A Provable Algorithm for Learning Interpretable Scoring Systems

Nataliya Sokolovska
University Paris 6, INSERM, France

Yann Chevaleyre
University Paris Dauphine, France

Jean-Daniel Zucker
IRD Bondy, INSERM, France

Abstract

Score learning aims at taking advantage of supervised learning to produce interpretable models which facilitate decision making. Scoring systems are simple classification models that let users quickly perform stratification. Ideally, a scoring system is based on simple arithmetic operations, is sparse, and can be easily explained by human experts.

In this contribution, we introduce an original methodology to simultaneously learn interpretable binning mapped to a class variable, and the weights associated with these bins contributing to the score. We develop and show the theoretical guarantees for the proposed method. We demonstrate by numerical experiments on benchmark data sets that our approach is competitive compared to the state-of-the-art methods. We illustrate by a real medical problem of type 2 diabetes remission prediction that a scoring system learned automatically purely from data is comparable to one manually constructed by clinicians.

1 Introduction

Scoring systems are simple linear classification models that are based on addition, subtraction, and multiplication of a few small numbers. These models are applied to make quick predictions, without use of a computer. Traditionally, a problem in supervised machine learning is cast as a binary or multi-class classification where the goal is to learn real-valued weights of a model. However, although the generalizing error is an important criterion, in some applications, the interpretability of a model plays even a more significant role. Most machine learning methods produce highly complex models, not designed to provide explanations about predictions.

Variable	Thresholds	Score
Age	<40	0
	40–49	1
	50 – 59	2
	>60	3
Glycated hemoglobin	<6.5	0
	6.5 – 6.9	2
	7 – 8.9	4
	> 9	6
Insuline	No	0
	Yes	10
Other drugs	No	0
	Yes	3

Classify as `Remission` if sum of scores < 7
Classify as `Non-remission` if sum of scores ≥ 7

Table 1: The DiaRem Score to assess the outcome of the bariatric surgery [24]

Clinical scoring systems are of particular interest since they are expected to predict a state of a patient and to help physicians to provide accurate diagnostics. An example of such a score, shown in Table 1, is the DiaRem score [24] which is a preoperative method to predict remission of type 2 diabetes after a gastric bypass surgery. The DiaRem is based on four clinical variables and a few thresholds per variable. Only one arithmetic operation is involved into the DiaRem computation: the scores are added, and if the sum is < 7, then a patient is likely to benefit from the surgery and to get the diabetes remission. Some other widely used medical scores are SAPS I, II, and III [10, 21] and APACHE I, II, III to assess intensive care units mortality risks [14], CHADS$_2$ to assess the risk of stroke [9]; TIMI to estimate the risk of death of ischemic events [2]. Despite widespread use in clinical routines, there has been no principled approach to learn scores from observational data. Most of existing clinical scores are built by a panel of experts, or by combining multiple heuristics.

In many applications, although continuous features are available for a prediction task, it is often beneficial to use discretized features or categories. Predictors that use categorical variables need smaller memory footprint, are easier

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

to interpret, and can be applied directly by a human expert to make a new prediction. The difficulty to learn discrete classifiers is well known (see, e.g., [4]): minimizing a convex loss function with discrete weights is NP-complete.

In this paper, we propose a principled approach to learn discrete scoring systems. Our approach is unique since it learns both the thresholds to discretize continuous variables, and the weights for the corresponding bins. The weights can be also discretized with a randomized rounding after training. To our knowledge, this paper is the first attempt to learn a discrete scoring system which relies on simultaneous learning of bins and their corresponding scores.

The algorithm we provide has the best of two worlds: accuracy and interpretability. It is fully optimised for feature selection, and it converges to an optimal solution.

This paper is organised as follows. We discuss the related work in Section 2. In Section 3, we introduce the novel algorithm and show its theoretical properties. The results of the numerical experiments are discussed in Section 4. Concluding remarks and perspectives close the paper.

2 Related Work

Our contribution is related to the new methods for interpretable machine learning. The SLIM (Supersparse Linear Integer Models) [27] is formulated as an integer programming task and optimizes directly the accuracy, the 0-1 loss, and the degree of sparsity. However, optimizing the 0-1 loss is NP-hard even with continuous weights, and training of a SLIM model on a large data set can be challenging.

Another modern avenue of research are Bayesian-based approaches to learn scoring systems. So, [7] introduced a Bayesian model where a prior favours fewer significant digits, and, therefore, the solution is sparse. A Bayesian model is also developed in [29] to construct a falling rule list, which is a list of simple if-then rules containing a decision-making process and which stratifies patients from the highest at-risk group to the lowest at-risk group. A similar idea, also based on Bayesian learning is considered by [16, 30] where the main motivation is to construct simple rules which are interpretable by human experts and can be used by healthcare providers.

Recently, [28] proposed to solve the score learning task with a cutting plane algorithm which is computationally efficient, since it iteratively solves a surrogate problem with a linear approximation of the loss function.

The state-of-the-art methods [27, 7, 16, 30, 28] are reported to be accurate, but an obvious drawback is that their output, the learned scores, apply to real-valued data (if the input data were real). Although medical data are often real indeed, a model which provides some interpretable discretization or learns *diagnostic thresholds*, is of a bigger interest for diagnostic purposes.

