Comparison of ensemble cost sensitive algorithms: application to credit scoring prediction

Meryem Saidi
High School of Management
GBM Laboratory, Tlemcen University
miryem.saidi@gmail.com

Mostafa El Habib Daho
Biomedical Engineering Laboratory
Tlemcen University
mostafa.elhabibdaho@gmail.com

Nesma Settouti
Biomedical Engineering Laboratory
Tlemcen University
nesma.settouti@gmail.com

Mohammed El Amine Bechar
Biomedical Engineering Laboratory
Tlemcen University
am.bechar@gmail.com

Abstract

In recent years, the increase in the demand for credit leads the financial institutions to consider artificial intelligence and machine learning techniques as a solution to make decisions in a reduced time. These decision support systems reach good results in classifying loan applications into good loans and bad loans. Albeit they suffer of some limitations, mainly, they consider that the misclassification errors have the same financial impact.
In this work, we study the performance of ensemble cost sensitive algorithms in reducing the most expensive errors. We apply these techniques on German credit data. By comparing the different algorithms, we demonstrate the effectiveness of cost sensitive ensemble algorithms in determining the potential loan defaulters to reduce the financial cost.

Keywords Cost sensitive learning, credit scoring, ensemble algorithms.

In: Proceedings of the 3rd Edition of the International Conference on Advanced Aspects of Software Engineering (ICAASE18), Constantine, Algeria, 1,2-December-2018, published at http://ceur-ws.org

1 Introduction

Credit scoring is the process of analyzing credit files, to decide the creditworthiness of an individual. Distinguishing a good applicant for a loan from a bad one is important to cut financial institution's losses [AEW13]. The use of machine learning tools allows auditors to analyze large amounts of information for evaluating the credit risk in a reasonable time [Yu17].

These algorithms tend to decrease the classification error and assume that all misclassification's have the same cost. However, the cost for labeling a positive example as negative is different from the cost for labeling a negative example as positive. Indeed, approving a bad loan is much more costly than rejecting a potentially good loan [KBC16]. Indeed, if a loan can not full fill its loan obligations this may result in negative impacts on bank profits and big financial losses. However, if a good loan is rejected, it causes lower profits losses.

These algorithms tend to decrease the classification error and assume that all misclassification's have the same cost. However, the cost for labeling a positive example as negative is different from the cost for labeling a negative example as positive. Indeed, approving a bad loan is much more costly than rejecting a potentially good loan [KBC16]. Indeed, if a loan can not full fill its loan obligations this may result in negative impacts on bank profits and big financial losses. However, if a good loan is rejected, it causes lower profits losses.

On the other hand, credit datasets are highly imbalanced which worsens the situation. Traditional machine learning algorithms tend to maximize accuracy by considering most of the cases as good loans (majority class), thus causing significant default loss.

Motivated by the non-uniform cost classification problem, the data mining researchers propose new cost-sensitive learning approaches for taking into account the misclassification costs or other types of costs such as acquisition cost or computer cost [Tur00, Dom99, Elk01, Mar02]. Some studies have been conducted on the use of cost-sensitive (CS) learning in credit scoring as a CS-boosted tree [XLL17], CS-Neuronal network [AGM+13], CS-decision tree [BAO15] and CS-logistic regression [BAO14].

The objective of this study is to compare the effectiveness of different techniques to assist the loan officer in screening out potential loan defaulters in the credit environment. The rest of paper is framed as follows: Section 2 describes the used algorithms. Experimental results and discussion are presented in Section 3, Finally, we conclude with a summary of results and directions for future works.

2 Research methodology

In this section, we present the cost sensitive learning principle and the selected algorithms for the evaluation.

2.1 Cost sensitive learning

There are several methods to deal with unequal misclassification costs. The first one is to use a learning algorithm that takes into account the costs when building the classifier. The second strategy is to use sampling (oversampling and under-sampling) to alter the class distribution of the training data. In cost-sensitive classification, the misclassification cost plays an important role in the learning process. A cost matrix is used to encode the penalty of misclassifying an example from one class as another [Dom99]. Table 1 represents a misclassification cost matrix, used to obtain the cost of a false positive (FP), false negative (FN), true positive (TP), and true negative (TN).

