Tree-based Cost Sensitive Methods for Fraud Detection in Imbalanced Data

G. Metzler[1,3], X. Badiche[3], B. Belkasmi[3], E. Fromont[2], A.Habrard [1] and M. Sebban[1]

1. Univ. Lyon, UJM-Saint-Etienne, Laboratoire Hubert Curien UMR CNRS 5516, F-42023, Saint-Etienne, France
2. Univ. Rennes 1, IRISA/Inria, 35042 Rennes cedex, France
3. BLITZ inc., France

Context

Bank fraud detection is a difficult classification problem where the number of fraud is much smaller than the number of genuine transactions.
Blitz Business Service is company specialized in check fraud detection. The company is analyzing millions of transactions per year for its customers:

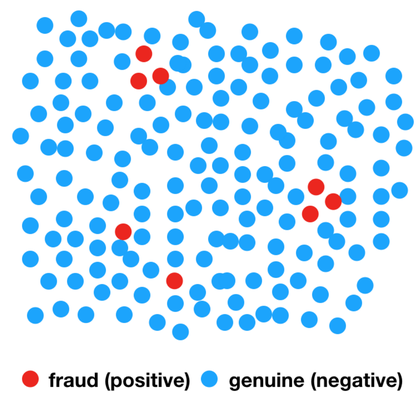
- Around 3.5 millions of transactions in six months for one customer.
- Only 6.5k frauds, which represent 0.16% of transactions $\Longrightarrow$ **imbalanced data**

The company is advising its customers by saying which transactions they should accept or not using a model based on decision trees. However, even if the model is efficient, it is based on the simple error rate loss function which is not suitable in a imbalanced setting. Furthermore, such methods are not easy to understand for non experts.
The notion of **benefits** has then be introduced, using a cost-sentive approach, because it has more sense for the retailers. It also gives them the possiblity to manage the weights of each transactions to be closer from the reality than a simple classification error. By doing so, we are able to increase the retailer's benefits by 1.43%.

Imbalanced learning and Cost-Sensitive Methods

Imbalanced Learning

- m_+ (resp. m_-) number of fraudulent (resp. genuine) transactions, where $m_+ >> m_-$,
- $y_i \in \{0, 1\}$ is the label: 0 for genuine and 1 for fraud,
- error based model $\Longrightarrow$ high accuracy but everything is predicted genuine.



Solution: assign more weight to the minority class (frauds) $\Longrightarrow$ use cost-sensitive methods

Cost Sensitive Learning

Use of a cost-sensitive matrix [1] to defined the weight of each class and/or instances: (i) fraud and genuine transactions have not the same weight and (ii) each transaction has its own weight:

	Pred. fraud	Pred. genuine
Actual fraud	c_{TP_i}	c_{FN_i}
Actual genuine	c_{FP_i}	c_{TN_i}

where:

- $c_{TP_i} = 0$
- $c_{FP_i} = \rho \cdot r \cdot m - p$
- $c_{FN_i} = (r - c) \cdot m$
- $c_{TN_i} = r \cdot m$

- m : the amount of the transaction,
- p : measure the disappointment of the customer,
- r : average rate of benefits,
- ρ : probability of continuing the transaction if the check is refused.

Objective: Maximizing the profits of the retailer, i.e. maximize:

$$\sum_{i=1}^{m} [y_i(\hat{y}_i c_{TP_i} + (1 - \hat{y}_i)c_{FN_i}) + (1 - y_i)(\hat{y}_i c_{FP_i} + (1 - \hat{y}_i)c_{TN_i})],$$

Tree-based Cost Sensitive Methods

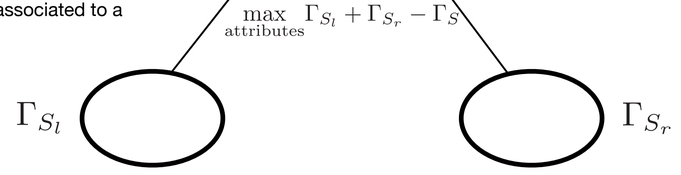
Cost-Sensitive Trees

$$\Gamma_S = \sum_{i \in S_-} \left(\frac{m_+}{m} c_{FP_i}(x_i) + \frac{m_-}{m} c_{TN_i}(x_i)\right) + \sum_{i \in S_+} \left(\frac{m_+}{m} c_{TP_i}(x_i) + \frac{m_-}{m} c_{FN_i}(x_i)\right)$$

Each splits are made using the processus shown here

S collection of example in each node or leaf

γ profit associated to a leaf



We continue to split until we reach the maximum depth or the minimum number of instances in a node

Compute the profits γ_0, γ_1 in each leaf l and predict « fraud » if: $\gamma_1 \geq \gamma_0$

$$\gamma_1(l) = \frac{1}{|l|}\left(\sum_{i:x_i \in l \cap S_-} c_{FP_i} + \sum_{i:x_i \in l \cap S_-} c_{TP_i}\right) \qquad \gamma_0(l) = \frac{1}{|l|}\left(\sum_{i:x_i \in l \cap S_-} c_{TN_i} + \sum_{i:x_i \in l \cap S_+} c_{FN_i}\right)$$

Trees are finally combined and several decision rules are tested

Cost-Sensitive Gradient Boosting

We build a model which estimate the probability p_i of an instance of being a fraud.