In our work, we cast the problem of binning as a feature selection task, where to add a bin, i.e. to add a threshold, is equivalent to add a feature into a model. It is known that feature selection and data categorization can slightly degrade performance relative to a real-valued predictor, however, in domains such as medical diagnostics, an interpretable model is preferred to a complex real-valued model which is the most accurate, if their performances are comparable. It was demonstrated [23] that it is possible to estimate sparse predictors efficiently while compromising on prediction accuracy. Binning or supervised discretization was reported to simplify the models, and not to degrade the generalizing performance. Usually, binning is performed as a pre-processing step before learning (see, e.g., [6, 19, 18, 3, 20, 11]).

Very recently, [1] introduced a new penalization called binarsity which penalizes the weights of a model learned from grouped one-hot encodings. Their approach is an attempt to learn an interpretable model using a penalty term.

3 Learning Scoring Systems

In this section, we introduce a novel algorithm called Fully Corrective Binning (FCB) which efficiently performs both binning and continuous weights learning. We also discuss how to produce a discrete scoring system, i.e. a model with discrete weights after the fully corrective binning procedure.

3.1 Preliminaries

In a supervised learning scenario, an algorithm has access to training data $\{X_i, Y_i\}_{i=1}^N \in (\mathcal{X} \times \mathcal{Y})^N$, and the goal is to find a rule to discriminate observations into two or more classes as accurate as possible. The matrix of observations X has N rows (samples), and p columns (variables), and let $X_{ij} \in [-\Omega, \Omega]$.

Definition 1. (Encodings). For any $X \in \mathcal{X}$, we define the interval encoding

$$Z_{ijlu} = \begin{cases} 1, \text{ if } X_{ij} \in]l, u] \,, \\ 0, \text{otherwise}. \end{cases} \tag{3.1}$$

Therefore, Z could be viewed as a matrix with N rows and an extended number of d columns (where $d \gg p$) indexed by the triplets j, l, u. The j-th column $X_{\cdot j}$ is thus replaced in Z by d_j columns containing only zeros and ones.

We will show later that our problem can be cast as learning a linear prediction model on Z. This linear model will be represented by a parameter vector $\theta \in \Theta \subset \mathbb{R}^d$

Without loss of generality, we consider a binary classification problem, where $\mathcal{Y} \in \{-1, 1\}$.

	var 1
X_1	-1.6
X_2	2.2

	$(-\infty, -1.6]$	$(-1.6, +\infty)$
Z_1	1	0
Z_2	0	1

Table 2: A one-dimensional dataset composed of two samples (on the left), and the interval encoding of the dataset (on the right).

The learning problem is defined as the minimization of a loss function $\ell(.,.,.)$ as follows:

$$R(\theta) = \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \ell(Z_i, Y_i, \theta). \tag{3.2}$$

The sparsity of the vector θ is defined as a number of non-zero elements in θ, and is defined as the L_0 norm:

$$\|\theta\|_0 = |\{i : \theta_i \neq 0\}|. \tag{3.3}$$

In the following, the set of integers $\{1, \dots, d\}$ is denoted by $[d]$. For a vector θ, the support of θ is defined as

$$\text{supp}(\theta) = \{i \in [d] : \theta_i \neq 0\}. \tag{3.4}$$

Following the notations of [23], if $F = \text{supp}(\theta)$, and $F' = \text{supp}(\theta')$, the set difference is $F - F'$.

3.2 Problem Statement

We define the problem of scoring systems learning as follows. We have a set of training examples $\{Z_i, Y_i\}_{i=1}^N$, where Z is the interval encoding of some matrix X, and Y is a class label. A score function is defined as $\langle \theta, Z \rangle$, where θ is a coefficient vector, and $\langle \cdot, \cdot \rangle$ is the scalar product. Given Z, and estimated weights θ, a score s_i for an observation Z_i is equal to $\langle \theta, Z_i \rangle$. A class can be predicted according to the conditional probability

$$p(y = 1 | Z) = \frac{1}{1 + \exp(-\langle \theta, Z \rangle)}. \tag{3.5}$$

Definition 2. (Scoring model). Using the original matrix X, a scoring model is defined as a real-valued vector θ such that there exists a function s_j which for every possible $X_{\cdot j}$ returns its weight (or score) θ_{jlu}:

$$s_j(X_{ij}) = \theta_{jlu} \quad \text{for } X_{ij} \in \left]l, u\right]. \tag{3.6}$$

A scoring model is in its *minimal form* if for a variable j for any two consecutive intervals $]l, r]$ and $]r, u]$

$$\theta_{jlr} \neq \theta_{jru}. \tag{3.7}$$

Note that the minimal form is unique.

