Expert Systems with Applications 42 (2015) 1789–1796

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Classification Restricted Boltzmann Machine for comprehensible credit scoring model

Jakub M. Tomczak *, Maciej Zięba

Institute of Computer Science, Wroclaw University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

ARTICLE INFO

Article history:
Available online 18 October 2014

Keywords:
Credit scoring
Comprehensible model
Restricted Boltzmann Machine
Imbalanced data

ABSTRACT

Credit scoring is the assessment of the risk associated with a consumer (an organization or an individual) that apply for the credit. Therefore, the problem of credit scoring can be stated as a discrimination between those applicants whom the lender is confident will repay credit and those applicants who are considered by the lender as insufficiently reliable. In this work we propose a novel method for constructing comprehensible scoring model by applying Classification Restricted Boltzmann Machines (ClassRBM). In the first step we train the ClassRBM as a standalone classifier that has ability to predict credit status but does not contain interpretable structure. In order to obtain comprehensible model, first we evaluate the relevancy of each of binary features using ClassRBM and further we use these values to create the scoring table (scorecard). Additionally, we deal with the imbalanced data issue by proposing a procedure for determining the cutting point using the geometric mean of specificity and sensitivity. We evaluate our approach by comparing its performance with the results gained by other methods using four datasets from the credit scoring domain.

1. Introduction

The problem of building a decision model for identification of consumers with unsecured repayment status can be seen as an issue of training a dichotomous classifier, where the positive class (usually less numerous) represents "bad" applicants and the negative class stays behind "good" cases. Each example in training data refers to a customer described by a vector of attributes that codes the most important information about her or him in the considered credit approval context, and the corresponding class label represents the real repayment status. The main goal of the training procedure is to construct the model that will be able to correctly classify as many new clients as possible.

Unfortunately, the specificity of the problem enforces couple of limitations related to the issue of constructing decision models from data. First, according to the regulation of banking supervision, financial institutions in some countries are obliged to present a comprehensible justification in the case the credit application is denied. Therefore, the process of decision making performed by the data-based scoring model ought to be interpretable. As a consequence, only comprehensible models like scoring tables (scorecards), decision trees and rules are suitable to be used to deal with that issue. Second, the training set used to construct decision model is usually influenced by imbalanced data phenomenon, because the data is dominated by applicants with positive credit approval. Consequently, the typical learner constructed on such data is biased toward majority class, what practically means that it has tendency to assign positive repayment status even to very risky consumers. As a result, the problem of uneven class distribution should be taken into account in the process of constructing credit scoring model.

The issue of constructing credit scoring models directly from data has been successfully studied since Durand proposed to use discriminant function to separate "good" and "bad" consumers in 1941 (Crook, Edelman, & Thomas, 2007). Current models used for the credit risk assessment utilize the machine learning techniques to increase the accuracy of prediction, to deal with the imbalanced data phenomenon, or to construct comprehensible learners. Various classification methods are considered to be used to predict credit repayment such as neural networks (West, 2000), Gaussian Processes (Huang, 2011), Support Vector Machines (SVMs) (Bellotti & Crook, 2009; Huang, Chen, & Wang, 2007), or ensemble classifiers (Nanni & Lumini, 2009). Many authors recognize the need of constructing the comprehensible models directly from data by applying rules and trees inducers (Crook et al., 2007), or indirectly, by extracting interpretable models from strong learners, so-called "black-box" classifiers, such as SVMs (Martens, Baesens, Van Gestel, & Vanthienen, 2007) or ensemble classifiers (De Bock & Van den Poel, 2012). The issue of imbalanced data in the context

* Corresponding author. Tel.: +48 71 320 44 53.
E-mail addresses: jakub.tomczak@pwr.edu.pl (J.M. Tomczak), maciej.zieba@pwr.edu.pl (M. Zięba).

http://dx.doi.org/10.1016/j.eswa.2014.10.016
0957-4174/

of constructing credit scoring models was also considered in the literature, i.e., adaptive, cost-sensitive version of SVM (Yang, 2007), balanced neural networks (Huang, Hung, & Jiau, 2006), or ensemble classifier with switching class labels (Zieba & Świątek, 2012).

In this work we propose a novel machine learning technique which takes advantage of Classification Restricted Boltzmann Machine (ClassRBM) to construct the credit scoring table. We use ClassRBM as a universal approximator over the binary random variables which is further applied to determine weights (scoring points) in the scoring table. Moreover, the scoring tables are the simplest models to interpret and can be easily implemented in any bank system. Additionally, our approach deals with the imbalanced data by selecting the cutting point with the highest geometric mean of specificity and sensitivity. Unlike standard methods, our approach combines issues which are typically considered separately: (i) it makes use of the strong classifier (ClassRBM), (ii) it deals with the uneven class distribution problem, and (iii) it constructs highly comprehensible and easy-to-implement scoring model.

This work is organized as follows. In Section 2 Classification Restricted Boltzmann Machine is introduced, the procedure of constructing credit scoring table using this model is presented and the procedure for determining the cutting point for imbalanced data is outlined. In Section 3 the quality of the presented approach is examined by comparing its performance to state-of-the-art methods using the datasets from the credit scoring domain. This paper is summarized by conclusions in Section 4.

2. Methodology

In this section we present novel method to construct the scoring table basing on the trained Classification Restricted Boltzmann Machine (ClassRBM). In the first part of this section we introduce the ClassRBM as a standalone classifier which allows to use the probabilistic framework to evaluate the relevancy of the binary attributes. Further, we present a method for constructing comprehensible credit scoring model using the class-dependent importance of each of the attributes indicated by ClassRBM. Finally, we describe the procedure to determine the cutting point by maximizing the geometric mean of the specificity and the sensitivity on training data.

