A Comparative Study of Classification Algorithms for Credit Card Fraud Detection

February 2018

Abstract

When compared to other classification problems, financial transaction data for fraud detection have high dimensionality and are not amenable to visualization. Also, the class imbalance seems to be huge as the number of non-fraud transactions is much higher than fraudulent ones. Comparative studies are also difficult given privacy concerns and the scarcity of datasets. Using a publicly available dataset, we have experimented to discover the best approaches for fraud detection in financial transaction datasets. It is known that randomly sampling, without considering the underlying distribution causes bias. Hence, we used K-means clustering for getting the underlying distributions and undersampled accordingly, retaining the representation of non-fraud data. We have applied various model classes, like logistic regression, SVM, etc., and concluded that logistic regression gave best results with clustering, whereas SVM gave best results without clustering.

Keywords: machine learning, K-means, fraud detection, logistic regression, SVM, decision tree, random forest, MVG

1 Introduction

With the advancements in machine learning algorithms and artificial intelligence, many optimized algorithms have been constructed. Many of these algorithms behave best for a particular task and a set of data. One of the main issues in fraud detection is the lack of data. Also, the number of fraudulent transactions is very small as compared to non-fraudulent transactions. So these algorithms need to make use of whatever data are available, and deal with its skewed nature—the data can be highly imbalanced, skewed, and probably have a lot of hidden fields.

In this paper, we analyze several state of the art algorithms and modeling techniques on one real data set for fraud detection focusing on some open questions like: Which machine learning algorithm should be used? Should the data be analyzed in their original unbalanced form? If not, what is the best way to re-balance them? What are the benefits of using clustering before classification? These are just some of potential questions that arise during the design of a detection system. We do not claim to be able to give a definite answer to the problem, but we hope that our work serves as a guideline for other people in the field. Our goal is to show what worked and what did not work in a real case study. In this paper, we give a formalization of the learning problem in the context of credit card fraud detection.

If we randomly sample the data without taking into consideration the underlying distribution of the data, bias tends to be introduced [1]. Hence, it is important to maintain the structure of the original data. To do this we are using K-means clustering, and from those clusters, we pick random samples in the proportion of the cluster size. Undersampling done using this method does not lose the underlying distribution and hence no bias is introduced.

Our findings show that the best result was for classification on undersampled data using logistic regression after clustering using K-means technique. The accuracy, recall, precision and F1 score are all above 95% for the same. Without using clustering Support vector machine(SVM) worked best on randomly undersampled data with an accuracy of above 93%. Decision tree had accuracy of slightly above 92% with clustering and around 89% without clustering, while for random forest the same was around 94% and 92.5% respectively. The accuracy of Multivariate normal distribution or Multivariate Gaussian(MVG) was above 83% and 85% with and without clustering respectively.

In summary, this paper has the following salient features:

1. It addresses an important real-life problem for industries like banking, e-commerce, and insurance services, due to fraud. Experimentation is done on a real credit card transaction dataset to arrive at our conclusions and recommendations.

2. It offers a comparison of common classification techniques used in fraud detection—logistic regression, SVM, random forests, etc., on various hypothesis classes. Analysis is done using clustering before undersampling, and its effect on various models is studied.

3. It is shown that a technique of using clustering as a means for picking random samples followed by classification for fraud detection gives better results. It is thus recommended to be incorporated into industry practices, for handling skewed data.