Table 1: Cost matrix

	Predict	
Actual	Positive	Negative
Positive	C_{TP}	C_{FN}
Negative	C_{FP}	C_{TN}

The positive class is the most expensive class and $C(i, j)$ denote the cost of predicting an instance from class i as class j. Usually, $C(i, i)$ have a null or a negative cost and the FN cost is more expensive than a FP cost ($C(0, 1) > C(1, 0)$). The best evaluation metrics in cost sensitive learning is total cost (see equation 1).

$$Total\ Cost = (FN * C_{FN}) + (FP * C_{FP}) \quad (1)$$

The cost-sensitive learning methods can be categorized into two categories: direct and indirect.

Direct methods

In the direct method, the learning algorithm is itself cost-sensitive (CS). The CS learning algorithms use the misclassification cost during the learning process. There are several works on cost-sensitive learning algorithms such as ICET [Tur95], an evolutionary algorithm using a misclassification cost in the fitness function. Many cost sensitive decision tree approaches were proposed [MZ12, Tur95, DHR+06, FCPB07, ZL16]. In [KK98], the authors perform a comparative study of different cost-sensitive neural networks. Other researches propose cost sensitive ensemble methods [KW14, KWS14, SKWW07, MSV11, Mar99].

Indirect methods

On the other hand, the indirect methods, called Cost-sensitive meta-learning, convert existing cost-insensitive learning algorithms into cost-sensitive ones without modifying them. The cost-sensitive meta-learning technique, propose two major mechanisms: a pre-process instance sampling or weighting of the training dataset and a threshold adjusting of the output of a cost-insensitive algorithm [Zha08]. In this category, we can cite MetaCost [Dom99] which manipulate the training set labels, Costing [ZLA03], Weighting [Tin02]or Empirical Thresholding [SL06].

2.2 Used algorithms

Classification and regression trees (CART)

Proposed by Breiman et al. [BFOS84], CART is a binary decision tree. This algorithm processes continuous and categorical attributes and target. CART uses the Gini splitting rule to search the best possible variable to split the node into two child nodes and grow the trees to their maximum size until no splits are possible.

Bagging

Bootstrap Aggregation (Bagging), is one of the earliest and simplest ensemble algorithms [Bre96]. The learners are fitted to bootstrap replicates of the training set by sampling randomly from original set with replacement i.e.: an observation x_i may appear multiple times in the sample. After the base learners have been fit, the aggregated response is the majority vote.

Hence, Bagging has no memory, it is easily parallelize (as can be seen in Algorithm 1).

Algorithm 1 CS-bagging Algorithm

Input: $S = ((x_1, y_1), ..., (x_m, y_m)))$, P: the number of classifier to train.
for p:=1 **to** P **do**
S_p = Bootstrap(S), i.i.d. sampling with replacement from S.
h_p = TrainClassifier(S_t).
Add h_p = to the ensemble.
end for

Boosting

Proposed by Schapire [SFBL97, SF12], boosting is a technique for sequentially combining multiple base classifiers whose combined performance is significantly better than that of any of the base classifiers. Each base classifier is trained on data that is weighted based on the performance of the previous classifier and each classifier votes to obtain a final decision.

CS-CART

To generate a cost-sensitive CART algorithm Breiman et al. [BFOS84] modify the class probabilities, $P(i)$ used in the information gain measure. Instead of estimating $P(i)$ by N_i/N, it is weighted by the relative cost.

$$P(i) = C_{ij} * (N_i/N)/\sum_j cost(j)(N_j/N)$$

The cost of misclassifying an example of class j as class i is : $cost(j) = \sum C_{ij}$

CS-bagging

It learns the different individual classifiers then it uses the available classifiers for a better estimation of the posterior probabilities according to a voting scheme. This approach is applicable regardless of the underlying learning method [Shi15].

MetaCost

This algorithm was proposed by [Dom99]. MetaCost estimates the class probabilities then relabel the training instances to minimize the expected cost. Finally, a new classifier is built on the relabeled dataset.

We test different combinations of the former algorithms:

- The insensitive cost classifiers: CART, BAGGING of CART, BOOSTING of CART.

Algorithm 2 CS-bagging Algorithm

Input: $S = ((x_1, y_1), ..., (x_m, y_m)))$, P: the number of classifier to train.
for p:=1 **to** P **do**
S_p = Bootstrap(S), i.i.d. sampling with replacement from S.
h_p = TrainClassifier(S_t).
Add h_p = to the ensemble.
end for
for p:=1 **to** P **do**
$\hat{Y}_p(w)$ =Set the prediction with h_p .
end for
According to the proportions observed on the $P's$ prediction, we have an estimate of $P(Y = y_k/X(w))$.
Make the prediction which minimizes the cost.