2.1. Classification Restricted Boltzmann Machine

2.1.1. Model definition

ClassRBM (Larochelle & Bengio, 2008; Larochelle, Mandel, Pascanu, & Bengio, 2012) is a three-layer undirected graphical model where the first layer consists of visible input variables $\mathbf{x} \in \{0,1\}^D$, the second layer consists of hidden variables (units) $\mathbf{h} \in \{0,1\}^M$, and the third layer represents observable output variable $y \in \{1,2,\ldots,K\}$. We use the 1-to-K coding scheme which results in representing output as a binary vector of length K denoted by $\mathbf{y}$, such that if the output (or class) is k, then all elements are zero except element y_k which takes the value 1. We allow only the inter-layer connections, i.e., there are no connections within a layer.

With each state ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{h}$) we associate the energy given by the following equation:

$$E(\mathbf{x},\mathbf{y},\mathbf{h}|\boldsymbol{\theta}) = -\mathbf{b}^\top\mathbf{x} - \mathbf{c}^\top\mathbf{h} - \mathbf{d}^\top\mathbf{y} - \mathbf{x}^\top\mathbf{W}^1\mathbf{h} - \mathbf{h}^\top\mathbf{W}^2\mathbf{y} \tag{1}$$

with parameters $\boldsymbol{\theta} = \{\mathbf{b},\mathbf{c},\mathbf{d},\mathbf{W}^1,\mathbf{W}^2\}$. A ClassRBM with M hidden units is a parametric model of the joint distribution of visible and hidden variables, that takes the following form:

$$p(\mathbf{x},\mathbf{y},\mathbf{h}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\exp\{-E(\mathbf{x},\mathbf{y},\mathbf{h}|\boldsymbol{\theta})\} \tag{2}$$

where

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x},\mathbf{y},\mathbf{h}} \exp\{-E(\mathbf{x},\mathbf{y},\mathbf{h}|\boldsymbol{\theta})\} \tag{3}$$

is a partition function.

The advantage of the ClassRBM is that crucial conditional probabilities which are further used in the inference can be calculated analytically (Larochelle & Bengio, 2008; Larochelle et al., 2012)[1]:

$$p(\mathbf{x}|\mathbf{h}) = \prod_i p(x_i|\mathbf{h}) \tag{4}$$

$$p(x_i = 1|\mathbf{h}) = \mathrm{sigm}(b_i + \mathbf{W}^1_{i\cdot}\mathbf{h}) \tag{5}$$

$$p(y_k = 1|\mathbf{h}) = \frac{\exp\{d_k + (\mathbf{W}^2_{\cdot k})^\top\mathbf{h}\}}{\sum_l \exp\{d_l + (\mathbf{W}^2_{\cdot l})^\top\mathbf{h}\}} \tag{6}$$

$$p(\mathbf{h}|y_k = 1,\mathbf{x}) = \prod_j p(h_j|y_k = 1,\mathbf{x}) \tag{7}$$

$$p(h_j = 1|y_k = 1,\mathbf{x}) = \mathrm{sigm}(c_j + (\mathbf{W}^1_{\cdot j})^\top\mathbf{x} + W^2_{jk}) \tag{8}$$

where $\mathrm{sigm}(\cdot)$ is the logistic sigmoid function, $\mathbf{W}^\ell_{i\cdot}$ is ith row of weights matrix $\mathbf{W}^\ell$, $\mathbf{W}^\ell_{\cdot j}$ is jth column of weights matrix $\mathbf{W}^\ell$, W^ℓ_{ij} is the element of weights matrix $\mathbf{W}^\ell$.

An important advantage of the ClassRBM is that for enough number of hidden units this model can represent any distribution over binary vectors and its likelihood can be improved by adding new hidden units, unless the generated distribution already equals the training distribution (Le Roux & Bengio, 2008; Martens, Chattopadhya, Pitassi, & Zemel, 2013). This is a significant fact because we have (at least theoretical) assurance that the ClassRBM is a universal approximator for distributions over binary inputs.

For the considered problem of credit repayment the vector of binary inputs $\mathbf{x}$ represents the characteristics which describe the applicant and the output vector $\mathbf{y}$ stays behind the credit decision variant. Therefore, the vector of the hidden units allows to approximate the distribution over the entire space representing credit applicants.

2.1.2. Prediction

For given parameters $\boldsymbol{\theta}$ it is possible to compute the distribution $p(y|\mathbf{x},\boldsymbol{\theta})$ which can be further used to choose the most probable class label. This conditional distribution takes the following form (Larochelle & Bengio, 2008; Larochelle et al., 2012):

$$p(y_k = 1|\mathbf{x},\boldsymbol{\theta}) = \frac{\exp\{d_k\}\prod_j\left(1 + \exp\{c_j + (\mathbf{W}^1_{\cdot j})^\top\mathbf{x} + W^2_{jk}\}\right)}{\sum_l \exp\{d_l\}\prod_j\left(1 + \exp\{c_j + (\mathbf{W}^1_{\cdot j})^\top\mathbf{x} + W^2_{jl}\}\right)}. \tag{9}$$

Notice that the ClassRBM can be used as a standalone classifier to predict the credit repayment status. However, because it is hard-to-interpret "black box" model, it is rather unlikely to be used as a credit scoring model.