2 Analysis and Preprocessing of Data

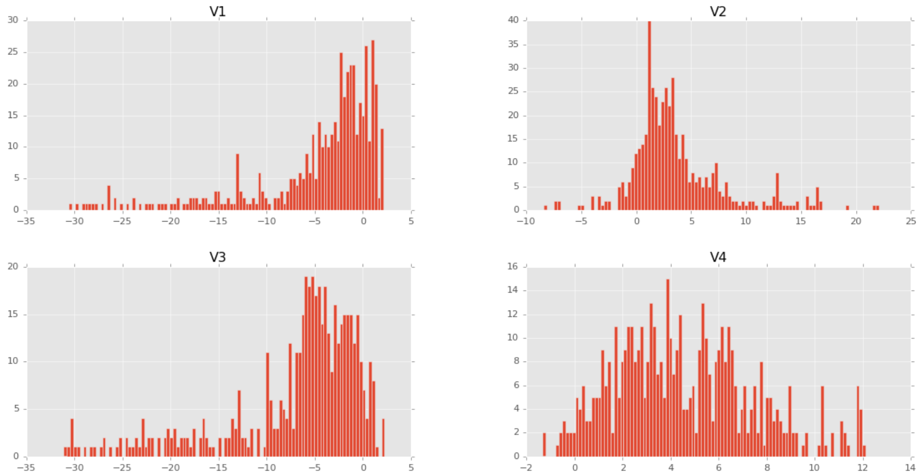
2.1 Dataset Used

The dataset is taken from Worldline and the Machine Learning Group of ULB (Université Libre de Bruxelles) on big data mining and fraud detection [2, 3], and features anonymized credit card transactions labelled as fraudulent or genuine.

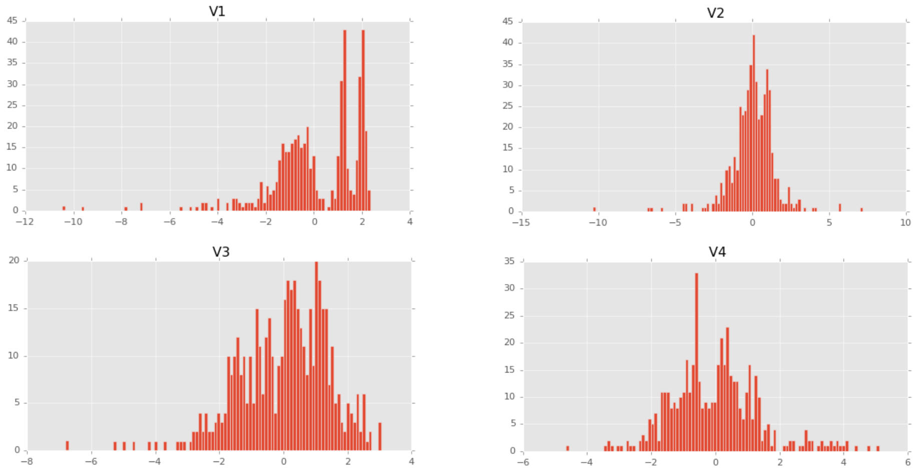
The dataset holds aggregated credit card transactions for September 2013 by European cardholders. This dataset presents transactions that happened over two days, where we have 492 frauds out of 284,807 transactions.

The dataset is thus exceedingly skewed, the positive class (frauds) representing just 0.172% of all transactions. As the transaction details contain confidential information, all the information cannot be provided, because of security issues. As a result, we are given 28 features indexed by numerical values obtained by Principal Component Analysis (PCA)[4] conversion from the actual fields in the original transaction records (whose values are confidential). The only features which have not been converted for PCA are ‘Time’ and ‘Amount’, while the feature ‘Class’ is the label which indicates whether the transaction is fraudulent or not.

The ‘Time’ feature indicates the seconds elapsed between the middle of transactions. The feature ‘Amount’ is the transaction amount, this feature might be utilized for cost-sensitive learning. The feature ‘Class’ is the reaction variable, taking the value 1 in the event of fraud, and 0 otherwise.



(a) Fraud Data: Distributions for Features V1 to V4



(b) Non-Fraud Data: Distributions for Features V1 to V4

Figure 1: Frauds and Non-Frauds: Plotting Histograms of Features V1 to V4

Analyzing the above histograms of fraud and non-fraud data, of various attributes, we can see that fraud and non-fraud data have similar distribution, and hence it is difficult to distinguish them based on just their distributions. Also, the variance in fraud data is comparatively higher than in non-fraud data. Here we have a plot of features V1, V2, V3, and V4. Other features V5-V28 also follow the same pattern. Hence we require more information on the data to directly or indirectly determine if a transaction is fraudulent or not.

