Comparative study of classification algorithms for fraud detection with and without clustering

November 2017

Abstract

Fraud is a serious crime and fraud detection is a complex task with large real-world ramifications. Machine learning and classification algorithms have some significance here. However, 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 number of non-fraud transactions are much higher than fraudulent ones. Comparative studies are also difficult given the lack of publicly available datasets. Using a dataset from Kaggle.com, 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 class, 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, distribution of data

1 Introduction

Frauds are one of the major causes for loss of billions of dollars of money each year. The PwC global economic crime survey of 2016 suggests that more than one in three (36%) of organizations experienced economic crime[1]. Fraud detection is, given a set of credit card transactions, the process of identifying if a new authorized transaction belongs to the class of fraudulent or genuine transactions. 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 tasks and a set of data. One of the main issues in fraud detection is the lack of data. The number of fraud is infinitesimal as compared to non-fraud transactions. So these algorithms need to make use of whatever data is available. The data can be highly imbalanced, skewed, with probably a lot of hidden fields.

In this paper we try to analyze an experimental comparison of 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, which is the best way to re-balance them? Benefits of using clustering before classification? These are just some of potential questions that could raise 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 to that our work serves as guideline for other people in the field. Our goal is to show what worked and what did not 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[2]. 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 shows 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 were 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.

Our recommendations according to our findings are: Using logistic regression after clustering if computation is not an issue and the data is undersampled; If a two factor verification is already implemented and the features are gaussian we recommend using MVG with or without clustering; If clustering is not a feasible technique then SVM or random forest can be used by simply randomly sampling the data.

The remainder of this paper is structured as follows. Section 2 has related works previously implemented. Section 3 gives us a brief introduction about our data set and its underlying while in section 3.2 we have our findings on K-means. Section 3.3 contains about the algorithms we have used and their results. Results and observations are in section 4 where we have the evaluation metric used, the tabulated result and our observations. Also, in section 4 we suggest some recommended ways how to work with respect to different aspects. The paper concludes with a summary of our research findings in section 5.

2 Related Work

Several papers have been published which deal with specific algorithms and their modifications for various kinds of frauds. For instance, Chan *et al.* [3]

have proposed methods of combining multiple learned fraud detectors under a "cost model". The result was that we can significantly reduce loss due to fraud through distributed data mining of fraud models.

Du Jardin [4] evaluates the prediction accuracy of models designed using different classification methods.

Fanning and Cogger used an alternative approach using Artificial Neural Networks (ANNs) for detection of management fraud. Using publicly available predictors of fraudulent financial statements, they found a model of eight variables with a high probability of detection[5, 6].

Andrea Dal in his paper paper *et at.* [2], explained how biases have been introduced when data is randomly undersampled.

3 Fraud Detection

3.1 Data-set Introduction

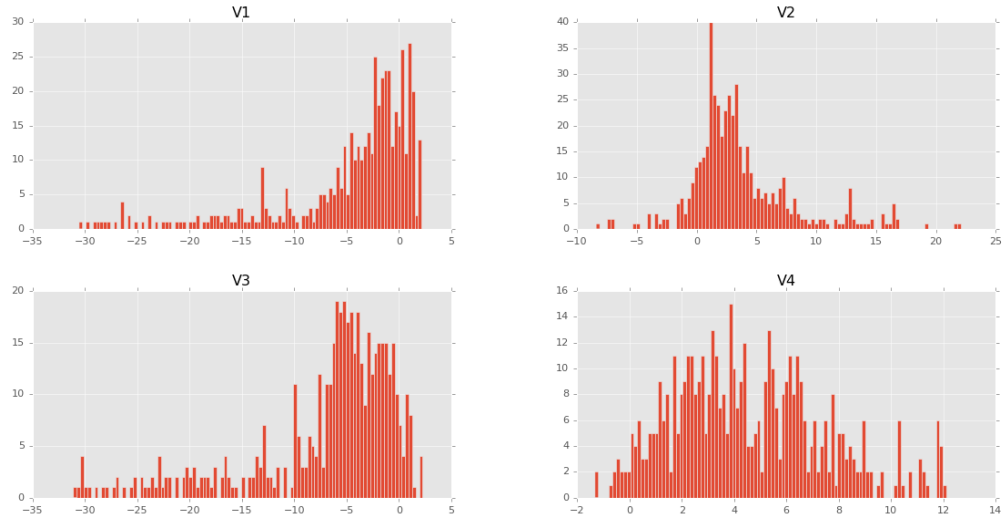
The data-set is taken from Worldline and the Machine Learning Group of ULB (Université Libre de Bruxelles) on big data mining and fraud detection [7][8] (Anonymous credit card transactions labelled as fraudulent or genuine). The data-set has been collected and analyzed during a research collaboration of Worldline and the Machine Learning Group

