2014 13th International Conference on Machine Learning and Applications

Example-Dependent Cost-Sensitive Logistic Regression for Credit Scoring

Alejandro Correa Bahnsen, Djamila Aouada and Björn Ottersten
Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg, Luxembourg
http://www.uni.lu/snt/
Email: {alejandro.correa, djamila.aouada, bjorn.ottersten}@uni.lu

***Abstract*—Several real-world classification problems are example-dependent cost-sensitive in nature, where the costs due to misclassification vary between examples. Credit scoring is a typical example of cost-sensitive classification. However, it is usually treated using methods that do not take into account the real financial costs associated with the lending business. In this paper, we propose a new example-dependent cost matrix for credit scoring. Furthermore, we propose an algorithm that introduces the example-dependent costs into a logistic regression. Using two publicly available datasets, we compare our proposed method against state-of-the-art example-dependent cost-sensitive algorithms. The results highlight the importance of using real financial costs. Moreover, by using the proposed cost-sensitive logistic regression, significant improvements are made in the sense of higher savings.**

***Keywords*—Cost sensitive classification; Credit Scoring; Logistic Regression.**

I. Introduction

The objective in credit scoring is to classify which potential customers are likely to default a contracted financial obligation based on the customer's past financial experience, and with that information decide whether to approve or decline a loan [1]. This tool has become a standard practice among financial institutions around the world in order to predict and control their loans portfolios. When constructing credit scores, it is a common practice to use standard cost-insensitive binary classification algorithms such as logistic regression, neural networks, discriminant analysis, genetic programing, decision trees, among others [2], [3]. However, in practice, the cost associated with approving what is known as a bad customer, i.e., a customer who default his credit loan, is quite different from the cost associated with declining a good customer, i.e., a customer who successfully repay his credit loan. Furthermore, the costs are not constant among customers. This is because loans have different credit line amounts, terms, and even interest rates. Some authors have proposed methods that include the misclassification costs in the credit scoring context [4]–[7]. However, they assume a constant misclassification cost, which is not the case in credit scoring.

The classification framework where the misclassification costs vary across examples is called example-dependent cost-sensitive classification. The approaches that solve this problem are usually based on a cost matrix per example i that adapts the costs to a pair of predicted label c_i and true class label y_i [8], where the prediction c_i is a function of the k features of example i, $X_i = [x_i^1, x_i^2, ..., x_i^k]$. In TABLE I the cost matrix is

TABLE I. Classification cost matrix

	Actual Positive $y_i = 1$	Actual Negative $y_i = 0$
Predicted Positive $c_i = 1$	C_{TP_i}	C_{FP_i}
Predicted Negative $c_i = 0$	C_{FN_i}	C_{TN_i}

presented, where the cost associated with two types of correct classification, namely, true positives C_{TP_i}, and true negatives C_{TN_i}; and the two types of misclassification errors, namely, false positives C_{FP_i}, and false negatives C_{FN_i}, are presented.

With the objective of evaluating an example-dependent cost-sensitive classification problem, we use the cost measure defined in [9]. This measure takes into account the actual costs $C_i = [C_{TP_i}, C_{FP_i}, C_{FN_i}, C_{TN_i}]$ of each example i. Let S be a set of N examples i, $N = |S|$, where each example is represented by the augmented feature vector $X_i^a = [X_i, C_i]$, and labelled using the class label $y_i \in \{0, 1\}$. A classifier f which generates the predicted label c_i for each example i, is trained using the set S.Then the cost of using f on S is calculated by

$$Cost(f(S)) = \sum_{i=1}^{N} \bigg(y_i(c_i C_{TP_i} + (1 - c_i) C_{FN_i}) + (1 - y_i)(c_i C_{FP_i} + (1 - c_i) C_{TN_i}) \bigg). \quad (1)$$

Moreover, by evaluating the cost of classifying all examples as the class with the lowest cost $Cost_l(S) = \min\{Cost(f_0(S)), Cost(f_1(S))\}$ where f_0 refers to a classifier that predicts all the examples in S as belonging to the class c_0, and similarly f_1 predicts all the examples in S as belonging to the class c_1, the cost improvement can be expressed as the cost savings as compared with $Cost_l(S)$.

$$Savings(f(S)) = \frac{Cost(f(S)) - Cost_l(S)}{Cost_l(S)}. \quad (2)$$

Recently, some methods have been proposed to use the different misclassification costs, not only for evaluating but also for training the algorithms. The main approach consists in re-weighting the training examples based on their costs, either by cost-proportionate rejection-sampling [10], or over-sampling [8]. The rejection-sampling approach consists in selecting a random subset S_r by randomly selecting examples from S, and accepting each example i with probability $w_i / \max_{1,\dots,N}\{w_i\}$, where w_i is defined as the expected misclassification error of example i: $w_i = y_i \cdot C_{FN_i} + (1 - y_i) \cdot C_{FP_i}$.