Two scoring models θ and θ' are *equivalent* if

$$\text{supp}(\theta) \subseteq \text{supp}(\theta') \text{ or } \text{supp}(\theta') \subseteq \text{supp}(\theta), \tag{3.8}$$

$$\text{and } \langle \theta, Z \rangle = \langle \theta', Z \rangle, \tag{3.9}$$

For any scoring system in its minimal form, we define

$$\|\theta\|_{fused} = \sum_{j=1,\dots,p; l,r,u \in [-\Omega;\Omega]} |\theta_{jlr} - \theta_{jru}|. \tag{3.10}$$

For example, a possible scoring model θ for the data set presented in Table 2 could be

$$\{\theta_{1\ -\infty\ -1.6} = -2, \quad \theta_{1\ -1.6\ +\infty} = 2\}, \tag{3.11}$$

where the weights $\theta = [-2, 2]$ are either provided by human experts or estimated purely from data. In this example, the values of $X_{\cdot 1}$ are split into two bins $(-\infty, -1.6]$ and $(-1.6, +\infty)$.

3.3 Relation to Feature Selection

We formulate the problem of optimal binning as a feature selection task, where to *split a bin* means to add a feature into a model, and to *merge two bins*, means to *delete* a feature from this model.

The trade-off between accuracy and sparsity of feature selection methods was extensively studied by [23]. The goal is to find a reasonable balance between $R(\theta)$ and $\|\theta\|_0$, and the aim is to solve the following constrained optimization problem

$$\min_{\theta : \|\theta\|_0 \leq B} R(\theta), \tag{3.12}$$

i.e. to minimize the empirical risk with the L_0 norm bounded by a sparsity constraint B. It is easy to see that the problem (3.12) is not convex due to the constraint $\|\theta\|_0 \leq B$, and the task is NP-hard. Several approaches are considered in [23] in order to find an approximation of equation (3.12). One of the methods discussed in their paper is the fully corrective greedy selection which first fully adjusts weights of the current model so as to minimize the empirical risk, and then adds a new feature. Under *fully corrective* it is meant that the weights are optimized over all features added so far. A post-processing procedure based on replacement steps was also proposed by [23], and it aims to remove the feature with the smallest weights.

3.4 Continuous Scoring Models

In this section, we introduce the Fully Corrective Binning algorithm which efficiently performs binning and learning of the corresponding scores. Our method is parameter-free. It needs neither the L_1, nor the L_0 constraints, and relies on early stopping or on a similar greedy criterion.

The proposed algorithm at each iteration finds an optimal model over all already added features, and adds a new feature, i.e., splits one of the existing bins into two bins, if this operation minimizes the empirical risk:

$$j, l, u, r = \underset{\text{for all } j,]l,u], r \in]l,u]}{\operatorname{argmax}} \Big(\max(|(\nabla R)_{jlr}|, |(\nabla R)_{jru}|) \Big), \tag{3.13}$$

$$\theta = (\theta \cup \{\theta_{jlr}, \theta_{jru}\}) - \{\theta_{jlu}\}. \tag{3.14}$$

In a replacement step of the algorithm, the least important feature

$$j, l, u, q = \underset{\text{for all } j,]l,q],]q,u], q \in]l,u]}{\operatorname{argmin}} \Big(|\theta_{jlq} - \theta_{jqu}| \Big), \tag{3.15}$$

$$\theta = (\theta \cup \{\theta_{jlu}\}) - \{\theta_{jlq}, \theta_{jqu}\}. \tag{3.16}$$

is removed from the model if this operation does not degrade the performance. In other words, one of the bins is merged with its neighbour. Now let us consider the theoretical guarantees of the newly introduced procedure which is given as Algorithm 1. The parameter K controls the number of bins, and can vary for different applications.

The proof of convergence of the introduced algorithm relies heavily on the following lemma [23].

Lemma 3. (Progress of one greedy iteration. Lemma A.5 of [23]).

Let θ and $\hat{\theta}$ be two scoring models. Let F and $\hat{F}$ be the corresponding supports of vectors θ and $\hat{\theta}$, with $\hat{F} - F \neq 0$, and such that

$$\theta = \underset{\theta : supp(\theta) = F}{\operatorname{argmin}} R(\theta). \tag{3.17}$$

Assume the loss function used in R is β-smooth. Then, for

$$]\hat{l}, \hat{u}] = \underset{]l,u]}{\operatorname{argmax}} \left| (\nabla R(\theta))_{]l,u]} \right|, \tag{3.18}$$

we have:

$$R(\theta) - \min_{\alpha} R(\theta + \alpha \mathbf{e}^{]\hat{l},\hat{u}]}) \geq \frac{\Big(R(\theta) - R(\hat{\theta})\Big)^2}{2\beta \left(\sum_{b \in \hat{F} - F} |\hat{\theta}_b| \right)^2},$$

where $\mathbf{e}^{]\hat{l},\hat{u}]}$ refers to the unitary vector where only component $\left]\hat{l}, \hat{u}\right]$ is non-zero.