2.1.3. Learning

The key issue in the ClassRBM is the choice of a learning procedure. We assume given N data, $\mathcal{D} = \{\mathbf{x}_n, y_n\}$, and the likelihood function as the objective. However, in order to train ClassRBM we may consider two approaches. The first one, called *generative approach*, aims at maximizing the likelihood function for joint distribution $p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta})$. The second one, which we refer to as *discriminative approach*, considers the likelihood function for conditional distribution $p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$. The generative approach in the context of the ClassRBM is troublesome because exact gradient of the likelihood function for joint distribution cannot be calculated analytically, and only an approximation can be applied, e.g.,

[1] Further in the paper, sometimes we omit explicit conditioning on parameters $\boldsymbol{\theta}$.

Contrastive Divergence (Hinton, 2002). On the other hand, the latter approach allows to compute exact gradient (Larochelle et al., 2012). Additionally, we are interested in obtaining high predictive accuracy, and thus, it is more advantageous to learn ClassRBM in a discriminative manner. Therefore, to train ClassRBM we consider minimization of the negative log-likelihood in the following form:

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{n=1}^{N} \log\ p(\mathbf{y}_n|\mathbf{x}_n, \boldsymbol{\theta}). \tag{10}$$

As stated before, since the distribution $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ can be calculated exactly, the gradient of (10) can be computed exactly too (for details see (Larochelle & Bengio, 2008; Larochelle et al., 2012)). In this work we will apply the discriminative approach to determine parameters $\boldsymbol{\theta}$.

To prevent the model from overfitting, additional regularization term can be added to the learning objective:

$$\mathcal{L}_{\Omega}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) + \lambda\Omega(\boldsymbol{\theta}), \tag{11}$$

where $\lambda > 0$ is the regularization coefficient, and $\Omega(\boldsymbol{\theta})$ is the regularization term. In the following, we use the *weight decay* regularization, i.e., $\Omega(\boldsymbol{\theta}) = \|\mathbf{W}\|_F$, where $\|\cdot\|_F$ is the Frobenius norm.

2.1.4. Calculating inputs relevancy

The ClassRBM can be also used to determine relevancy of inputs by calculating conditional probabilities $p(x_i = 1|\mathbf{x}_{\setminus i}, y)$. In the credit scoring context this conditional probability expresses the probability of occurring ith applicant's characteristic for given other inputs (characteristics) and the class label. For example, if $y = 1$ denotes the credit was not repaid, we can quantitatively determine which inputs are important (relevant) during prediction of unreliable credit applicants. Further, we make an assumption that for ith input all other inputs are inactive, i.e, $\mathbf{x}_{\setminus i} = \mathbf{0}$. Hence, we get the following expression for the conditional probability:

$$\begin{aligned} p(x_i = 1|\mathbf{x}_{\setminus i} = \mathbf{0}, y_k = 1, \boldsymbol{\theta}) &= \frac{e^{b_i}\prod_j\left(1 + e^{c_j + (\mathbf{W}^1_j)^\top \mathbf{x} + W^2_{jk}}\right)}{\sum_{x_i=\{0,1\}} e^{b_i}\prod_j\left(1 + e^{c_j + (\mathbf{W}^1_j)^\top \mathbf{x} + W^2_{jk}}\right)} \\ &= \frac{e^{b_i}\prod_j\left(1 + e^{c_j + W^1_{ij} + W^2_{jk}}\right)}{e^{b_i}\prod_j\left(1 + e^{c_j + W^1_{ij} + W^2_{jk}}\right) + \prod_j\left(1 + e^{c_j + W^2_{jk}}\right)} \end{aligned} \tag{12}$$

2.2. Comprehensible credit scoring model

Since the mid-twentieth century, the volume of credit consumers drastically increased by several magnitudes (Crook et al., 2007). These large increases in lending and credit applications enforce development of automatic tools for consumer credit risk assessment including statistical and machine learning methods which constitute a general class of methods known as *credit scoring*.

In general, credit scoring is the assessment of the risk associated with a consumer (an organization or an individual) that apply for the credit (Crook et al., 2007; Hand & Henley, 1997). Therefore, the problem of credit scoring can be stated as a discrimination between those applicants whom the lender is confident will repay credit and those applicants who are considered by the lender as insufficiently reliable.

Let us assume that the applicant is described by D sociodemographic characteristics which are binary random variables, $\mathbf{x} \in \{0, 1\}^D$. Additionally, by $y \in \{0, 1\}$ we denote the credit status, e.g., whether the credit was repaid or not. We denote the minority class as *positive*, $y = 1$, and the majority class as negative, $y = 0$. In the considered context of the credit scoring this choice of naming classes may seem unreasonable, because typically the majority class corresponds to the applicants who repaid the credit, but this convention is widely used in the case of the imbalanced data phenomenon (He & Garcia, 2009).

Variable	Range	Points	Applicant's characteristic
balance of current account	no current account	17	0
	<300$	30	0
	[300, 1000) $	42	1
	>=1000 $	50	0
previous credits	no credits	23	0
	all paid	41	0
	current credit paying on time	32	1
	problems with current credit	15	0
purpose of credit	new car	32	0
	used car	20	1
	repair	35	0
	other	15	0
age of applicant	<18	0	0
	[18, 35]	31	0
	(35,60]	40	1
	>60	23	0
		APPLICANT'S SCORE:	134

Fig. 1. A scoring table for an exemplary credit applicant.

The problem of the credit scoring can be formalized as follows. We are interested in finding a discriminant function $f(\mathbf{x}, \boldsymbol{\alpha})$ which approximates decision boundaries determined by the true predictive distribution $p(y \mid \mathbf{x})$, where $\boldsymbol{\alpha} \in \mathbb{R}^{D+1}$ denotes adaptive parameters. Further, we restrict ourselves to comprehensible models, more precisely, *scoring tables*. The scoring table associates each input with a fixed amount of points which corresponds to importance (relevancy) of the input. A credit applicant obtains a certain amount of points basing on the scoring table, which is called the applicant's score, and if the score is above the given threshold, an appropriate credit decision is made. An exemplary scoring table is given in Fig. 1.