Lastly, the over-sampling method consists in creating a new set S_o, by making w_i copies of each example i. However, cost-proportionate over-sampling increases the training since $|S_o| >> |S|$, and it also may result in over-fitting [11]. Furthermore, none of these methods uses the the full cost matrix but only the misclassification costs.

In a recent paper, we have proposed an example-dependent cost-sensitive Bayes minimum risk (BMR) for credit card fraud detection [9], [12]. The BMR classifier is a decision model based on quantifying tradeoffs between various decisions using probabilities and the costs that accompany such decisions [9], [13]. This is done in a way that for each example the expected losses are minimized. In what follows, we consider the probability estimates p_i as known, regardless of the algorithm used to calculate them. The risk that accompanies each decision is calculated. In the specific framework of binary classification, the risk of predicting the example i as negative is $R(c_i = 0|X_i) = C_{TN_i}(1 - \hat{p}_i) + C_{FN_i} \cdot \hat{p}_i$, and $R(c_i = 1|X_i) = C_{TP_i} \cdot \hat{p}_i + C_{FP_i}(1 - \hat{p}_i)$, is the risk when predicting the example as positive, where $\hat{p}_i$ is the estimated positive probability for example i. Subsequently, if $R(c_i = 0|X_i) \leq R(c_i = 1|X_i)$, then the example i is classified as negative. This means that the risk associated with the decision c_i is lower than the risk associated with classifying it as positive. However, when using the output of a binary classifier as a basis for decision making, there is a need for a probability that not only separates well between positive and negative examples, but that also assesses the real probability of the event [14].

In this paper, we first propose an example-dependent cost matrix for credit scoring, one that incorporates all the real financial costs associated with the lending business. Furthermore, we go beyond the current example-dependent cost-sensitive methods, that works by modifying either the input or output of a cost-insensitive classifier, and propose a method to introduce the example-dependent costs into a logistic regression, by changing the objective function of the model to one that is cost-sensitive. We evaluate the proposed example-dependent cost-sensitive logistic regression using two publicly available credit scoring datasets. The results will shown that the proposed method outperforms the state-of-the-art example-dependent cost-sensitive methods. Furthermore, the source code used for the experiments is publicly available as part of the *CostSensitiveClassification*[1] library.

The remainder of this paper is organized as follows. In Section II, the credit scoring problem and the proposed example-dependent cost-sensitive credit scoring cost matrix, are presented. Afterwards, the proposed cost-sensitive logistic regression is shown in Section III. In Sections IV, the experimental setup and the results of the different experiments are presented. Finally, the conclusions of the paper are given in Section V.

[1]https://github.com/albahnsen/CostSensitiveClassification

II. Credit scoring example-dependent cost matrix

In this section we first present the general concept of credit scoring, then the standard method for calculating the probability threshold. Finally, we present the proposed example-dependent cost matrix for credit scoring.

A. Credit Scoring

In order to mitigate the impact of credit risk and make more objective and accurate decisions, financial institutions use credit scores to predict and control their losses. The objective of a credit score is to estimate the risk of a customer defaulting his contracted financial obligation if a loan is granted, based on past experiences [1]. Formally, a credit score is a statistical model that allows the estimation of the probability $\hat{p}_i = P(y_i = 1|X_i)$ of a customer i defaulting a contracted debt. Additionally, since the objective of credit scoring is to estimate a classifier c_i to decide whether or not to grant a loan to a customer i, a threshold t is defined such that if $\hat{p}_i < t$, then the loan is granted, i.e., $c_i(t) = 0$, and denied otherwise, i.e., $c_i(t) = 1$.

There exists different approaches for defining the probability threshold. The sensitivity versus specificity ($SvsS$) approach is the most widely used among financial institutions [1], where specificity is the true positive rate $F_0(t)$ for a threshold t, and the sensitivity is one minus the false positive rate $F_1(t)$ given a threshold t [15]. In this method the objective is to fix the threshold at the point where the sensitivity is equal to the specificity $F_0(t) = 1 - F_1(t)$, where $F_0(t)$ and $F_1(t)$ are calculated using $F_k(t) = \frac{1}{n_k}|\{(x_i, y_i) \in D_k | \hat{p}_i \leq t\}|$, for $k \in \{0, 1\}$. Lastly, the $SvsS$ threshold t_{SvsS} is found by using $t_{SvsS} = \mathrm{argmin}_t\, |F_0(t) - (1 - F_1(t))|$.

After the classifier c_i is estimated, there is a need to evaluate its performance. In practice, many statistical evaluation measures are used to assess theperformance of a credit scoring model. Measures such as the area under the receiver operating characteristic curve (AUC), Brier score, Kolmogorov-Smirnoff (K-S) statistic, F_1-Score, and misclassification are among the most common [6]. Nevertheless, none of these measures takes into account the business and economical realities that take place in credit scoring. Costs that the financial institution had incurred to acquire customers, or the expected profit due to a particular client, are not considered in the evaluation of the different models.