Lemma 4. Let $\hat{\theta}$ and θ be two minimal scoring models such that supp$(\hat{\theta}) \neq$ supp(θ). Let $\bar{\theta}$ be a scoring model containing all possible splits $]l, r]$ and $]r, u] \in]l, u]$ for all variables $j \in \{1, \ldots, p\}$. Then $\bar{\theta}$ is equivalent to θ, and θ' is also equivalent to θ. It can be verified that

$$\|\theta\|_1 \leq \|\hat{\theta}\|_1 + \|\hat{\theta}\|_{fused}. \tag{3.19}$$

Proposition 5. *Let us consider an arbitrary scoring model in minimal form $\hat{\theta}$. After T iterations of the Fully Corrective Binning, we will get a scoring model θ such that*

$$R(\theta) \leq R(\hat{\theta}) + \frac{2\beta(\|\hat{\theta}\|_1 + \|\hat{\theta}\|_{\textit{fused}})^2}{T}. \tag{3.20}$$

Proof. Let F and $\hat{F}$ be the supports of θ and $\hat{\theta}$ respectively, $\hat{\theta} \neq \theta$ and $F \neq \hat{F}$. Let us build a scoring model $\bar{\theta}$ that includes all possible splits of all existing bins

$$]l, r],]r, u] \in]l, u], l < r < u. \tag{3.21}$$

The Fully Corrective Binning considers all these candidate splits in the binning phase of the learning procedure. Note that θ is equivalent to $\bar{\theta}$, since supp$(\theta) \subseteq$ supp$(\bar{\theta})$, and also $\hat{\theta}$ is equivalent to $\bar{\theta}$, since supp$(\hat{\theta}) \subseteq$ supp$(\bar{\theta})$.

If we apply Lemma 4, we see that all conditions required by Lemma 3 are met:

$$R(\bar{\theta}) = R(\theta) \leq R(\hat{\theta}), \tag{3.22}$$

i.e. one greedy iteration of the algorithm leads to a substantial improvement.

To evaluate the model produced after T iterations, let us now apply the same trick recursively. Finding the maximal value among the gradient coordinates over all possible splits, what is exactly done by our algorithm, stems down to computing $\max_{\text{for all }]l,u]} |(\nabla R)_{]l,u]}|$, since the gradient values over intervals in supp(θ) are equal to zero given that θ is optimal. Assume that $]l, r]$ is an interval on which the absolute value of the gradient is maximal, and, hence, we split $]l, u]$ into $]l, r]$ and $]r, u]$. Lemma 3 gives us a bound on the progress brought by changing simultaneously the weights corresponding to $]l, r]$ and $]r, u]$.

Let θ be the updated model after the split. Applying the result recursively T times, we get the bound.

□

3.5 Discrete Scoring Systems

Although real-valued scoring systems are of a big interest, discrete scores are even easier to be interpreted and to be used by human experts. In this section we discuss two methods how to construct scoring systems where the weights are integers.

A natural idea to learn a system where the weights are discrete is to apply the interval encoding, eq. (3.1), and to minimize the 0-1 loss penalized by the L_0 norm:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{Y_i \theta^T X_i \leq 0\}} + \Phi(\theta), \tag{3.23}$$

Algorithm 1. **The Fully Corrective Binning**

```
Input: Training data $\{X_i, Y_i\}_{i=1}^N$,    $X : N \times p$
Output: Scoring model $\theta$
Construct matrix $Z$ from $X$ according to eq. (3.1)                    // Initialize the bins
for all $j \in \{1, \ldots, p\}$
    $\theta_{j-\infty+\infty} = 0$                                       // Initialize the weights
end for
$\theta = \arg \min_{\text{supp}(\theta)} R(\theta)$                    // Update the parameters
for $t = 1, \ldots, T$
    $j, l, u, r = \text{argmax}_{\text{for all } j, ]l,u], r \in ]l,u]} \Big( \max(|(\nabla R)_{jlr}|, |(\nabla R)_{jru}|) \Big)$,    // Split (add) a variable and update the binning
    $\theta = (\theta \cup \{\theta_{jlr}, \theta_{jru}\}) - \{\theta_{jlu}\}$.
    $\theta = \arg \min_{\text{supp}(\theta)} R(\theta)$                // Update the parameters, update $Z$
    if $t > K$
        $j, l, u, q = \text{argmin}_{\text{for all } j, ]l,q], ]q,u], q \in ]l,u]} \Big( |\theta_{jlq} - \theta_{jqu}| \Big)$,    // Merge (delete) a variable and update the binning
        $\theta = (\theta \cup \{\theta_{jlu}\}) - \{\theta_{jlq}, \theta_{jqu}\}$.
        $\theta = \arg \min_{\text{supp}(\theta)} R(\theta)$            // Update the parameters again, update $Z$
    end if
end for
```

where

$$\Phi(\theta) = C_1 \sum_{j=1}^{p} \sum_{\text{for all }]l,u]} |\theta_{jlu}|_1 + \tag{3.24}$$

$$C_0 \sum_{j=1}^{p} \sum_{\text{for all }]l,u]} \mathbb{1}_{\{\theta jlu\}} + \tag{3.25}$$

$$C_{1f} \sum_{j=2}^{p} \sum_{\text{for all }]l,r],]r,u]} |\theta_{jru} - \theta_{jlr}|_1 + \tag{3.26}$$

$$C_{0f} \sum_{j=2}^{p} \sum_{\text{for all }]l,r],]r,u]} \mathbb{1}_{\{\theta_{jru} - \theta_{jlr} \neq 0\}}, \tag{3.27}$$

with C_1, C_0, C_{1f}, and C_{0f} chosen by cross validation. Such an approach is a generalization of the SLIM scoring system [27], and we provide its integer programming formulation in Appendix. The task is presented and solved as an integer programming problem, and we use the Matlab implementation[1] provided by the SLIM authors. The training procedure relies on the IBM ILOG CPLEX Optimization Studio[2] which efficiently performs the constrained optimization. In particular, integrity constraints are added to the optimisation problem to obtain integer solutions.