More formally, the scoring table can be used as a classifier which can be represented as a linear threshold discriminant function:

$$f(\mathbf{x}, \boldsymbol{\alpha}) = \begin{cases} 1, & \mathbf{w}^\top \mathbf{x} \leqslant \tau \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\boldsymbol{\alpha} = \{\tau, \mathbf{w}\}$, and the weights $\mathbf{w}$ represent *relevancy of inputs*, τ is a threshold or *cutting point*.[2] Further, we call $\mathbf{w}^\top \mathbf{x}$ the *score* of the object $\mathbf{x}$.

The scoring table is a comprehensible model because of its simple interpretation. Each i^{it} input can get a fixed amount of points equal w_i and the cutting point corresponds to a minimal sum of points at each the positive decision about the credit consumer is made. Moreover, in the case of refusing the credit to an applicant the reason of the decision can be easily explained by the weights and the cutting point.

An additional advantage of the scoring table is that it can be almost effortlessly modified by a human expert. Both the weights $\mathbf{w}$ and the cutting point τ can be tuned by hand, i.e., if the domain expert decides that some value is inappropriate, she can freely increase or decrease the values of the parameters.

The most important issue in the proposed approach is learning of the discriminant function, here, the scoring table, from data $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$. In other words, for given N pairs $(\mathbf{x}_n, y_n)$ we aim at determining the parameters $\boldsymbol{\alpha} = \{\tau, \mathbf{w}\}$.

Since the weights represent relevancy of inputs or, in other words, the amount of points the inputs get, the weights can be seen as probabilities of the inputs for given class, i.e.,

[2] For simplicity, we assume the weights $\mathbf{w}$ take values in $[0, 1]^D$. Naturally, they can be rescaled to take any values from $\mathbb{R}^D_+$ but then the threshold needs to be rescaled too.

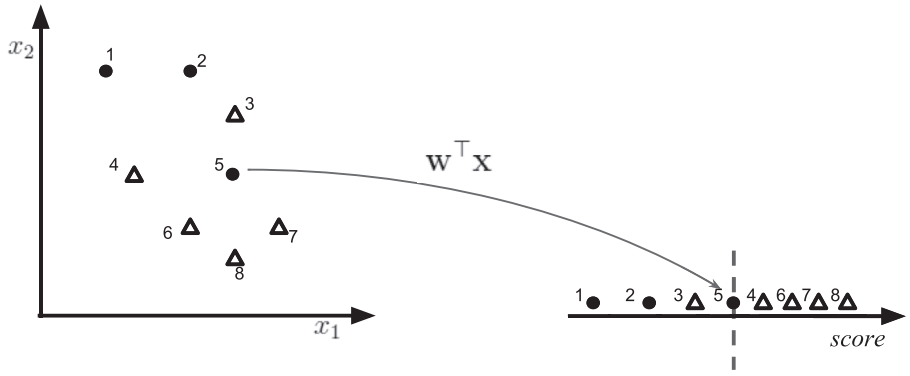

Fig. 2. On the left examples are presented and on the right – examples sorted according to the score. The cutting point is depicted with a dashed line in the figure on the right.

$w_i = p(x_i = 1|y_k = 1)$, for $i = 1 \dots D$. However, because of tge limited number of observations and the complex relationships among the inputs, it is very troublesome to estimate these probabilities sufficiently well. We propose to take advantage of the ClassRBM, which is proven to be the universal approximator of the binary random variables, and approximate the weights using the Eq. (6), i.e., $w_i \approx p(x_i = 1|\mathbf{x}_{\setminus i} = \mathbf{0}, y_k = 1, \boldsymbol{\theta})$,[3] where the parameters $\boldsymbol{\theta}$ need to be learnt first (see Section 2.1.3).

The determination of the cutting point is essential in order to discriminate the credit applicants. The way of establishing the threshold can be influenced by the deficiencies of the training dataset, e.g., the imbalanced data. Typically, the number of applicants who repaid the credit is much higher than the number of those who failed. Any learning method which does not include this issue will produce biased estimates. Therefore, for determined weights, we propose the following procedure for determining the cutting point:

1. For each nth datum calculate the score, i.e., $\mathbf{w}^\top\mathbf{x}_n$.
2. Sort the examples according to their scores.
3. For each example in the dataset $\mathcal{D}$, set the score of this example as the cutting point and calculate the value of *geometric mean* using the scoring table and the data $\mathcal{D}$. The geometric mean is defined as follows (Kubat & Matwin, 1997; Kubat, Holte, & Matwin, 1997)[4]:

$$Gmean = \sqrt{\frac{TP}{TP+FN}\frac{TN}{TN+FP}}. \tag{14}$$

4. Set the cutting point τ to be the score of the example for which the highest value of the *Gmean* was obtained.

An example of the application of this procedure is presented in Fig. 2.

The *Gmean* is defined as a square root of the *True Positive Rate* (*TPR*, called *Sensitivity*):

$$TPR = \frac{TP}{TP+FN} \tag{15}$$

and the *True Negative Rate* (*TNR*, called *Specificity*):

$$TNR = \frac{TN}{TN+FP}. \tag{16}$$

The *Gmean* metric can be seen as a measure of balanceness between correct classification of positive class and negative class considered separately.