2.1.1 Skewed Data

If we compare the number of fraud data vs number of non-fraud data, only 0.172% of the whole dataset pertains to fraudulent transactions. This is a clear example where using a typical accuracy score for evaluating our model class would not be sufficient. For example, considering fraud data as labelled 1 and non-fraud data as labelled 0, even if we classified all the data as non-fraudulent, we would arrive at a very high accuracy score. But that way we would be wrongly classifying all fraudulent data as non-fraudulent. Hence skewed data have to be handled differently. There are various pre-processing techniques[5, 6, 7] which are applied to skewed data, such as undersampling, oversampling, SMOTE, ADASYN[8].

2.1.2 Knowing the Underlying Distribution

As described by Dal Pozzolo *et al.* [1], random undersampling introduces a bias in the dataset. If sampling is done in a manner that accurately represents the population, then we may expect better results, in comparison to randomly undersampling the dataset.

Hence while undersampling the non-fraud data we have tried to preserve the underlying distribution, and then applying various machine learning models on this and comparing the result.

Once the clusters are formed, undersampling of the data is done from the clusters itself. Depending on the size of the clusters, the data points are picked from the clusters, corresponding to a similar ratio.

2.2 Verifying the Underlying Distribution

The total size of non-fraud data is 284315. After applying K-means clustering, the non-fraud data is distributed into 3 clusters.

	A	B	C
Cluster size	147471	3282	133562
Elements picked	255	5	231
Intra-cluster variance of sample	1.04615	11.6564	0.661371
Intra-cluster variance of population	1.07105	15.681	0.662193

Table 1: Analysis of clusters using K-means

From the variance, it can be seen that the dataset forms clusters, and the variances in clusters A and C are clearly low. Hence, the underlying distribution of the non-fraud data can be taken as reflected in these clusters. After undersampling is done, we apply the algorithms below and compare their results (comparison is done between randomly under-sampled data and data sampled using K-means clustering).

3 Algorithms, Results and Observations

We have kept our focus on the basic machine learning algorithms of: Logistic Regression, Support Vector Machine (SVM), Decision Tree Learning, Random Forest and Multivariate Gaussian (MVG).

Evaluation Metric Used: We use a "confusion matrix" (that has true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN)), and thereby calculate the accuracy, precision, recall and F1 score. A confusion matrix shows the number of correct and incorrect predictions made by the classification model compared to the actual outcomes (target value) in the data.

1. Accuracy (A): the ratio of the total number of predictions that were correct.

2. Precision (P): the ratio of positive cases that were correctly identified.

3. Recall (R): the ratio of actual positive cases which are correctly identified.

4. F1 score (F1): the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

Model	With K-means	Without K-means
Logistic Regression	A : 95.608 % R : 95.804 % P : 95.139 % F1 : 95.470 %	A : 91.891 % R : 93.571 % P : 89.726 % F1 : 91.608 %
SVM	A : 93.919 % R : 93.706 % P : 93.706 % F1 : 93.706 %	A : 93.243 % R : 94.285 % P : 91.667 % F1 : 92.957 %
Decision Tree	A : 92.229 % R : 88.811 % P : 94.776 % F1 : 91.696 %	A : 89.864 % R : 90.714 % P : 88.194 % F1 : 89.436 %
Random Forest	A : 93.919 % R : 99.300 % P : 89.308 % F1 : 94.039 %	A : 92.567 % R : 97.142 % P : 88.311 % F1 : 92.517 %
MVG	A : 83.923 % R : 100.0 % P : 56.521 % F1 : 72.222 %	A : 85.530 % R : 100.0 % P : 59.091 % F1 : 74.285 %

Table 2: Comparative Output with and without K-means

Observations: From Table 2, we can see that for all models except MVG, K-means applied on the data performs better than the results seen from data which does not have K-means applied to it. This is because using K-means, we are not just randomly picking up samples from the dataset, but are also maintaining the underlying distribution of the original data.