Another idea to construct a model with integer weights, is to discretize the real-valued weights after the fully corrective binning procedure, e.g., a randomized rounding method (see [12, 4] for details) can be applied to the continuous scores after training.

4 Experiments

In this section, we share the results of our experiments on simulated data, on two standard benchmarks, and a real biomedical challenge. We compare the proposed approach both with continuous and discrete weights to the state-of-the-art SLIM scoring system [27], and to the 0-1 loss penalized by the fused L_0. We also test the performance of models where we perform data discretization as the pre-processing step using top-down discretization methods such as CAIM [15], CACC [26], Ameva [13], and the Minimum Description Length Principle method [8]. FCB on the figures below stands for the proposed Fully Corrective Binning approach. We show the performance of the continuous and discrete versions of the algorithm.

4.1 Synthetic Data

We first illustrate how the proposed approach performs binning and weight estimation on a simulated data set. The binary artificial task is constructed as follows. We fix the number of variables, and the number of optimal bins per each variable. We randomly draw the optimal weights associated with each bin of each variable according to the Gaussian normal distribution. The class label is equal to 1 if the sum of corresponding weights over all variables is bigger or equal to 0, and otherwise the class label is set

[1] https://github.com/ustunb/slim-matlab

[2] http://www-03.ibm.com/software

to 0.

We test two simple cases. The first case with one variable and two bins, and a slightly more complex problem with two variables and two bins for each variable. We generate 10 000 observations and labels, and we show the estimated cuts defining bins, and the corresponding weights. We minimize the hinge loss to perform training. Note that the model we fit is not the model from which the data were generated. Hence, the model we learn is always ill-specified.

Figure 1 illustrates the cuts and the weights. The red lines are the optimal cuts defined by the true unknown model. On the left is the simplest case with 1 variable with two bins, and our algorithm finds the optimal cut at the first iteration, and the learned weights let to predict the class with accuracy which is very close to 100%. In the case with 2 variables, our algorithm finds a reasonable model after 2 iterations only, and after 10 iterations it finds the true cuts.

Next, we illustrate on Figure 2 the performance of the Fully Corrective Binning on three simulated data sets. The tasks are two-dimensional binary classification problems, with blue points belonging to class 1, and red points belonging to class 2. The first row of Figure 2 shows the distribution of points in the data sets. In the second row we plot the boundaries found by the linear hinge loss classifier. The third row displays the boundaries obtained with the Fully Corrective Binning algorithm. The last row shows the performance in terms of 10-fold cross validation test error of the hinge loss and of the FCB. Our results are similar to the findings of [1]: the discretized classifier is promising on non-linear data sets, shown in the columns 1 and 2 of Figure 2. However, in some cases such as the case of a linear classification problem shown in the column 3, it seems that the binarization of features can lead to an important overfitting. If we allow the algorithm to produce too many bins, i.e. if K in Algorithm 1 is chosen too big, a model can be overfitted.

4.2 Standard Biomedical Benchmarks

In this section, we share our results on two standard benchmarks, Glaucoma and Breast cancer prediction tasks. The Breast cancer data are downloadable from the UCI Machine Learning repository[3] [17]. In the Breast Cancer Wisconsin (Prognostic) data set, we dispose of about 30 parameters describing characteristics of the cell nuclei present in the medical images for 198 patients [25]. All parameters are continuous. Glaucoma diagnosis set includes data from laser scanning images taken from the eye background for 170 patients and 66 attributes, providing information on the morphology of the optic nerve head, the visual field, the intra occular pressure and a membership variable. The data is part of the “ipred” R package [22].

Variable	Thresholds	Score
Age	<38	0
	38 – 52	2
	52 – 70	4
Glycated hemoglobin	<7.0	0
	7.0 – 7.4	2
	7.4 – 20	4
Insuline	No	0
	Yes	7
Other drugs	No	0
	Yes	2

Classify as `Remission` if sum of scores < 8
Classify as `Non-remission` if sum of scores ≥ 8

Table 4: Diabetes remission scoring model learnt by for the Fully Corrective Binning algorithm.

To evaluate our approach, we perform 10-fold cross validation and boxplot the testing error. The error rates for all tested methods are shown on Figure 3. On the left, we show the results for the Breast cancer data, in the center, for the Glaucoma data, and on the right, for the original Diabetes remission task described in the following section. It is easy to see that the discrete fully corrective binning outperforms the state-of-the-art. Note that it does not make any sense to test the FCB on the data sets chosen by [16, 30, 28], since these benchmarks are discrete.