Initial approaches to include the different costs have been published in recent years, particularly the one proposed by Beling et al. [6], [7], in whichthe costs of misclassification are assigned for each error. Specifically, setting the cost of a false positive C_{FP} to the loan's annual interest rate charged to the customer int_r, the cost of a false negative C_{FN} to the loss given default L_{gd}, which is the percentage of loss over the total credit line when the customer defaulted, and setting to zero the costs of true positive C_{TP} and true negative C_{TN}. Using that, they proposed the expected cost (EC) method to find the probability threshold that minimizes those costs,$t_{ec} = \frac{C_{FN}}{C_{FN} + C_{FP}} = \frac{L_{gd}}{L_{gd} + int_r}$. Nevertheless, this approach assumes a constant cost within examples, which is a strong assumption, since in practice each example carries a very different cost,

TABLE II. Credit scoring example-dependent cost matrix

	Actual Positive $y_i = 1$	Actual Negative $y_i = 0$
Predicted Positive $c_i = 1$	$C_{TP_i} = 0$	$C_{FP_i} = r_i + C^a_{FP}$
Predicted Negative $c_i = 0$	$C_{FN_i} = Cl_i \cdot L_{gd}$	$C_{TN_i} = 0$

given by the different credit limits and conditions of each loan. Consequently, there is a need for an example-dependent cost matrix that takes into account the cost of misclassifying each example.

B. *Credit scoring example-dependent cost matrix*

In order to take into account the varying costs that each example carries, we propose a cost matrix with example-dependent misclassification costs as given in TABLE II. First, we assume that the costs of a correct classification, C_{TP_i} and C_{TN_i}, are zero for every customer i. We define C_{FN_i} to be the losses if the customer i defaults to be proportional to his credit line Cl_i. We define the cost of a false positive per customer C_{FP_i} as the sum of two real financial costs r_i and C^a_{FP}, where r_i is the loss in profit by rejecting what would have been a good customer. The calculation of r_i depends on the loan parameters as detailed in Appendix A. The second term C^a_{FP}, is related to the assumption that the financial institution will not keep the money of the declined customer idle. It will instead give a loan to an alternative customer [16]. Since no further information is known about the alternative customer, it is assumed to have an average credit line $\overline{Cl}$ and an average profit $\overline{r}$. Then, $C^a_{FP} = -\overline{r} \cdot \pi_0 + \overline{Cl} \cdot L_{gd} \cdot \pi_1$, in other words minus the profit of an average alternative customer plus the expected loss, taking into account that the alternative customer will pay his debt with a probability equal to the prior negative rate, and similarly will default with probability equal to the prior positive rate.

III. Cost-sensitive logistic regression

Logistic regression is a classification model that, in the specific context of binary classification, estimates the posterior probability of the positive class, as the logistic sigmoid of a linear function of the feature vector [17]. The estimated probability is evaluated as

$$\hat{p}_i = P(y = 1|X_i) = h_\theta(X_i) = g\left(\sum_{j=1}^{k} \theta^j x_i^j\right), \qquad (3)$$

where $h_\theta(X_i)$ refers to the hypothesis of i given the parameters θ, and $g(\cdot)$ is the logistic sigmoid function, defined as $g(z) = 1/(1 + e^{-z})$.
The problem then becomes on finding the right parameters that minimize a given cost function. Usually, in the case of logistic regression the cost function $J(\theta)$ refers to the negative logarithm of the likelihood, such that

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} J_i(\theta), \qquad (4)$$

where

$$J_i(\theta) = -y_i \log(h_\theta(X_i)) - (1 - y_i)\log(1 - h_\theta(X_i)). \qquad (5)$$

TABLE III. Model parameters

Parameter	Kaggle Credit	PAKDD Credit
Interest rate (int_r)	4.79%	63.0%
Cost of funds (int_{cf})	2.94%	16.5%
Term (l) in months	24	24
Loss given default (L_{gd})	75%	75%
Times income (k)	3	3
Maximum credit line (Cl_{max})	25,000	25,000

Since this cost function is convex [18], it is usually optimized using either maximum likelihood estimator or gradient descent. However, this cost function assigns the same weight to different errors, both false positives and false negatives. As discussed before, this is not the case in many real-world applications. In particular

$$J_i(\theta) \approx \begin{cases} 0 & \text{if} \quad y_i \approx h_\theta(X_i) \\ \inf & \text{if} \quad y_i \approx (1 - h_\theta(X_i)) \end{cases}$$

which in the context of cost-sensitive classification means that $C_{TP_i} = C_{TN_i} \approx 0$ and $C_{FP_i} = C_{FN_i} \approx \inf$.