Use of the boosting approach introduced by Friedman [4] where we combine weak learners $F_t = F_{t-1} - \alpha_t f_t$
Use a « gradient descent » in the space of the predictions. Each models f_t are trained on the residuals:

$$r_i = g_t = -\left[\frac{\partial L(y, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)}\right] \quad \text{and} \quad (f_t, \alpha_t) = \underset{\alpha, f}{\text{argmin}} \sum_{i=1}^{m} (r_i - \alpha f(x_i))^2$$

Using a Bayes Rule for classification [3], an instance is then predicted positive (or fraudulent) if:

$$p_i > \frac{c_{TN_i} - c_{FP_i}}{c_{TP_i} - c_{FN_i} + c_{TN_i} - c_{FP_i}} = s_i$$

We need to approximate the indicator function which is not differentiable. We use the following upper bound:

$$\mathbb{I}_{p_i > s_i} \leq \left(\frac{1 - s_i}{s_i}\right)^{1/2} \left(\frac{p_i}{1 - p_i}\right)^{1/2} = e^{\hat{F}_i} \quad \text{where} \quad p_i = \frac{1}{1 + \frac{1 - s_i}{s_i} e^{-2\hat{F}_i}}$$

From our 0-1 loss: $L(y \mid p) = -\frac{1}{m} \sum_{i=1}^{m} (y_i c_{TP_i} + (1 - y_i)c_{FP_i}) \mathbb{I}_{p_i > s_i} + (y_i c_{FN_i} + (1 - y_i)c_{TN_i}) \mathbb{I}_{p_i \leq s_i}$

↓

To an exponentiel loss: $L(y \mid p) \leq \tilde{L}(y \mid F) = \frac{1}{m} \sum_{i=1}^{m} (1 - s_i) y_i e^{-\hat{F}_i} + s_i (1 - y_i) e^{\hat{F}_i}$

Implementation using the XGBoost

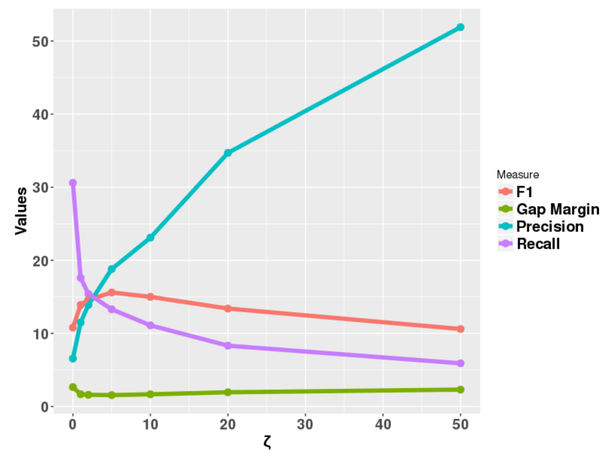
Experiments and Practical Evaluation

Protocol: Train/Validation set 6 months, Test set 4 months. Several algorithms, using the notion of profits, are compared with the current model **RF** (fraud if 9/24 trees say so):

- **RF**$_{maj}$: leaves are labeled according to the majority class then a majority vote is done
- **RF**$_{maj-mar}$: leaves are labeled to maximize the profit then a majority vote is used.
- **RF**$_{mean-mar}$: leaves are labeled to maximize the profit then we use the average profit.
- **GB**$_{tune}$: learned to minimize logistic loss, the threshold is computed w.r.t. to different criteria
- **GB**$_{margin}$: a direct implementation of the presented method.

Experiments	Gap max profits	Precision	Recall	F_1
$\mathbf{RF}$	2.99%	68.1%	5.66%	10.5%
$\mathbf{RF_{maj}}$	2.88%	73.8%	4.71%	8.86%
$\mathbf{RF_{maj-mar}}$	1.81%	30.2%	10.6%	15.7%
$\mathbf{RF_{mean-margin}}$	1.87%	30.3%	9.52%	14.5%
$\mathbf{GB_{tune-Pre}}$	3.01%	**61.0%**	6.49%	11.7%
$\mathbf{GB_{tune-mar}}$	2.26%	19.1%	**16.6%**	**17.8%**
$\mathbf{GB_{tune-F1}}$	2.70%	45.4%	9.24%	15.4%
$\mathbf{GB_{margin}}$	**1.56%**	18.8%	13.3%	15.6%

Gap to the maximal margin of each algorithm. In this table, the value of ζ was set to 5. The results are separated into two groups: Random Forest models and Gradient Boosting models.



Study the influence of the parameter ζ in the definition of $C_{FP_i} = rm - \zeta$.

Conclusion

We have presented several strategies based on cost-sensitive methods to improve the current model of the company which take to notion of "benefits" into account. We have seen that it gives the possibility to reach higher performance in terms of both "benefits" and F-measure which is a standard measure used in imbalanced scenarios. The presented framework also has the advantage to be clearer for the retailers.
A perspective consists of using more informations about the customers: its past or the tickets using recurrent neural networks or dimensionality reduction methods.

References

[1] Bahnsen, A.C. et al., Fraud Detection by Stacking Cost-Sensitive Decision Trees, *DSCS* (2017).
[2] Breiman, L. et al.: Classification and Regression Trees, Wadsworth and Brooks, Monterey, CA (1984).
[3] Elkan, C., The Foundations of Cost-Sensitive Learning, *IJCAI* (2001)
[4] Friedman, J.H., Greedy Function Approximation: A Gradient Boosting Machine, *Annals of Statistics* (2000)

Acknowledgements