Algorithm 3 MetaCost Algorithm

Input: $S = ((x_1, y_1), ..., (x_m, y_m)))$, L: cost matrix, H : classifier.
Estimate the class probabilities $P(y_i|x_i)$.
Relabel $y_i = argmin \sum j = 1kP(j|x_i)L(\gamma, j)\forall i$.
$T = H(x, y)$.
Output: T.

- One phase Cost classifiers : CS-CART, BAGGING CS-CART, CS-BAGGING CART, MULTI COST CART, BOOSTING CS-CART.
- Two phases Cost classifier: CS-BAGGING CS-CART

3 Experimentation

3.1 Dataset

The empirical evaluation was made on the German credit scoring dataset from the UCI Machine Learning Repository. This dataset consists of 20 features and 1000 instances including 700 instances of credit-worthy applicants and 300 instances of insolvent customers who should not have been granted credit. This dataset is provided with a cost matrix,

Table 2: Cost matrix

	Predict	
Actual	Insolvent	Creditworthy
Insolvent	0	5
Creditworthy	1	0

3.2 Results and discussion

In this section, we present the results obtained by the different algorithms on the German Credit dataset. All computations were performed under Tanagra [1].

[1]http://eric.univ-lyon2.fr/ ricco/tanagra/fr/tanagra.html

Table 3: Classification Results

	Error	Specificity	Sensibility	Cost	Friedman test
CART	27.57	88.52	35.87	47.68	7
CS-CART	39.34	49.76	83.70	26.91	3.5
BAGGING CART	**21.93**	**91.87**	46.74	39.16	6.62
BAGGING CS-CART	36.88	55.50	80.43	27.35	4.76
CS-BAGGING CART	38.87	51.67	82.61	27.06	4
CS-BAGGING CS-CART	43.85	41.63	**89.13**	**25.71**	3
MULTICOST CART	40.53	49.76	81.52	28.40	5
BOOSTING CART	**21.93**	87.08	57.61	33.18	5.87
BOOSTING CS-CART	37.21	55.50	79.35	28.10	5.25

We compare the performances of the methods with different metrics: error, misclassification cost, sensitivity and specificity.

$$Error = \frac{FP}{(TP+FP)} + \frac{FN}{(FN+TN)}$$

$$Cost = FP * C_{FP} + FN * C_{FN}$$

$$Specificity = \frac{TP}{(TP+FN)}$$

$$Sensitivity = \frac{TN}{(FP+TN)}$$

Table 3 presents the general results of the nine algorithms. Following the recommendation of [TSTL12], we employ the non-parametric Freidman test to compare the classifiers. The Friedman test ranks the algorithms; to the best performing is the rank of 1, the second best is the rank 2, etc. The last column depicts the statistical test.

A number of conclusions emerge from this table. First, it emphasizes the superiority of ensemble methods compared to the individual classifier. When we consider the classification error, the best performances are reached by classical bagging and boosting. However, these algorithms focus on improving the classification accuracy at the expense of the minority class. So, they obtain a low sensibility which increases the cost.

On the other hand, a CS-bagging of CS-CART obtains the lowest misclassification cost followed by the individual classifier CS-CART. In this case, the statistical improvement is not significant (just 1.19). We can consider that this little improvement not worth the the computational cost. However, in some cases a small gain in performances represents a great gain in economical benefits. Albeit this technique obtains the highest classification error.

Figure 1 compares the average results of the different methods. In figure 2 and 3, we can see the results error vs cost and specificity vs sensitivity for each classifier. Considering those values, we can suppose that boosting-CART offers a good trade-off between error and misclassification cost followed by bagging-CART. But, it is clear that even if the ensemble cost-insensitive algorithms obtain generally good results, they promote a good classification of the majority class at the expense of the minority class (insolvent loans). However, when we carry out a bagging of cost sensitive trees the recognition of the insolvent loan is higher without a major lose in specificity. On the other hand, the CS-bagging obtains better sensibility but lower specificity.

From a general point of view, if we consider both misclassification error and cost, we can say that the winner in this study is a bagging of CS-trees.

4 Conclusion

In recent years, the number of insolvent loans has increased due to the financial crisis. It becomes necessary for banks to find new methods for the evaluation credit application. Machine learning techniques have been used to perform financial decision making. However, these methods intended to minimize the misclassification error and assume that the different errors are equals. The cost sensitive techniques are used to handle the misclassification cost in many real world problems.