In order to incorporate the different costs from TABLE I into the logistic regression, first we analyze the expected costs for each case

$$J_i^c(\theta) = \begin{cases} C_{TP_i} & \text{if} \quad y_i = 1 \text{ and } h_\theta(X_i) \approx 1 \\ C_{TN_i} & \text{if} \quad y_i = 0 \text{ and } h_\theta(X_i) \approx 0 \\ C_{FP_i} & \text{if} \quad y_i = 0 \text{ and } h_\theta(X_i) \approx 1 \\ C_{FN_i} & \text{if} \quad y_i = 1 \text{ and } h_\theta(X_i) \approx 0 \end{cases}$$

Finally, we merge the different costs into a cost function which is dependent on new costs:

$$J^c(\theta) = \frac{1}{N}\sum_{i=1}^{N}\Big(y_i(h_\theta(X_i)C_{TP_i} + (1 - h_\theta(X_i))C_{FN_i}) + (1 - y_i)(h_\theta(X_i)C_{FP_i} + (1 - h_\theta(X_i))C_{TN_i})\Big). \qquad (6)$$

IV. Experiments

In this section, the dataset used for the experiments is described. Then the partitioning of the dataset and the algorithms used for credit scoring are given. Lastly, we present the experimental results.

A. *Databases*

For this paper we use two different publicly available credit scoring datasets. The first dataset is the **2011 Kaggle competition Give Me Some Credit**[2], in which the objective is to identify those customers of personal loans that will experience financial distress in the next two years. The second dataset is from the **2009 Pacific-Asia Knowledge Discovery and Data Mining conference (PAKDD) competition**[3]. Similarly, this competition had the objective of identifying which credit card applicants were likely to default and by doing so deciding whether or not to approve their applications. The Kaggle Credit and PAKDD Credit datasets contain information regarding the features, and more importantly about the income of each

[2]http://www.kaggle.com/c/GiveMeSomeCredit/
[3]http://sede.neurotech.com.br:443/PAKDD2009/

TABLE IV. DESCRIPTION OF DATASETS

Database	Set	N	π_1	C_0
Kaggle Credit	Total	112,915	.0674	83,740,181
	Training	44,901	.0675	33,360,130
	Under-sampling	6,090	.5058	33,360,130
	SMOTE	81,177	.4841	436,577,294
	Rejection-sampling	5,108	.4381	29,009,564
	Over-sampling	66,123	.3616	296,515,655
	Validation	33,919	.0668	24,786,997
	Testing	33,732	.0681	25,593,055
PAKDD Credit	Total	38,969	.1988	3,117,960
	Training	15,353	.1997	1,221,174
	Under-sampling	6,188	.4956	1,221,174
	SMOTE	24,554	.4996	1,604,231
	Rejection-sampling	2,776	.3577	631,595
	Over-sampling	33,805	.3393	6,798,282
	Validation	11,833	.2036	991,795
	Testing	11,783	.1930	904,991

example, from which an estimated credit limit Cl_i can be calculated (see Appendix B).

The Kaggle Credit dataset contains 112,915 examples, each one with 10 features and the class label. The proportion of default or positive examples is 6.74%. On the other hand, the PAKDD Credit dataset contains 38,969 examples, with 30 features and the class label, with a proportion of 19.88% positives. This database comes from a Brazilian financial institution, and as it can be inferred from the competition description, the data was obtained around 2004.

Since no specific information regarding the datasets is provided, we assume that they belong to average European and Brazilian financial institutions. This enabled us to find the different parameters needed to calculate the cost measure described in Section II-B. In Table III, the different parameters are shown. In particular, we obtain the average interest rates in Europe during 2013 from the European Central Bank [19], and the average interest and exchange rates in Brazil during 2004 from Trading Economics [20]. Because the income is not in the same currency on both datasets, we convert the PAKDD Credit dataset to Euros. Additionally, we use a fixed loan term l for both datasets, considering that in the Kaggle Credit dataset the class was constructed to predict two years of credit behavior, and because the PAKDD Credit dataset is related to credit cards the term is fix to two years [21]. Moreover, we set the loss given default L_{gd} using information from the Basel II standard[4], k to 3 since it is the average personal loan requests related to monthly income, and the maximum credit limit Cl_{max} to 25,000 Euros.

B. Database partitioning

From the total dataset, three different datasets are extracted: training, validation and testing. Each one containing 50%, 25% and 25% of the observations, respectively. Afterwards, because classification algorithms suffer when the label distribution is skewed towards one of the classes [22], an under-sampling of the positive examples is made, in order to have a balanced class distribution. The under-sampling has proved to be the better approach for such problems, see [22]. A new training dataset containing a balanced number of positive and negative examples is created. Furthermore, recently the synthetic minority over-sampling technique (SMOTE) [23] has also been applied to credit scoring [24]. With SMOTE the objective is to find

[4]http://www.bis.org/publ/bcbsca.htm.