Multivariate Gaussian Distribution (MVG) behaves better on the dataset which did not have K-means; this is because randomly picking samples increases the probability of the sample being a Gaussian distribution, which is the most important requirement of the MVG algorithm. Hence, when we take the sample, it is possible that a little modification is done on the original data set and now the sample is more Gaussian than it was for the original dataset. In our dataset, some of our attributes do not follow the Gaussian distribution very well. Also, since our fraud and non-fraud distributions are highly similar, we had too many false positives. This is the reason why MVG did not behave as well as the other models.

We also observe that when precision is high, recall is comparatively low, and vice-versa. It is well known that precision and recall often have an inverse relationship, with one being high when the other is low [9]. This is because precision for a class is the number of true positives divided by the total number of elements labelled as belonging to the positive class, and recall is the number of true positives divided by the total number of elements that actually belong to the positive class.

Recommendations: According to our observations and tests on the dataset, we recommend the following as standard practices for fraud and non-fraud classification in financial transaction data:

1. Use logistic regression with K-means clustering, if the scale of computation is not an issue and the technique followed is undersampling using clustering.

2. If the data are not very skewed, and all the underlying distributions are Gaussian, then without sampling the dataset, we recommend using MVG along with a two-factor verification.

3. If clustering cannot be used, or if the dataset available is quite large (like for several months worth of transactions, or even more), then we suggest using SVM or random forest on randomly sampled data.

Also, since our data are highly unbalanced, we recommend the following options for overcoming the skewness of datasets:

4. Collect more data if possible, for better training.

5. Synthesize the data which is in the minority class.

6. Over-sample, i.e., add copies of the minority class.

7. Under-sample, i.e., select all instances of the minority class and an equal number of instances from the majority.

The above methods can yield a sample which has approximately 50-50% of the positive and the negative class instances.

4 Conclusion

From the above experiment and results, we can say that fraudulent data and non-fraudulent are highly similar in our dataset, and that randomly undersampling would fail to correctly represent the underlying distribution, and bias could be introduced in the under-sampled data. From the intra-cluster variance, it can be seen that the non-fraudulent data follows a clustering structure. And hence, when data are sampled according to their distribution, we obtain better results than randomly undersampling, in all cases. We conclude by saying that logistic regression gave best results for accuracy, precision and recall. However, depending on the dataset available, SVM without K-means can also be used. And if we are using two-factor verification, or if we have an attribute that highly distinguishes fraudulent and non-fraudulent transactions, then MVG too proves to give good results in detecting fraudulent transactions.

References

[1] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *6th IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2015)*, Cape Town, South Africa, Dec. 2015.

[2] ——, "Calibrating probability with undersampling for unbalanced classification," Dec. 2015.

[3] "Credit card data in r," 2015. [Online]. Available: http://www.ulb.ac.be/di/map/adalpozz/data/creditcard.Rdata

[4] J. I.T, *Principal Component Analysis, Series*, 2nd ed. Springer Series in Statistics, 2002.

[5] Rahman, M. Davis, and D.N., "Addressing the class imbalance problem in medical datasets," *International Journal of Machine Learning and Computing*, vol. 3, no. 2, p. 224–228, 2013.

[6] Chawla and N. V., *Data Mining for Imbalanced Datasets: An Overview*, 2010.

[7] M. Oded and R. Lior, *Data Mining and Knowledge Discovery Handbook*. Springer Series in Statistics.

[8] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," *IEEE Xplore*, 2008.

[9] B. Michael and F. Gey, "The relationship between recall and precision," *Journal of the American Society for Information Science*, vol. 45, no. 1, 1994.