In this paper, we compare the performance of different cost-sensitive and cost-insensitive ensemble algorithms in determining the creditworthiness of an individual. The experiments drew the following conclusions (1) the ensemble approaches obtain better results than individual classifier; (2) the insensitive approaches reached the best classification accuracy but since the class distribution is highly imbalanced the minority class (insolvent loan) is less well recognized; (3) the cost sensitive approaches intended to reduce the cost at the expense of the accuracy.

Finally, we found that the cost sensitive bagging algorithm offers the best trade-off between accuracy and misclassification cost. For future research, we aim to use techniques to handle imbalanced datasets and experiment with other cost sensitive algorithms.

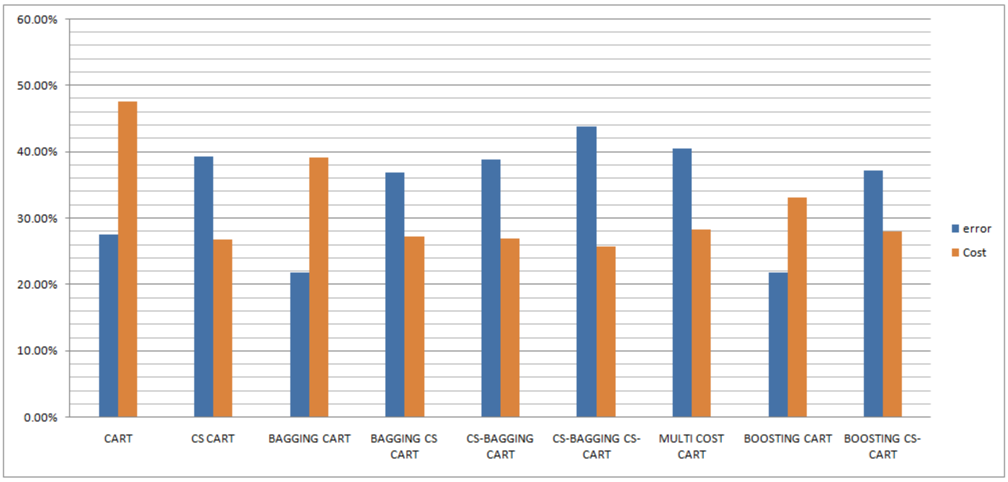

Figure 1: Error classification Vs misclassification cost

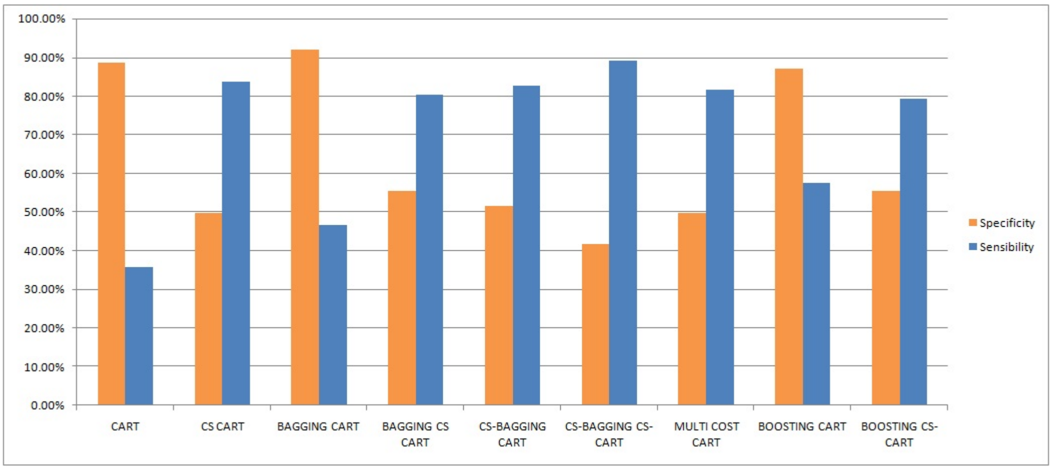

Figure 2: Sensitivity Vs Specificity

References

[AEW13] Shweta Arya, Catherine Eckel, and Colin Wichman. Anatomy of the credit score. *Journal of Economic Behavior & Organization*, 95:175 – 185, 2013.

[AGM+13] R. Alejo, V. García, A. I. Marqués, J. S. Sánchez, and J. A. Antonio-Velázquez. Making accurate credit risk predictions with cost-sensitive mlp neural networks. In Jorge Casillas, Francisco J. Martínez-López, Rosa Vicari, and Fernando De la Prieta, editors, *Management Intelligent Systems*, pages 1–8, Heidelberg, 2013. Springer International Publishing.