TABLE V. RESULTS ON THE KAGGLE AND PAKDD CREDIT DATASETS OF THE DECISION TREE (DT), LOGISTIC REGRESSION (LR) AND RANDOM FOREST (RF) ALGORITHMS, ESTIMATED USING THE DIFFERENT TRAINING SETS: TRAINING (t), UNDER-SAMPLING (u), SMOTE (s), COST-PROPORTIONATE REJECTION-SAMPLING (r) AND COST PROPORTIONATE OVER-SAMPLING (o)

set	Algorithm	Kaggle Credit dataset	PAKDD Credit dataset
t	DT	19.88	-8.36
	LR	2.87	0.38
	RF	15.83	3.25
u	DT	34.49	-20.0
	LR	43.63	15.81
	RF	49.63	9.65
s	DT	3.46	-4.56
	LR	40.12	15.43
	RF	3.01	0.0
r	DT	33.57	7.59
	LR	33.14	22.97
	RF	**50.01**	**30.1**
o	DT	19.6	8.95
	LR	33.56	23.03
	RF	21.69	23.26

(those models with the highest savings are marked as bold)

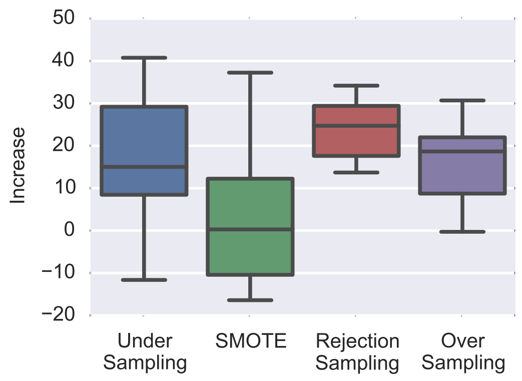

Fig. 1. Average increase in savings of the algorithms trained using the under-sampled, SMOTE, cost-proportionate rejection-sampling and cost-proportionate over-sampling compared against the ones trained in the training set. In average the rejection-sampling method is the one that gives the highest improvement.

a new balanced dataset which includes all the majority class examples and a synthetic over-sampled replica of the minority class examples, such that the new set is balanced.

Lastly, we also applied the cost-proportionate rejection-sampling [10] and cost-proportionate over-sampling [8], described in Section I. TABLE IV, summarizes the different datasets. It is important to note that the different sampling procedures were only applied to the training set since the validation and test sets must reflect the real fraud distribution.

C. Results

For the experiments we used three different well known machine learning algorithms, decision tree (DT), logistic regression (LR) and a random forest (RF). Using the implementation of Scikit-learn [25], the three algorithms are trained in the five different training sets: training (t), under-sampling (u), SMOTE (s), cost-proportionate rejection-sampling (rs) and cost-proportionate over-sampling (o). Furthermore, with

TABLE VI. Results on the two datasets of the standard sensitivity versus specificity method ($SvsS$), the minimum cost threshold (MC) and the Bayes minimum risk (BMR), using the probabilities estimated using a DT, LR and RF, trained on the five sets.

set	Algorithm	Kaggle Credit dataset	PAKDD Credit dataset
t	$DT-SvsS$	-4.8	-91.11
	$DT-MC$	-4.8	-91.11
	$DT-BMR$	13.47	27.22
	$LR-SvsS$	25.57	15.78
	$LR-MC$	24.94	15.78
	$LR-BMR$	29.14	29.38
	$RF-SvsS$	49.05	3.09
	$RF-MC$	48.57	9.24
	$RF-BMR$	49.39	30.11
u	$DT-SvsS$	-4.8	-91.11
	$DT-MC$	-4.8	-91.11
	$DT-BMR$	34.58	26.49
	$LR-SvsS$	19.15	0.07
	$LR-MC$	11.69	0.07
	$LR-BMR$	45.25	29.6
	$RF-SvsS$	9.16	0.58
	$RF-MC$	0.0	0.1
	$RF-BMR$	51.47	31.14
s	$DT-SvsS$	-4.8	-91.11
	$DT-MC$	-4.8	-91.11
	$DT-BMR$	-0.54	26.84
	$LR-SvsS$	19.16	0.44
	$LR-MC$	11.66	0.44
	$LR-BMR$	43.26	29.63
	$RF-SvsS$	32.57	-91.11
	$RF-MC$	32.57	-42.11
	$RF-BMR$	43.11	26.75
r	$DT-SvsS$	-4.8	-91.11
	$DT-MC$	-4.8	-91.11
	$DT-BMR$	33.58	25.99
	$LR-SvsS$	9.18	0.22
	$LR-MC$	9.18	0.22
	$LR-BMR$	35.6	29.98
	$RF-SvsS$	12.13	0.0
	$RF-MC$	12.13	0.0
	$RF-BMR$	50.57	28.11
o	$DT-SvsS$	-4.8	-91.11
	$DT-MC$	-4.8	-91.11
	$DT-BMR$	11.77	27.12
	$LR-SvsS$	22.45	0.28
	$LR-MC$	19.33	0.28
	$LR-BMR$	43.09	29.53
	$RF-SvsS$	48.09	1.85
	$RF-MC$	42.65	1.85
	$RF-BMR$	49.38	28.03

(those models with the highest savings are marked as bold)

the different estimated probabilities in conjunction of the BMR model, as reference the sensitivity versus specificity method ($SvsS$) and the minimum cost threshold (MC). Lastly, we compare our proposed cost-sensitive logistic regression ($CSLR$) against the previous results. In particular we minimize the $CSLR$ cost function (6) using binary genetic algorithms [26], setting the number of chromosomes to 100, percentage of mutations to 25%, number of elite to 10 and performing single point cross-over. All models are evaluated using the savings as defined in (2).

