Financial Fraud Detection Using Machine Learning Techniques

Yordan Ivanov

# 1.Introduction

What we provide here is an extensive study of machine learning methods on different both real-world and simulated datasets in an attempt to provide better guidelines for fraud detection.

* specify the methods that are going to be used in the study

# 2. Literature Review

# 3. Methodology

Classification is one of the most widely used model framework used for the application of machine learning techniques in terms of FFD (Ngai et al. 2011). Some of the most common classification techniques include logistic regression, neural networks, support vector machine and decision trees.

## 3.1. Preliminaries

In the current section, we will give a description of the machine learning techniques that we apply to predict fraudulent transactions.

Let us define our binary classification dataset as , where represents an n-dimensional data point and represents the label of the class of that data point, . Let represent the vector of features and the vector of dependent variables.

## 3.2. Machine Learning Algorithms

### 3.2.1. Logistic Regression

The logistic regression framework falls under the category of generalized linear models and allows the prediction of discrete outcomes. Then, by defining the probability of a transaction being fraudulent by , we can portray the relationship between the dependent and independent variables as follows:

The number of independent variables is indexed by *p*. After manipulating (), we can also see that

with the LHS being called the logit. Using equation (), we will predict the probabilities of a transaction being fraudulent i.e. . The fitting of a logistic regression is done by the method of maximum likelihood (see Appendix). The logistic regression has been among the most widely used framework in fraud detection (Ngai et al. 2011) due to simplicity of ease of implementation, but it does have its shortcomings - it tends to underperform when there are multiple or non-linear decision boundaries (*SEARCH FOR SOME PAPER OR BOOK ON IT?*)

### 3.2.2. Neural Networks

### 3.2.3. Support Vector Machines

Support Vector Machines, developed by Vapnik et. al. (Cortes and Vapnik 1995), have become a popular machine learning method that has seen its implementation rise in various domains that require the use of classification models (Batuwita and Palade 2013). Among the factors for its success is the fact that the SVMs are linear classifiers, which work in a high-dimensional feature that represents a non-linear mapping of the input space of the problem being dealt with (Bhattacharyya et al. 2011). Working in a high-dimensional feature space has its benefits - often, the problem of non-linear classification in the original input space is transformed to a linear classification task in the high-dimensional feature space.

The goal of the SVM classifier consists of finding the optimal separating hyperplane, which manages to effectively separate the observations from the data into two classes. As mentioned above, the observations are initially transformed by a nonlinear mapping function . Thus, we can write a possible separating hyperplane that resides in the transformed higher dimensional feature space by:

with the weight vector normal to the hyperplane.

We will further use two variations of the SVM soft margin optimization problem - one that assigns the same cost for missclassification of the different classes and one that penalizes more the missclassification of the minority class.

#### Non-cost sensitive learning

For the same missclassification cost case, we can write the soft optimization problem as follows:

The slack variables hold for missclassified examples. Thus, the penalty term can be perceived as the total number of missclassified observations of the model. Thusm from (), we can see that there are two goals - maximizing the margin the minimizing the number of missclassifications. The cost parameter C controls the trade-off between them, i.e. assigned misclassification cost. The quadratic optimization problem in () can be represented by a dual Lagrange problem and then solved:

are the Lagrange multipliers also satisfying the Karush-Kuhn-Tucker (KKT) conditions (see Appendix). Thanks to another one of the strenghts of SVM - kernel representation - we don’t need to explicitly know the mapping function , but by applying a kernel function (i.e. ), we can rewrite () as:

The solution then gives us for the optimal values of and , while is determined from KKT. The data points that have different than zero are called the support vectors. Thus, the decision function can be written was:

#### Cost Sensitive learning

The regular SVM model has been effectively implemented when the dataset used has balanced classes, however it fails to produce good results when applied on imbalanced data (Batuwita and Palade 2013). When trained on a dataset with extremely imbalanced classes, the SVM framework could produce so skewed hyperplances that all observations are recognized as the majority class (Akbani, Kwek, and Japkowicz 2004, Veropoulos et al. (1999)). This is due to the fact that when we take the soft margin optimization problem, we try to maximize the margin and minimize the penalty for the misclassifications. As we consider a constant C for all training examples, the minimization of the penalty is achieved through the minimization of all misclassifications. However, when the used dataset suffers from imbalanced classes, the majority class density would be higher than the minority class density, even when considering the class boundary region (through which the ideal hyperplane would pass).

Thus, we consider here the application of the Different Error Costs (DEC) variation of the SVM algorithm proposed by Veropoulos et al. (1999). The DEC method introduces different misclassification costs - for the minority and for the majority class. With the inclusion of the higher misclassification cost for the minority class observations, the imbalanced class effect could be brought down. The soft margin optimization problem then has the following form:

The Dual Lagrange optimization form is the same as before, with the exception of replacing with for . The and are the Lagrange multipliers. The solution of the DEC dual Lagrangian problem follows the same outline as in the normal form.