[BAO14] Alejandro Correa Bahnsen, Djamila Aouada, and Bjrn Ottersten. Example-dependent cost-sensitive logistic regression for credit scoring. In *13th International Conference on Machine Learning and Applications*, 2014.

[BAO15] Alejandro Correa Bahnsen, Djamila Aouada, and Bjrn Ottersten. Example-dependent cost-sensitive decision trees. *Expert Systems with Applications*, 42(19):6609 – 6619, 2015.

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification And Regression Trees.* Chapman and Hall, New York, 1984.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[DHR+06] J.V. Davis, J. Ha, C.J. Rossbach, H.E. Ramadan, and E. Witchel. Cost-sensitive decision tree learning for forensic classification. In *Proceedings of the 17th European Conference on Machine Learning,*, page 622629, 2006.

[Dom99] Pedro M. Domingos. Metacost: a general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[Elk01] Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pages 973–978, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[FCPB07] A. Freitas, A. Costa-Pereira, and P. Brazdil. Cost-sensitive decision trees applied to medical data. *Data Warehousing and Knowledge Discovery*, page 303312, 2007.

[KBC16] Yeonkook J. Kim, Bok Baik, and Sungzoon Cho. Detecting financial misstatements with fraud intention using multi-class cost-sensitive learning. *Expert Systems with Applications*, 62:32 – 43, 2016.

[KK98] M. Kukar and I. Kononenko. Cost-sensitive learning with neural networks. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence*, Chichester, NY., 1998.

[KW14] Bartosz Krawczyk and Michał Woźniak. Evolutionary cost-sensitive ensemble for malware detection. In *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14*, pages 433–442, Cham, 2014. Springer International Publishing.

[KWS14] Bartosz Krawczyk, Micha Woniak, and Gerald Schaefer. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Applied Soft Computing*, 14:554 – 562, 2014.

[Mar99] D.D. Margineantu. Building ensembles of classifiers for loss minimization. In *Proceedings of the 31st Symposium on the Interface, Models, Predictions and Computing*, pages 190–194, 1999.

[Mar02] Dragos Dorin Margineantu. *Methods for Cost-sensitive Learning.* PhD thesis, Oregon State University, Corvallis, OR, USA, 2002. AAI3029569.

[MSV11] H. Masnadi-Shirazi and N. Vasconcelos. Cost-sensitive boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):294309, 2011.

[MZ12] F. Min and W. Zhu. A competition strategy to cost-sensitive decision trees. *Rough Sets and Knowledge Technology*, pages 359–368, 2012.

[SF12] R.E. Schapire and F. Freund. *Boosting: Foundations and Algorithms.* The MIT Press, 2012.

[SFBL97] R. E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. In *Machine Learning: Proceedings of the Fourteenth International Conference*, 1997.

[Shi15] S.A. Shilbayeh. *Cost sensitive meta learning.* PhD thesis, School of computing, science and engineering university of salford manchester, UK, 2015.

[SKWW07] Yanmin Sun, Mohamed S. Kamel, Andrew K.C. Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358 – 3378, 2007.

[SL06] V. S. Sheng and C. X. Ling. Thresholding for making classifiers cost-sensitive. In *Proceedings of the st national conference on artificial intelligence*, Boston, Massachusetts, 2006.

[Tin02] K. M. Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659 – 665, 2002.

[TSTL12] Bogdan Trawinski, Magdalena Smetek, Zbigniew Telec, and Tadeusz Lasota. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *Applied Mathematics and Computer Science*, 22(4):867–881, 2012.

[Tur95] P. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research (JAIR)*, 2:369–409, 1995.

[Tur00] Peter D. Turney. Types of cost in inductive concept learning. In *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, volume cs.LG/0212034, 2000.

[XLL17] Yufei Xia, Chuanzhe Liu, and Nana Liu. Cost-sensitive boosted tree for loan evaluation in peer-to-peer lending. *Electronic Commerce Research and Applications*, 24:30 – 49, 2017.

[Yu17] Xiaojiao Yu. Machine learning application in online lending risk prediction. *ArXiv e-prints*, July 2017.

[Zha08] Huimin Zhao. Instance weighting versus threshold adjusting for cost-sensitive classification. *Knowledge and Information Systems*, 15(3):321–334, Jun 2008.

[ZL16] H. Zhao and X. Li. A cost sensitive decision tree algorithm based on weighted class distribution with batch deleting attribute mechanism. *Information sciences*, 2016.

[ZLA03] Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 435–, Washington, DC, USA, 2003. IEEE Computer Society.