4.3 Real Biomedical Challenge

Recently physicians [24] proposed a discrete clinical score called DiaRem score to predict whether a gastric bypass surgery could lead to a diabetes remission. It is based on four clinical variables only, namely, age, glycated hemoglobin, and it takes into account whether insuline is taken, and whether other anti-diabetic drugs are prescribed to a patient. Each clinical continuous variable (age and glycated hemoglobin) was discretized in some meaningful for physicians and clinicians way. It was reported that to obtain the score for each category, the odd ratios were computed, and some heuristic method was applied to get the integer weights. The original DiaRem clinical thresholds and the scores for the bins are shown in Table 1.

Only one arithmetic operation is needed to predict an outcome for a new patient. If the final score which is the sum of weights associated with each clinical category is < 7, then this patient will benefit from the operation with probability 80%. If the sum of corresponding values for age, glycated hemoglobin, insuline, and other drugs for a particular patient is more than 7, then the remission is not likely. The separator value which is equal to 7 was reported to be the optimal one by [5].

[3] http://archive.ics.uci.edu/ml/

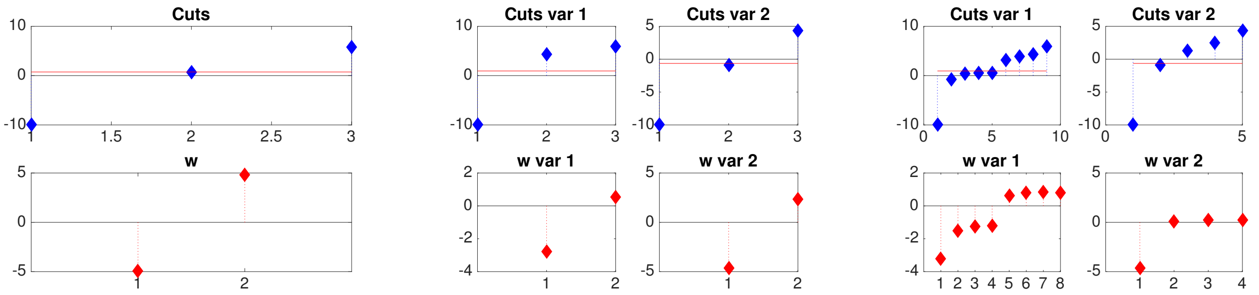

Figure 1: Simulated Data. On the left: 1 variable, 2 bins, and 1 iteration. In the center: 2 variables, 2 bins, and 2 iterations. On the right: 2 variables, 2 bins, and 10 iterations. Above: the cuts, below: the weights.