The results are given in TABLE V. For the Kaggle Credit dataset, the best model measured by savings is an RF estimated using the rejection-sampling set. Furthermore, the difference in savings between these two models is 6.4%. Analyzing the results in the PAKDD Credit dataset, the best model measured by savings is RF on the rejection-sampling set. In this case the difference in savings between the two models is 7%. Additionally, it is observed that there is not one algorithm that consistently outperforms the others in the different sets.

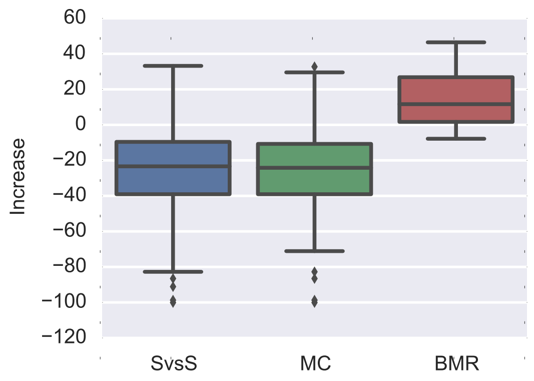

Fig. 2. Average increase in savings of the algorithms using the sensitivity versus specificity method ($SvsS$), the minimum cost threshold (MC) and the Bayes minimum risk (BMR), versus the standard algorithms is made. It is observed, that the only method that generates an increase in savings is the BMR.

Furthermore, as can be observed in Fig. 1, on average the rejection-sampling method is the one that gives the highest improvement in savings over using the training set. The methods trained on the $SMOTE$ set perform the worst. Subsequently, using the estimated probabilities from each model, we evaluate the $SvsS$, MC and BMR methods. The results are shown in TABLE VI. It is observed that, for both databases, the best model measured by savings is the RF trained using the under-sampled set, leading to an increase in savings of around 1% in both cases. Interestingly, the second best model is the RF on the training set. This corroborates the findings in [12], where the same algorithms were applied in a credit card fraud detection database. In Fig. 2, a comparison of the average increase in savings of the different methods is presented. It is observed, that the only method that generates an increase in savings is the BMR. In fact, when using the standard $SvsS$ method, there is a destruction in value from the use of the algorithm.

We evaluate our proposed $CSLR$ algorithm described in Section III. Results are shown in TABLE VII. On the Kaggle Credit database, we observe that the savings of the best model are 54.41%, almost 3% higher than the savings using the BMR method. Similarly, in the case of the PAKDD Credit database, the best model arise to savings of 34.83%, again an almost 3% increase compared against using the BMR method. The results also show, that using the BMR with the probabilities estimated with the $CSLR$ leads to higher saving. Nevertheless, using just the $CSLR$ model also yields to significantly good results measured by savings.

Finally, in Fig. 3, we compared the results in savings of the LR trained using the five different sets, and the $CSLR$. In all cases the best results are found using the $CSLR$, significantly outperforming the LR model. Leading to a clear conclusion of the need to use a model that not only incorporates the real financial costs after training, but also during this phase.