We decided to use the geometric mean because it is usually applied in order to resist the imbalanced data (Kubat & Matwin, 1997; Kubat et al., 1997). However, it is possible to consider other criteria, e.g., classification accuracy or the area under the ROC (AUC):

$$AUC = \frac{1}{2}\left(1 + \frac{TP}{TP+FN} - \frac{FP}{FP+TN}\right), \tag{17}$$

but then the obtained values of parameters may be biased towards the majority class.

3. Experiments

In this section we present results of experimental studies in which we compare our approach with other classification methods that can be applied to the problem of the credit risk assessment.

3.1. Experiment setup

3.1.1. Datasets

We evaluate the presented approach on four datasets from the credit scoring domain: *German Credit Data*, modified *Australian Credit Data*,[5] *Kaggle Credit Data*, and *Short-Term Loans Data* (see Table 1 for details). These datasets vary in the imbalance ratio,[6] number of instances, and number of attributes. First two of the considered datasets are taken from the UCI ML Repository,[7] *Kaggle Credit Data*, that is highly influenced by the imbalanced data phenomenon, was used in the Kaggle competition,[8] and the last dataset is obtained from a Polish financial institution.[9] For each of the dataset the vector of the attributes was transformed to the binary inputs because of the need of training the ClassRBM and the character of the scoring table comprehensible model. In the experiments, neither additional data preprocessing nor input selection techniques were applied.

3.1.2. Learning details

We performed learning the ClassRBM using *Stochastic Gradient Descent* with the gradient calculated for the regularized objective function (11) and the mini-batches of size 100. We applied the *Nesterov's Accelerated Gradient* technique which is a kind of the momentum term (Sutskever, Martens, Dahl, & Hinton, 2013). In order to determine model parameters we performed the model

[3] Usually the parameters w_i are rescaled and rounded for clarity and convenience.

[4] *TP* is the number of positive examples classified as positive, *FN* is the number of positive examples classified as negative, *FP* is the number of negative examples classified as positive, *TN* is the number of negative examples classified as negative.

[5] The original dataset was modified by eliminating about 70% of positive examples in order to obtain highly imbalanced data.

[6] The ratio between negatives and positives.

[7] <http://archive.ics.uci.edu/ml/datasets.html>.

[8] <http://www.kaggle.com/c/GiveMeSomeCredit>.

[9] For confidentiality reasons, we are obliged to keep the name of the financial institution unpublished.

Table 1
The number of examples, the number of inputs, and the imbalance ratio for datasets used in the experiments.

Dataset	Number of examples	Number of attributes	Imbalance ratio
German credit data	1000	78	2.33
Australian credit data	471	64	4.35
Kaggle credit data	150,000	59	13.96
Short-term loans data	1146	41	7.13

selection using the validation set and parameters' values as follows: learning rate equal $\eta \in \{0.001, 0.01, 0.1\}$, number of hidden units equal $M \in \{25, 100, 500\}$, momentum coefficient $\beta \in \{0.5, 0.9\}$, and regularization coefficient equal $\lambda \in \{0.001, 0.01, 0.1\}$. In all experiments the following setting obtained the best performance: $\eta = 0.001,\ M = 100,\ \beta = 0.5,\ \lambda = 0.001$.

3.1.3. Evaluation methodology

We compared the performance of the credit scoring table (ST) trained using the approach proposed in this paper with the results gained by the following non-interpretable reference methods: AdaBoost, Bagging, Multilayer Perceptron (MLP), Support Vector Machines (SVM), and Logistic Regression. We also took under consideration typical comprehensible models like classification and regression trees (CART), RIPPER and J48. In order to determine model parameters we performed the model selection using the validation set and collection of parameters' values, e.g., number of base learners of an ensemble equal $\{10, 20, 40\}$. Moreover, we evaluated the performance of ClassRBM, as a standalone classifier, that can be also used for credit repayment prediction (see Section 2.1.2 for details).

In order to compare the considered methods, we used the following evaluation criteria: **AUC** given by (17), **Gmean** given by (14), **TPR** given by (15), and **TNR** given by (16). As a testing methodology we used 10-fold cross-validation. Additionally, we repeated the experiment 10 times for *German*, *Australian*, and *Short-Term Loans* datasets. For *Kaggle* data we did not repeat the experiment because it was unnecessary due to the sufficient number of examples.

In order to evaluate the significance of the obtained results we applied statistical tests for each dataset (except *Kaggle*) separately. For the best performing model we used the Holm–Bonferroni method (Demšar, 2006; Holm, 1979) that is implemented to counteract the problem of multiple comparisons. First, the set of pairwise Wilcoxon tests is conducted to calculate the p-values for the hypothesis about the equality of medians of the both samples. Next, the calculated p-values are sorted ascending and the following inequality is examined:

$$p_i \leqslant FWER_i, \tag{18}$$

Table 4
Detailed test results for ST vs. reference methods for *Kaggle*. Best results in bold.

Method	Comprehensible	TPR	TNR	Gmean	AUC
AdaBoost	✗	0.106	0.995	0.325	0.550
Bagging	✗	0.194	0.989	0.438	0.591
MLP	✗	0.229	0.986	0.475	**0.607**
Random forest	✗	0.188	0.982	0.429	0.585
SVM	✗	0.114	0.994	0.336	0.554
Logistic regression	✗	0.192	0.990	0.436	0.591
CART	✓	0.178	0.991	0.420	0.585
RIPPER	✓	0.194	0.989	0.438	0.592
J48	✓	0.180	0.989	0.422	0.585
ClassRBM	✗	0.182	0.991	0.424	0.586
ST	✓	0.515	0.622	**0.566**	0.569

Table 2
Detailed test results for ST vs. reference methods for *German*. Best results in bold.