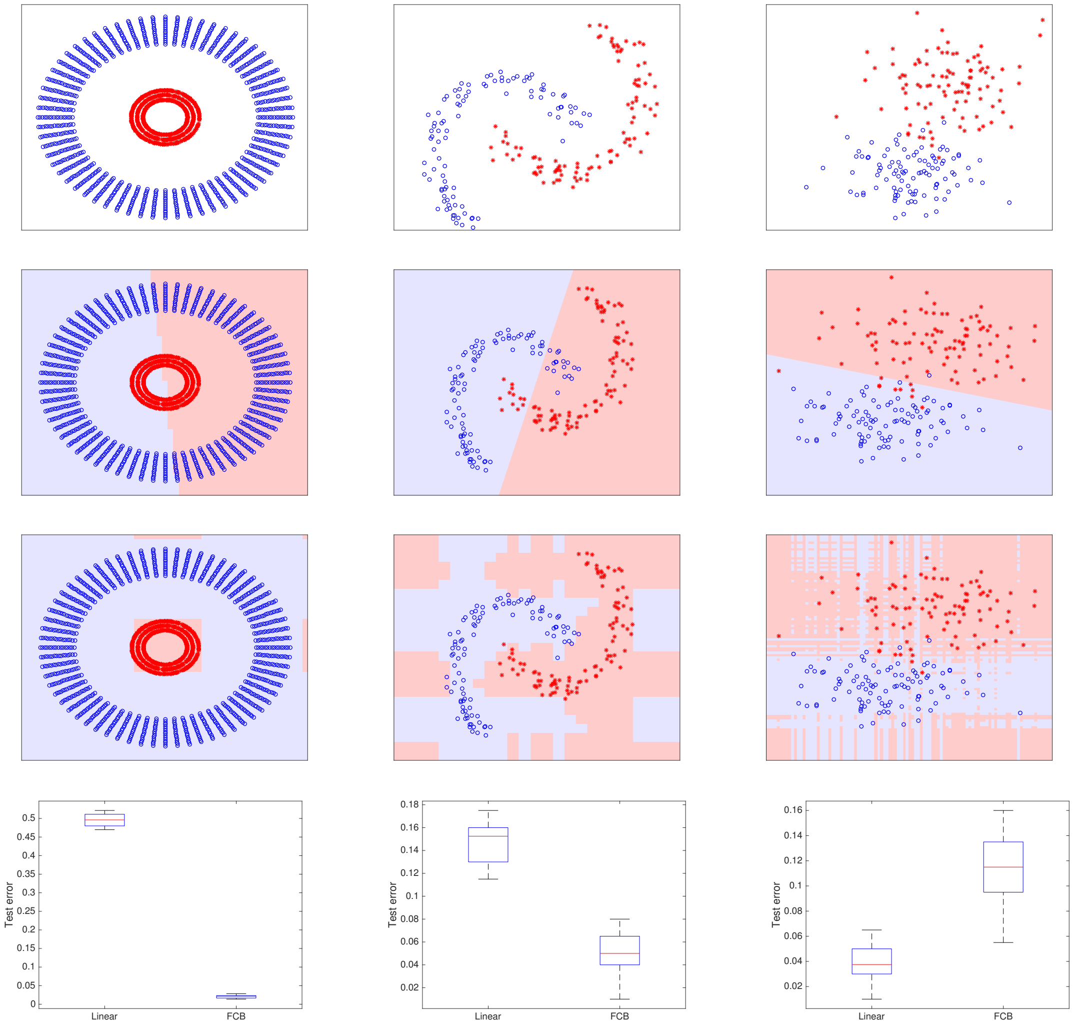

Figure 2: Comparison of the linear hinge loss and the FCB separators on three simulated toy data sets.

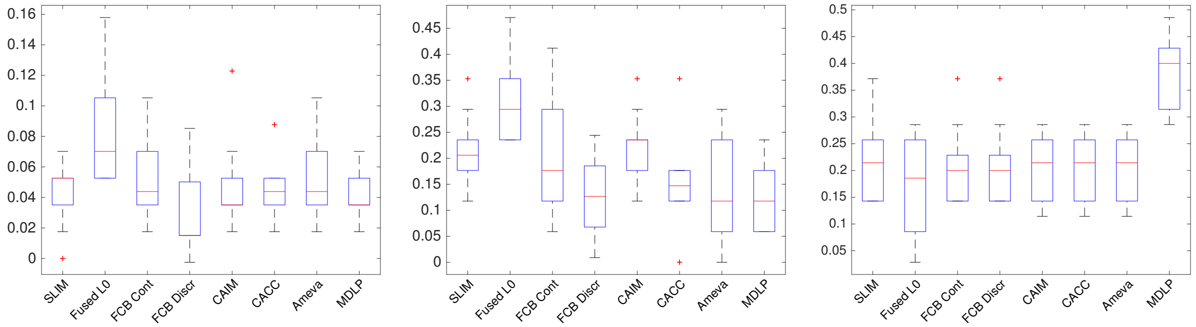

Figure 3: Test error. On the left: Breast cancer data; in the center: Glaucoma data set; on the right: Diabetes remission prediction.

We tried to learn the DiaRem score automatically. The data set of type 2 diabetic subjects is produced and managed by the Department of Nutrition, Center of Reference for Medical and Surgical Care of Obesity, at the Institute of Cardiometabolism and Nutrition (ICAN), Pitié-Salpêtrière Hospital (Paris, France).

We applied the fully corrective binning both continuous and discrete, the SLIM scoring system, and its version penalized by the L_0 norm to learn the diabetes remission scoring system in a completely automated way. The generalizing error is shown on Figure 3 on the right. With our algorithm, we got an alternative score which can be compared to the DiaRem (Table 1), presented as Table 4. Both scoring systems have similar accuracy (around 82%).

5 Conclusions

Our goal was to develop a principled approach to learn scores from continuous data where we simultaneously estimate interpretable thresholds to bin data, and the corresponding scores. The proper theoretical support for the proposed fully corrective binning algorithm is provided in Section 3. We have visualized the intuition how the proposed method learns the separator between two classes. We have demonstrated by our experiments on standard biomedical data from the UCI machine learning repository that the novel approach is promising and competitive compared to several modern methods. Namely, the algorithm outperforms or achieves the state-of-the-art accuracy. Also note, that the state-of-the-art method SLIM relies on linear programming optimization with integrity constraints what is computationally expensive, and can be intractable for huge data sets.

Another important result is discussed in Section 4.3 where we describe a real original medical challenge. We illustrated by a problem of type 2 diabetes remission the potential of the proposed algorithm to efficiently learn scores purely from data, what traditionally costs many hours of work of human experts. Although quite promising and efficient, the current version of the novel algorithm is not fully optimized. Currently we are investigating applications of the fully corrective binning to huge data sets such as metagenomic data, and consider the scalability issues. Another avenue of research is to adopt deep learning architectures for learning scoring systems.

Acknowledgements

The work was supported by the French National Research Agency (ANR JCJC *DiagnoLearn*).

Appendix

Here we provide an Integer Programming formulation to train the 0-1 loss penalized by the fused L_0 norm:

$$\min_{\lambda,\psi,\Phi,\alpha,\beta} \frac{1}{N}\sum_{i=1}^{N}\psi_i + \sum_{j=1}^{p}\Phi_j \tag{5.1}$$

$$\text{such that for } i = 1,\dots,N, j = 1,\dots,p \tag{5.2}$$

$$M_i\psi_i \geq \gamma - \sum_{j=1}^{p} y_i\lambda_j x_{ij} \tag{5.3}$$

$$\Phi_j = C_0\alpha_j + C_1\beta_j + C_{0f}\alpha_{jf} + C_{1f}\beta_{jf} \tag{5.4}$$

$$-\Lambda_j\alpha_j \leq \lambda_j \leq \Lambda_j\alpha_j, -\beta_j \leq \lambda_j \leq \beta_j \tag{5.5}$$

