, `enICS` will be expanded towards variable interactions.

## Acknowledgments

We graciously thank all other members of the SPARKLE group: W. Dankaerts, K. de Vlam, J. Geuens, L. Geurts, R. Puers, S. Seerden, V. Van den Abeele, B. Vanwanseele and R. Westhovens.

## References

[1] B. Gross, *The Managing of Organizations: The Administrative Struggle*. Free Press of Glencoe, 1964, no. 2, p. 857.

[2] P. Chowriappa, S. Dua, and Y. Todorov, "Introduction to machine learning in healthcare informatics," in *Machine Learning in Healthcare Informatics*. Springer, 2014, pp. 1–23.

[3] G. Y. Lip, R. Nieuwlaat, R. Pisters, D. A. Lane, and H. J. Crijns, "Refining clinical risk stratification for predicting stroke and thromboembolism in atrial fibrillation using a novel risk factor-based approach: The euro heart survey on atrial fibrillation," *Chest*, vol. 137, no. 2, pp. 263–272, 2010.

[4] R. Mounzer, C. J. Langmead, B. U. Wu, A. C. Evans, F. Bishehsari, V. Muddana, V. K. Singh, A. Slivka, D. C. Whitcomb, D. Yadav, P. A. Banks, and G. I. Papachristou, "Comparison of existing clinical scoring systems to predict persistent organ failure in patients with acute pancreatitis," *Gastroenterology*, vol. 142, no. 7, pp. 1476–1482, 2012.

[5] B.-H. Jeong, W.-J. Koh, H. Yoo, S.-W. Um, G. Y. Suh, M. P. Chung, H. Kim, O. J. Kwon, and K. Jeon, "Performances of prognostic scoring systems in patients with healthcare-associated pneumonia," *Clinical Infectious Diseases*, vol. 56, no. 5, pp. 625–632, March 2013.

[6] V. Van Belle, B. Van Calster, D. Timmerman, T. Bourne, C. Bottomley, L. Valentin, P. Neven, S. Van Huffel, J. A. K. Suykens, and S. Boyd, "A mathematical model for interpretable clinical decision support with applications in gynecology," *PLoS ONE*, vol. 7, no. 3, p. e34312, 2012.

[7] L. Billiet, S. Van Huffel, and V. Van Belle, "Interval coded scoring index with interaction effects: a sensitivity study," in *5th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, Rome, February 2016, pp. 33–40.

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

[9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J R Stat Soc Series B Stat Methodol*, vol. 58, no. 1, pp. 267–288, 1996.

[10] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J R Stat Soc Series B Stat Methodol*, vol. 67, no. 2, pp. 301–320, 2005.

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

[12] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.

[13] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J Imaging Sci*, vol. 1, no. 3, pp. 248–272, 2008.

[14] Q. Zhou, W. Chen, S. Song, J. Gardner, K. Weinberger, and Y. Chen, "A reduction of the elastic net to support vector machines with an application to gpu computing," in *AAAI Conference on Artificial Intelligence*, 2015.

[15] O. Chapelle, "Training a support vector machine in the primal," *Neural Computation*, vol. 19, pp. 1155–1178, 2006.

[16] J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, J. Suykens, and T. Van Gestel, *Large Scale Problems*. World Scientific, 2002, vol. 4, ch. 6, pp. 173–200.

[17] M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2013, last accessed 20/5/2015.

[18] H. A. Guvenir, B. Acar, G. Demiroz, and A. Cekin, "A supervised machine learning algorithm for arrhythmia analysis," in *Computers in Cardiology 1997*, Sep 1997, pp. 433–436.

[19] V. Van Belle, "Non-linear survival models and their application within breast cancer prognosis," Ph.D. dissertation, KU Leuven, December 2010.

[20] L. Billiet, T. W. Swinnen, R. Westhovens, K. de Vlam, and S. Van Huffel, "Accelerometry-based activity recognition and assessment in rheumatic and musculoskeletal diseases," *Sensors*, vol. 16, no. 12, p. 2151, 2016.