Method	Comprehensible	TPR	TNR	Gmean	AUC
AdaBoost	✗	0.289 ± 0.036	0.890 ± 0.015	0.506 ± 0.029	0.589 ± 0.015
Bagging	✗	0.426 ± 0.013	0.883 ± 0.009	0.613 ± 0.011	0.655 ± 0.010
MLP	✗	0.517 ± 0.037	0.814 ± 0.013	0.648 ± 0.023	0.665 ± 0.018
Random Forest	✗	0.478 ± 0.028	0.830 ± 0.009	0.630 ± 0.019	0.654 ± 0.015
SVM	✗	0.484 ± 0.010	0.867 ± 0.004	0.648 ± 0.007	**0.676 ± 0.006**
Logistic Regression	✗	0.479 ± 0.015	0.871 ± 0.006	0.646 ± 0.010	**0.675 ± 0.007**
CART	✓	0.382 ± 0.020	0.885 ± 0.010	0.582 ± 0.016	0.634 ± 0.012
RIPPER	✓	0.405 ± 0.031	0.846 ± 0.010	0.585 ± 0.021	0.625 ± 0.014
J48	✓	0.449 ± 0.015	0.831 ± 0.009	0.611 ± 0.010	0.640 ± 0.008
ClassRBM	✗	0.479 ± 0.019	0.872 ± 0.011	0.646 ± 0.011	**0.675 ± 0.007**
ST	✓	0.672 ± 0.027	0.680 ± 0.016	**0.676 ± 0.012**	**0.676 ± 0.012**

Table 3
Detailed test results for ST vs. reference methods for *Australian*. Best results in bold.

Method	Comprehensible	TPR	TNR	Gmean	AUC
AdaBoost	✗	0.566 ± 0.027	0.948 ± 0.010	0.732 ± 0.016	0.757 ± 0.012
Bagging	✗	0.614 ± 0.016	0.930 ± 0.007	0.755 ± 0.011	0.772 ± 0.010
MLP	✗	0.609 ± 0.039	0.905 ± 0.013	0.742 ± 0.021	0.757 ± 0.016
Random Forest	✗	0.515 ± 0.033	0.912 ± 0.008	0.685 ± 0.024	0.713 ± 0.019
SVM	✗	0.545 ± 0.020	0.922 ± 0.006	0.709 ± 0.012	0.734 ± 0.008
Logistic Regression	✗	0.603 ± 0.019	0.939 ± 0.004	0.753 ± 0.013	0.771 ± 0.011
CART	✓	0.608 ± 0.010	0.945 ± 0.002	0.758 ± 0.007	0.776 ± 0.006
RIPPER	✓	0.634 ± 0.012	0.928 ± 0.007	0.767 ± 0.009	0.781 ± 0.008
J48	✓	0.578 ± 0.027	0.919 ± 0.005	0.729 ± 0.017	0.749 ± 0.013
ClassRBM	✗	0.747 ± 0.005	0.910 ± 0.003	**0.824 ± 0.003**	**0.828 ± 0.003**
ST	✓	0.720 ± 0.013	0.746 ± 0.007	0.733 ± 0.002	0.733 ± 0.002

Table 5
Detailed test results for ST vs. reference methods for *Short-Term Loans*. Best results in bold.

Method	Comprehensible	TPR	TNR	Gmean	AUC
AdaBoost	✗	0 ± 0	1 ± 0	0 ± 0	0.500 ± 0
Bagging	✗	0.001 ± 0.002	0.999 ± 0.001	0.008 ± 0.025	0.5 ± 0.001
MLP	✗	0 ± 0	1 ± 0.001	0 ± 0	0.5 ± 0
Random Forest	✗	0.045 ± 0.018	0.959 ± 0.005	0.203 ± 0.040	0.502 ± 0.010
SVM	✗	0 ± 0	1 ± 0	0 ± 0	0.5 ± 0
Logistic Regression	✗	0 ± 0	1 ± 0	0 ± 0	0.5 ± 0
CART	✓	0 ± 0	1 ± 0	0 ± 0	0.5 ± 0
RIPPER	✓	0 ± 0	0.998 ± 0.002	0 ± 0	0.499 ± 0.001
J48	✓	0 ± 0	1 ± 0	0 ± 0	0.5 ± 0
ClassRBM	✗	0.004 ± 0.002	0.999 ± 0.001	0.044 ± 0.025	0.502 ± 0.001
ST	✓	0.550 ± 0.013	0.632 ± 0.004	**0.589 ± 0.006**	**0.591 ± 0.006**

where p_i represents ith p-value in the sequence. The factor $FWER_i$ is familywise error rate and for the given significance level α it can be calculated using the equation:

$$FWER_i = \frac{\alpha}{I+1-i}, \tag{19}$$

where I is the number of tested hypothesis. If the inequality (18) is satisfied, then the hypothesis about medians equality is rejected.

3.1.4. Implementation details

We implemented an experimental environment in the Java programming language in which all experiments were performed using the same testing methodology. For the reference methods we used the classifiers implemented in the WEKA library (Hall et al., 2009). The ClassRBM and the scoring table were implemented in Matlab®.

3.2. Results

The results of the experiment are gathered in Tables 2–5 and for the comprehensible models they are also presented in Fig. 3. Our approach (ST) outperformed the reference methods on *German*, *Kaggle* and *Short-Term Loans* datasets, which are characterized by high imbalance ratio and low general predictive accuracy.