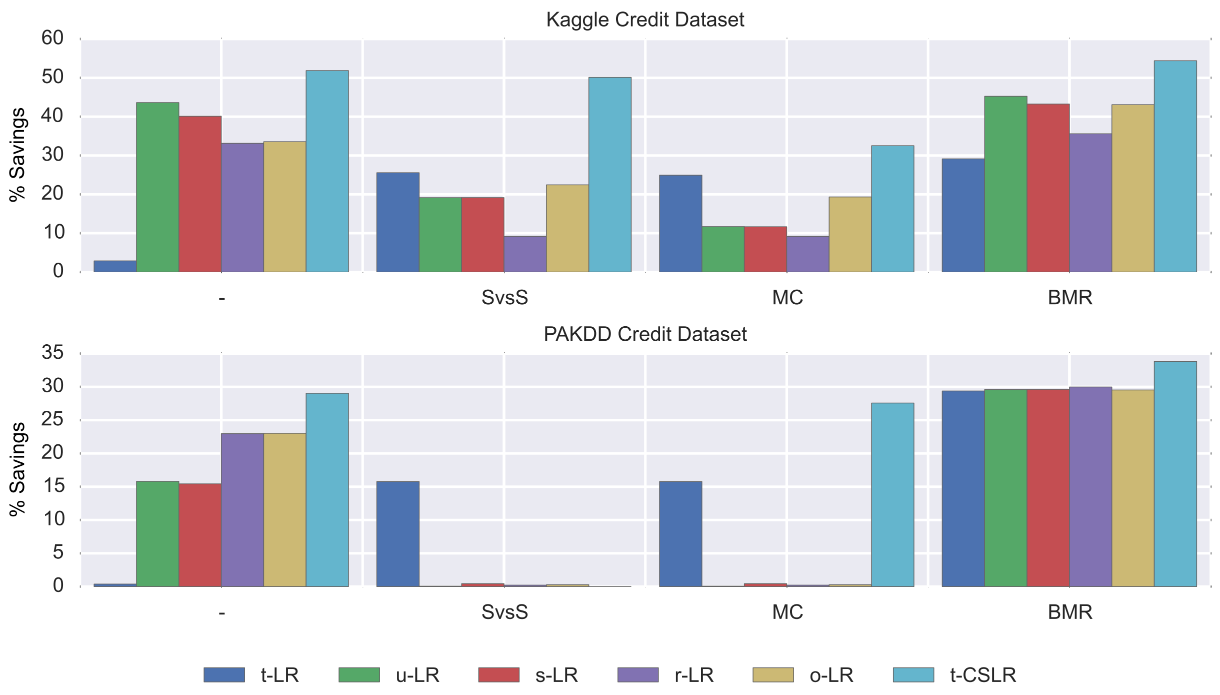

Fig. 3. Comparison of the $SvsS$, MC and BMR, in savings using the probabilities of a LR trained using the five different sets, and a $CSLR$. In all cases the best results are found using the $CSLR$, significantly outperforming the LR model.

V. Conclusion

In this paper we have shown the importance of using the real example-dependent financial costs, associated with the credit business when selecting a credit score model. Moreover, our evaluations confirmed that including the costs of each example and using an example-dependent cost-sensitive method such as Bayes minimum risk classifier, leads to better results in the sense of higher savings, regardless of the algorithm used for estimating the probabilities. Also, when using our proposed cost-sensitive logistic regression, further savings are found, which confirms out belief that there is a need to use a model that not only incorporates the real financial costs after training, but also during the training phase. Additionally, the proposed example-dependent cost matrix could be expanded to include other potentially example-dependent features, such as the loss given default or the interest rate in the calculation of the cost. Furthermore, the model could be adjusted to include features that may also depend on the estimated probabilities. For example, when determining the credit line, financial institutions often rely on the risk level for its calculation, similarly when assigning the interest rate, or calculating the loss given default of a portfolio.

TABLE VII. Results on the two datasets of the cost-sensitive logistic regression estimated using the different training sets, and then using the $SvsS$, MC and BMR methods.

set	Algorithm	Kaggle Credit dataset	PAKDD Credit dataset
t	$CSLR$	51.87	29.04
	$CSLR-SvsS$	50.12	0.0
	$CSLR-MC$	32.53	27.57
	$CSLR-BMR$	54.41	33.83
u	$CSLR$	-4.8	-12.17
	$CSLR-SvsS$	0.0	0.0
	$CSLR-MC$	0.0	0.0
	$CSLR-BMR$	21.65	32.46
s	$CSLR$	31.44	-6.98
	$CSLR-SvsS$	0.0	0.0
	$CSLR-MC$	0.0	0.0
	$CSLR-BMR$	34.82	32.85
r	$CSLR$	-4.8	15.91
	$CSLR-SvsS$	0.0	0.0
	$CSLR-MC$	0.0	0.0
	$CSLR-BMR$	13.16	34.54
o	$CSLR$	-4.8	20.26
	$CSLR-SvsS$	0.0	0.0
	$CSLR-MC$	0.0	0.0
	$CSLR-BMR$	38.83	34.83

Acknowledgment

Funding for this research was provided by the Fonds National de la Recherche, Luxembourg.

References

[1] R. Anderson, *The Credit Scoring Toolkit : Theory and Practice for Retail Credit Risk Management and Decision Automation*. Oxford University Press, 2007.

[2] D. J. Hand and W. E. Henley, "Statistical Classification Methods in Consumer Credit Scoring: A Review," *Journal of the Royal Statstical Society. Series A (Statistics in Society)*, vol. 160, no. 3, pp. 523–541, 1997.

[3] A. Correa Bahnsen and A. Gonzalez Montoya, "Evolutionary Algorithms for Selecting the Architecture of a MLP Neural Network: A Credit Scoring Case," in *IEEE 11th International Conference on Data Mining Workshops*. Ieee, Dec. 2011, pp. 725–732.

[4] T. Verbraken, C. Bravo, R. Weber, and B. Baesens, "Development and application of consumer credit scoring models using profit-based

classification measures," *European Journal of Operational Research*, vol. 238, no. 2, pp. 505–513, Oct. 2014.

[5] R. Alejo and V. Garc, "Making Accurate Credit Risk Predictions with Cost-Sensitive MLP Neural Networks," in *Advances in Intelligent Systems and Computing*, ser. Advances in Intelligent Systems and Computing, J. Casillas, F. J. Martínez-López, R. Vicari, and F. De la Prieta, Eds. Heidelberg: Springer International Publishing, 2013, vol. 220, pp. 1–8.

[6] P. Beling, Z. Covaliu, and R. M. Oliver, "Optimal scoring cutoff policies and efficient frontiers," *Journal of the Operational Research Society*, vol. 56, no. 9, pp. 1016–1029, Jul. 2005.

[7] R. Oliver and L. Thomas, "Optimal score cutoffs and pricing in regulatory capital in retail credit portfolios," 2009.