$$-\Lambda_{jf}\alpha_{jf} \leq \lambda_j - \lambda_{j-1} \leq \Lambda_{jf}\alpha_{jf} \tag{5.6}$$

$$-\beta_{jf} \leq \lambda_j - \lambda_{j-1} \leq \beta_{jf} \tag{5.7}$$

$$\psi_i \in \{0,1\}\Phi_j \in \mathbb{R}_+\lambda_j \in \mathcal{L}_j \tag{5.8}$$

$$\alpha_j \in \{0,1\} \quad \beta_j \in \{0,1\} \tag{5.9}$$

$$\alpha_{jf} \in \{0,1\} \quad \beta_{jf} \in \{0,1\} \tag{5.10}$$

References

[1] M. Z. Alaya, S. Bussy, S. Gaiffas, and A. Guilloux. Binarsity: a penalization for one-hot encoded features. arXiv:1703.08619, 2017.

[2] E. Antman et al. The TIMI risk score for unstable angina/non–st elevation mi. *The Journal of the American Medical Association*, 2000.

[3] R. Butterworth, D. Simovici, G. Santos, and L. Ohno-Machado. A greedy algorithm for supervised discretization. *Journal of Biomedical Informatics*, 2004.

[4] Y. Chevaleyre, F. Koriche, and J.-D. Zucker. Rounding methods for discrete linear classification. In *ICML*, 2013.

[5] A. Cotillard, C. Poitou, G. Duchâteau-Nguyen, J. Aron-Wisnewsky, J.-L. Bouillot, T. Schindler, and K. Clément. Type 2 diabetes remission after gastric bypass: What is the best prediction tool for clinicians? *Obesity Surgery*, 25(7):1128–1132, 2015.

[6] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *ICML*, 1995.

[7] S. Ertekin and C. Rudin. A Bayesian approach to learning scoring systems. *Big Data*, 3(4), 2015.

[8] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. *Artificial intelligence*, 13:1022–1027, 1993.

[9] B. F. Gage et al. Validation of clinical classification schemes for predicting stroke. *The Journal of the American Medical Association*, 2001.

[10] J.-R. Le Gall, S. Lemeshow, and F. Saulnier. A new simplified acute physiology score (SAPS II) based on a european/north american multicenter study. *The Journal of the American Medical Association*, 1993.

[11] S. Garcia, J. Luengo, J. A. Saez, V. Lopez, and F. Herrera. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750, 2013.

[12] D. Golovin, D. Sculley, H. B. McMahan, and M. Young. Large-scale learning with less ram via randomization. In *ICML*, 2013.

[13] L. Gonzalez-Abril, F.J. Cuberos, F. Velasco, and J.A. Ortega. Ameva: An autonomous discretization algorithm. *Expert Systems and Applications*, 36(3):5327–5332, 2009.

[14] W. A. Knaus, J. E Zimmerman, D. P. Wagner, E. A. Draper, and D. E. Lawrence. Apache-acute physiology and chronic health evaluation: a physiologically based classification system. *Critical Care Medicine*, 1981.

[15] L.A. Kurgan and K.J. Cios. Caim discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):145–153, 2004.

[16] B. Letham, C. Rudin, T. McCormick, and D. Madigan. Building interpretable classifiers with rules using Bayesian analysis. *Annals of applied statistics*, 2015.

[17] M. Lichman. UCI machine learning repository, 2013.

[18] H. Liu, F. Hussain, C. Lim Tan, and M. Dash. Discretization: an enabling technique. *Data Mining and Knowledge Discovery*, 6:393–423, 2002.

[19] J. L. Lustgarten, V. Gopalakrishnan, H. Grover, and S. Visweswaran. Improving classification performance with discretization on biomedical datasets. In *AMIA Symposium Proceedings*, 2008.

[20] D. M. Maslove, T. Podchiyska, and H. J. Lowe. Discretization of continuous features in clinical datasets. *J Am Med Inform Assoc*, 20:544–553, 2013.

[21] R. Moreno et al. Development of a prognostic model for hospital mortality at icu admission. *Intensive Care Medicine*, 2005.

[22] A. Peters, T. Hothorn, and B. Lausen. ipred: Improved predictors. *R News*, 2(2), 2002.

[23] S. Shalev-Shwartz, T. Zhang, and N. Srebro. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20(6):2807–2832, 2010.

[24] C.D. Still et al. A probability score for preoperative prediction of type 2 diabetes remission following rygb surgery. *Lancet Diabetes Endocrinol.*, 2(1):38–45, 2014.

[25] W. N. Street, O. L. Mangasarian, and W.H. Wolberg. An inductive learning approach to prognostic prediction. In *ICML*, 1995.

[26] C.-J. Tsai, C.-I. Lee, and W.-P. Yang. A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences*, 178(3):714–731, 2008.

[27] B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 2015.

[28] Berk Ustun and Cynthia Rudin. Learning optimized risk scores from large-scale datasets. In *KDD*, 2017.

[29] F. Wang and C. Rudin. Falling rule lists. In *AISTATS*, 2015.

[30] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable Bayesian rule lists. In *ICML*, 2017.