The results for the pairwise tests between ST and the reference methods are presented in Table 6 (for *German* data) and 9 (for *Short-Term Loans* data). For the set of Wilcoxon test the p-values are lower than the corresponding values $FWER_i$ for a given significance rate α equal 0.05. Therefore, with the probability equal 95% we can say that our approach performs significantly better than the other methods considered in the experimental studies for the two datasets.

In the case of the *Australian* data, ST performs better than Random Forest and SVM, however, it was defeated by Bagging, Logistic Regression, CART, RIPPER, and ClassRBM (see Table 7). Moreover, it turned out that the ClassRBM outperformed all other methods considered in the experiment which was justified by the Holm–Bonferroni procedure (see Table 8).

Concluding, our approach gains acceptable or even the best results for all datasets. It is especially important to highlight its performance in comparison to other interpretable models, see Tables 2, 4, 5 and Fig. 3, where ST obtains very high value of both the arithmetic mean of **TPR** and **TNR** (i.e. **AUC**) and the geometric mean of **TPR** and **TNR** (i.e. **Gmean**).

3.3. Discussion

ST model achieved the best results among other reference methods for the datasets in which the impact of imbalanced data is noticeable. This model performed worse than other comprehensible models on the *Australian* dataset, but it maintained the **Gmean** value at a satisfactory level. In the opposite to the other interpretable models, the scoring table can be easily modified by an expert to increase the prediction accuracy. Therefore, ST constructed with the proposed approach can be used as a good starting point for handmade tuning by bank analysts, if the prior knowledge can be included or the quality of classification is insufficient.

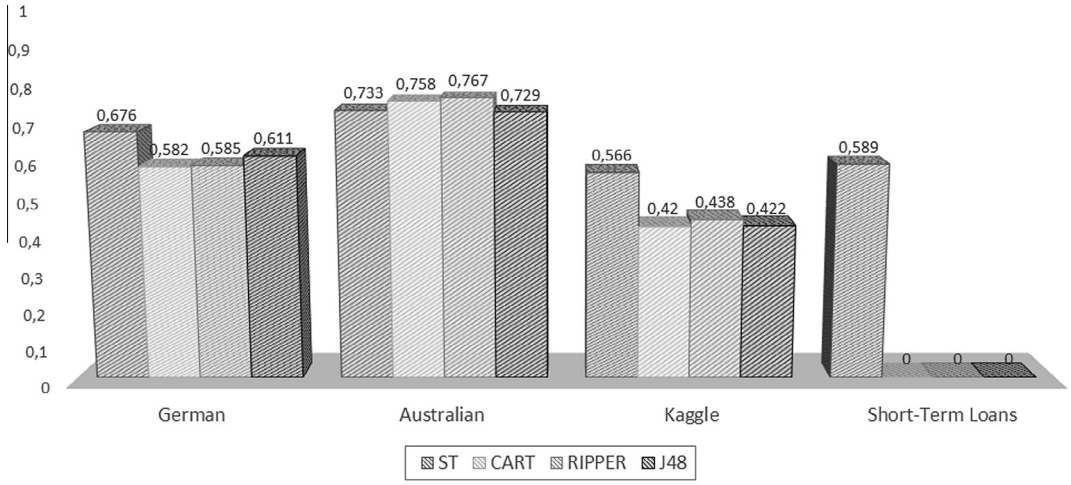

Fig. 3. Bar plot representing comparison among comprehensible methods using average of the *Gmean* evaluation metric obtained on the considered four datasets.

Table 6
Results of the Holm–Bonferroni method with the Wilcoxon test made between ST and reference methods for *German* data.

Methods	*p*-value	FWER	Hypothesis (alpha = 0.05)
ST vs. ADaBoost	0.0020	0.0050	Rejected for ST
ST vs. bagging	0.0020	0.0056	Rejected for ST
ST vs. MLP	0.0020	0.0063	Rejected for ST
ST vs. random forest	0.0020	0.0071	Rejected for ST
ST vs. SVM	0.0020	0.0083	Rejected for ST
ST vs. logistic regression	0.0020	0.0100	Rejected for ST
ST vs. CART	0.0039	0.0125	Rejected for ST
ST vs. RIPPER	0.0039	0.0167	Rejected for ST
ST vs. J48	0.0059	0.0250	Rejected for ST
ST vs. ClassRBM	0.0273	0.0500	Rejected for ST

Table 7
Results of the Wilcoxon test made between ST and reference methods for *Australian* data.

Methods	*p*-value	Hypothesis (alpha = 0.05)
ST vs. ADaBoost	0.9219	Not rejected
ST vs. bagging	0.0020	Rejected for bagging
ST vs. MLP	0.2324	Not rejected
ST vs. random forest	0.0020	Rejected for ST
ST vs. SVM	0.0039	Rejected for ST
ST vs. logistic regression	0.0039	Rejected for logistic regression
ST vs. CART	0.0020	Rejected for CART
ST vs. RIPPER	0.0020	Rejected for RIPPER
ST vs. J48	0.4316	Not rejected
ST vs. ClassRBM	0.0020	Rejected for ClassRBM

Table 8
Results of the Holm–Bonferroni method with the Wilcoxon test made between ClassRBM and reference methods for *Australian* data.