[8] C. Elkan, "The Foundations of Cost-Sensitive Learning," in *Seventeenth International Joint Conference on Artificial Intelligence*, 2001, pp. 973–978.

[9] A. Correa Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Cost Sensitive Credit Card Fraud Detection using Bayes Minimum Risk," in *International Conference on Machine Learning and Applications*. Miami, USA: IEEE, 2013.

[10] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Third IEEE International Conference on Data Mining*. IEEE Comput. Soc, 2003, pp. 435–442.

[11] C. Drummond and R. Holte, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *intl conf. machine learning, workshop learning from imbalanced data sets II*, 2003.

[12] A. Correa Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Improving Credit Card Fraud Detection with Calibrated Probabilities," in *Proceedings of the fourteenth SIAM International Conference on Data Mining*, no. January 2012. Philadelphia, USA: Society for Industrial and Applied Mathematics, 2014, pp. 677 – 685.

[13] G. Jayanta K., D. Mohan, and S. Tapas, "Bayesian Inference and Decision Theory," in *An Introduction to Bayesian Analysis*. Springer New York, Apr. 2006, vol. 13, no. 2, pp. 26–63.

[14] I. Cohen and M. Goldszmidt, "Properties and Benefits of Calibrated Classifiers," in *Knowledge Discovery in Databases: PKDD 2004*. Springer Berlin Heidelberg, 2004, vol. 3202, pp. 125–136.

[15] J. Hernandez-Orallo, P. Flach, and C. Ferri, "A Unified View of Performance Metrics : Translating Threshold Choice into Expected Classification Loss," *Journal of Machine Learning Research*, vol. 13, pp. 2813–2869, 2012.

[16] G. N. Nayak and C. G. Turvey, "Credit Risk Assessment and the Opportunity Costs of Loan Misclassification," *Canadian Journal of Agricultural Economics*, vol. 45, no. 3, pp. 285–299, 1997.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information science and statistics. Springer, 2006, vol. 4, no. 4.

[18] K. P. Murphy, *Machine Learning A Probabilistic Perspective*. MIT Press, 2012.

[19] ECB, "European Central Bank," 2014.

[20] T. Economics, "Brazil statistics," 2014.

[21] D. Lawrence and A. Solomon, *Managing a Consumer Lending Business*. Solomon Lawrence Partners, 2012.

[22] J. V. Hulse and T. M. Khoshgoftaar, "Experimental Perspectives on Learning from Imbalanced Data," in *International Conference on Machine Learning*, 2007.

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, no. 16, pp. 321–357, Jun. 2002.

[24] V. García, A. I. Marqués, and J. S. Sánchez, "Improving risk predictions by preprocessing imbalanced credit data," in *19th International Conference, ICONIP*, Doha, Qatar, 2012, pp. 68–75.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] R. Haupt and S. Haupt, *Practical genetic algorithms*, second edi ed. New Jersey: John Wiley & Sons, Inc., 2004.

APPENDIX

A. Calculation of a loan profit

The profit per customer r_i is calculated as the present value of the difference between the financial institution gains and expenses, given the credit line Cl_i, the term l_i and the financial institution lending rate int_{r_i} for customer i, and the financial institution of cost funds int_{cf}.

$$r_i = PV(A(Cl_i, int_{r_i}, l_i), int_{cf}, l_i) - Cl_i, \tag{7}$$

with A being the customer monthly payment and PV the present value of the monthly payments, which are calculated using the time value of money equations [21],

$$A(Cl_i, int_{r_i}, l_i) = Cl_i \frac{int_{r_i}(1+int_{r_i})^{l_i}}{(1+int_{r_i})^{l_i}-1}, \tag{8}$$

$$PV(A, int_{cf}, l_i) = \frac{A}{int_{cf}}\left(1-\frac{1}{(1+int_{cf})^{l_i}}\right). \tag{9}$$

B. Calculation of the credit limit

There exists several strategies to calculate the Cl_i depending on the type of loans, the state of the economy, the current portfolio, among others [1], [21]. Nevertheless, given the lack of information regarding the specific business environments of the considered datasets, we simply define Cl_i as

$$Cl_i = \min\left\{k \cdot Inc_i, Cl_{max}, Cl_{max}(debt_i)\right\}, \tag{10}$$

where Inc_i and $debt_i$ are the monthly income and debt ratio of the customer i, respectively, k is a parameter that defines the maximum Cl_i in times Inc_i, and Cl_{max} the maximum overall credit line. Lastly, the maximum credit line given the current debt is calculated as the maximum credit limit such that the current debt ratio plus the new monthly payment does not surpass the customer monthly income. It is calculated as

$$Cl_{max}(debt_i) = PV\left(Inc_i \cdot P_m(debt_i), int_{r_i}, l_i\right), \tag{11}$$

and

$$P_m(debt_i) = \min\left\{\frac{A(k \cdot Inc_i, int_{r_i}, l_i)}{Inc_i}, (1-debt_i)\right\}. \tag{12}$$