#### Kernels

As mentioned before, we used a kernel function () in order to transform the dual Lagrange problem. The advantages of using kernel functions are computational or in some cases it allows for computations that otherwise would be impossible (James et al. (2013)). For the purpose of this study, we are using the linear and radial kernels. The radial kernel shows good performance on non-linear class separation. They have the following representations:

### 3.2.4. Tree-Based Methods

Tree-based methods involve segmentation of the feature space into a set of regions and then fitting a simple model to each one. Even though are not too complex conceptually, they are still a very powerful method (Friedman, Hastie, and Tibshirani 2001). Firstly, we will give a short overview of a standard classification decision tree (DT) before moving on the methods used in this study.

Let us call the set of non-overlapping regions that are used to divide the feature space. The forms of those regions are high-dimensional rectangles - for simplicity and interpretability. The aim for DT would be to find the boxes that minimize the error term, which in the case of classification can be represented in several ways - misclassification error, Gini index or cross-entropy. However, it is very computationally taxing to consider every feasible partition of the feature space into boxes. Thus, the recursive binary splitting method is used, which is a top-down, greedy approach - it starts at the top of the tree and then it splits the feature space, making the best split possible, without looking forward. The algorithm is further described in the Appendix.

However, the beforementioned algorithm can sometimes lead to over-fitting and producing very ineffecient prediction results. Thus, the tree pruning technique is applied - first a large tree is grown, then it is “pruned” and a smaller version is obtain. The procedure leads to reduction in variance at the cost of some bias. Usually the cost complexity pruning algorithm is used in practice - a more thorough description is included in the Appendix.

The regular classification DT has high interpretability, but it sometimes lacks sufficient prediction power - they are often unstable and can be too sensitive to training data (Bhattacharyya et al. 2011). This leads us to variations of the classic classification DT.

#### Random Forests

#### Gradient Boosting Machines and Extreme Gradient Boosting Machines

## 3.3. Estimation

# 4. Data

## 4.1. Datasets

### 4.1.1. Real-World Datasets

#### UCSD-FICO Competition

#### Université Libre de Bruxelles

### 4.1.2. Simulated Datasets

#### PaySim

#### BankSim

## 4.2. Problems of Imbalanced Data and Data Sampling Techniques

### 4.2.1. Problem of Imbalanced Data

One of the biggest challenges faced in detecting fraudulent transactions is the one of unbalanced class sizes, with legitimate class outnumbering vastly the fraudulent one (Bhattacharyya et al. 2011). The application of data-sampling techniques has been widely used in the literature with various results when combined with different algorithms, as when such a problem occurs, it could hinder the model performances (Van Hulse, Khoshgoftaar, and Napolitano 2007). Moreover, in our particular case, the cost of missclassifying the minority class could prove to be a lot more costly than predicting wrongly the majority one.

### 4.2.2. Data Sampling Techniques

#### Random Oversampling (ROS)and Random UnderSampling (RUS)

The two techniques are the simplest and most common (Van Hulse, Khoshgoftaar, and Napolitano 2007). In minority oversampling (ROS), the observations from the minority group are randomly duplicated in order to balance the dataset. In majority undersampling (RUS), the aim is the same, but it is achieved by randomly removing observations of the majority class.

#### SMOTE

The Synthetic Minority Oversampling Technique (SMOTE), proposed by Chawla et al. (Chawla et al. 2002), artificial minority instances are created not simply through duplication, but rather with the extrapolation between preexisting observations. The technique starts by taking into account the k nearest neighbourhoods to a minority observation for every instance from that class. Then, the artificial observation are created, taking into account just a part of the nearest neighbours or all of them (with respect to the desired oversampling specification).

# 5. Results

# 6. Further Improvements

# 7. Conclusion

# 8. References

Akbani, Rehan, Stephen Kwek, and Nathalie Japkowicz. 2004. “Applying Support Vector Machines to Imbalanced Datasets.” In *European Conference on Machine Learning*, 39–50. Springer.

Batuwita, Rukshan, and Vasile Palade. 2013. “Class Imbalance Learning Methods for Support Vector Machines.”

Bhattacharyya, Siddhartha, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. 2011. “Data Mining for Credit Card Fraud: A Comparative Study.” *Decision Support Systems* 50 (3). Elsevier: 602–13.

Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. “SMOTE: Synthetic Minority over-Sampling Technique.” *Journal of Artificial Intelligence Research* 16: 321–57.

Cortes, Corinna, and Vladimir Vapnik. 1995. “Support-Vector Networks.” *Machine Learning* 20 (3). Springer: 273–97.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. Vol. 1. Springer series in statistics New York.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.

Ngai, EWT, Yong Hu, YH Wong, Yijun Chen, and Xin Sun. 2011. “The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and an Academic Review of Literature.” *Decision Support Systems* 50 (3). Elsevier: 559–69.

Van Hulse, Jason, Taghi M Khoshgoftaar, and Amri Napolitano. 2007. “Experimental Perspectives on Learning from Imbalanced Data.” In *Proceedings of the 24th International Conference on Machine Learning*, 935–42. ACM.

Veropoulos, Konstantinos, Colin Campbell, Nello Cristianini, and others. 1999. “Controlling the Sensitivity of Support Vector Machines.” In *Proceedings of the International Joint Conference on Ai*, 55:60.