Methods	*p*-value	FWER	Hypothesis (alpha = 0.05)
ClassRBM vs. ADaBoost	0.0020	0.0050	Rejected for ClassRBM
ClassRBM vs. bagging	0.0020	0.0056	Rejected for ClassRBM
ClassRBM vs. MLP	0.0020	0.0063	Rejected for ClassRBM
ClassRBM vs. random forest	0.0020	0.0071	Rejected for ClassRBM
ClassRBM vs. SVM	0.0020	0.0083	Rejected for ClassRBM
ClassRBM vs. logistic regression	0.0020	0.0100	Rejected for ClassRBM
ClassRBM vs. CART	0.0020	0.0125	Rejected for ClassRBM
ClassRBM vs. RIPPER	0.0020	0.0167	Rejected for ClassRBM
ClassRBM vs. J48	0.0020	0.0250	Rejected for ClassRBM
ClassRBM vs. ClassRBM	0.0020	0.0500	Rejected for ClassRBM

The application of the *Gmean* as a cutting point in the proposed procedure for learning the scoring table resulted in comparable values of **TPR** and **TNR**. This is a desirable effect because the prediction is not biased toward the majority class.

4. Conclusions

In this work we present the method for constructing the scoring table which is comprehensible and achieves high predictive performance. In the proposed approach the ClassRBM is trained directly on the data which is further used to compute the relevancy of each attribute that is essential component of the scoring table.

The research contribution of the paper is fourfold. First, the novel application of the ClassRBM to the construction of the scoring table is outlined. Second, original and straightforward manner of learning scoring table by maximizing selected criterion, i.e., *Gmean*, is presented. Third, the application of the *Gmean* criterion leads to constructing scoring table which is resistant to the imbalanced data phenomenon. Fourth, we propose a stand-alone comprehensible model which achieves high predictive performance justified by the empirical studies.

Table 9
Results of the Holm–Bonferroni method with the Wilcoxon test made between ST and reference methods for *Short-Term Loans* data.

Methods	*p*-value	FWER	Hypothesis (alpha = 0.05)
ST vs. ADaBoost	0.0020	0.0050	Rejected for ST
ST vs. bagging	0.0020	0.0056	Rejected for ST
ST vs. MLP	0.0020	0.0063	Rejected for ST
ST vs. random forest	0.0020	0.0071	Rejected for ST
ST vs. SVM	0.0020	0.0083	Rejected for ST
ST vs. logistic regression	0.0020	0.0100	Rejected for ST
ST vs. CART	0.0020	0.0125	Rejected for ST
ST vs. RIPPER	0.0020	0.0167	Rejected for ST
ST vs. J48	0.0020	0.0250	Rejected for ST
ST vs. ClassRBM	0.0020	0.0500	Rejected for ST

It is also worth mentioning that the presented approach has several practical advantages. First of all, once the parameters of the scoring table are determined, the model can be easily implemented in typical bank system. Moreover, the prior knowledge of a human expert can be easily incorporated into the model by modifying scoring points. Additionally, some comprehensible models, such as decision trees, may have complex structure and thus it may become difficult to interpret. In our approach the structure is fixed and depends on the number of attributes.

The proposed scoring table has couple of limitations. The need of binarizing the input variables may lead to losing information in data. However, the input binarization is necessary in learning the ClassRBM and the scoring table. Furthermore, the ClassRBM, which is essential in constructing the scoring table, consists of relatively large number parameters which may result in overfitting the model during learning. Therefore, it is important to apply some kind of regularization technique.

We see several possible extensions of the outlined approach. First, we would like to examine the quality of the proposed model using large number of datasets from other domains. Second, it seems to be interesting to investigate other than *Gmean* criterion in the training procedure of the scoring table. Moreover, the training procedure itself could be modified to increase other quality measures, i.e., *AUC*. Last but not least, it is worth considering to incorporate cost-matrix in the process of constructing the scoring table.

Acknowledgments

The work conducted by Maciej Zięba is co-financed by the European Union within the European Social Fund.

References

Bellotti, T., & Crook, J. (2009). Support vector machines for credit scoring and discovery of significant features. *Expert Systems with Applications, 36*, 3302–3308.

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research, 183*, 1447–1465.

De Bock, K. W., & Van den Poel, D. (2012). Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. *Expert Systems with Applications, 39*, 6816–6826.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research, 7*, 1–30.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter, 11*, 10–18.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society), 160*, 523–541.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*, 1263–1284.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*, 1771–1800.
Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 65–70.
Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*, 847–856.
Huang, S. C. (2011). Using Gaussian process based kernel classifiers for credit rating forecasting. *Expert Systems with Applications, 38*, 8607–8611.
Huang, Y. M., Hung, C. M., & Jiau, H. C. (2006). Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. *Nonlinear Analysis: Real World Applications, 7*, 720–747.
Kubat, M., Holte, R., & Matwin, S. (1997). Learning when negative examples abound. *ECML*, 146–153.
Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *ICML* (pp. 179–186).
Larochelle, H., & Bengio, Y. (2008). Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th international conference on machine learning* (pp. 536–543).
Larochelle, H., Mandel, M., Pascanu, R., & Bengio, Y. (2012). Learning algorithms for the classification restricted Boltzmann machine. *Journal of Machine Learning Research, 13*, 643–669.
Le Roux, N., & Bengio, Y. (2008). Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation, 20*, 1631–1649.
Martens, D., Baesens, B., Van Gestel, T., & Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research, 183*, 1466–1476.
Martens, J., Chattopadhya, A., Pitassi, T., & Zemel, R. (2013). On the expressive power of restricted Boltzmann machines. In *Advances in neural information processing systems 26* (pp. 2877–2885).
Nanni, L., & Lumini, A. (2009). An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 36*, 3028–3033.
Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning*.
West, D. (2000). Neural network credit scoring models. *Computers & Operations Research, 27*, 1131–1152.
Yang, Y. (2007). Adaptive credit scoring with kernel learning methods. *European Journal of Operational Research, 183*, 1521–1536.
Zięba, M., & Świątek, J. (2012). Ensemble classifier for solving credit scoring problems. *Technological Innovation for Value Creation*, 59–66.