`(http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles)`

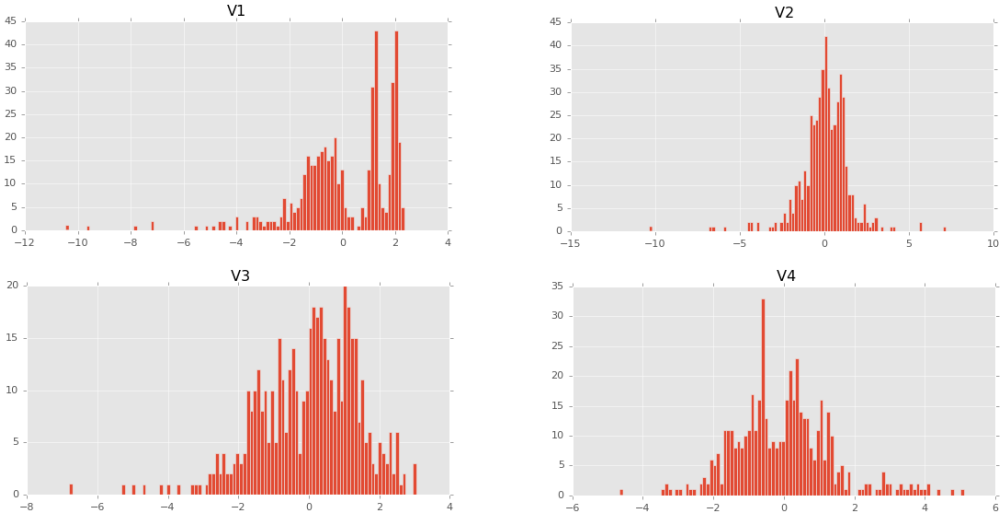
on big data mining and fraud detection. More details on current and past projects on related topics are available on

`http://mlg.ulb.ac.be/BruFence and http://mlg.ulb.ac.be/ARTML`

The data-sets hold transactions aggravated towards credit cards for September 2013 by European cardholders. This data-set presents exchanges that happened in two days, where we have 492 fakes out of 284,807 exchanges. The data-set is exceedingly unequal, the positive class (fakes) represent 0.172% of all exchanges. As the transaction details contains confidential information, all the information cannot be provided to us, because of security issues. As a result, we have 28 features, it holds main numerical enter variables which are the result of a PCA conversion. Unfortunately, because of secrecy issues, we can't be provided with the unique characteristics. The main features which have not been converted for PCA are 'Time' and 'Amount'. Characteristic 'Time' holds those seconds slipped by the middle of each transaction and the Initially transaction in the data-set. The characteristic 'Amount' is the transaction Amount, this characteristic might be utilized for example-dependant cost-sensitive learning. Characteristic 'Class' will be the reaction variable Furthermore it takes quality 1 in the event for fraud otherwise 0.



(a) Fraud data



(b) Non-fraud data

Figure 1: Fraud and Non-Fraud attribute analysis- Plotting Histogram of features V1 to V4

Analyzing the above plot of fraud and non fraud data, of various attributes, we can see that, fraud and non fraud are similar, and hence it is difficult to distinguish based on just their distribution. Also the variance in fraud data is comparatively higher than in non-fraud data. Here we have 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 which directly or indirectly determine if a transaction is fraud or not.

3.1.1 Skewed Data

If we compare number of fraud data vs number of non-fraud data only 0.172% is fraud transaction.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 labeled 1 and non-fraud data as labeled 0. Even i we classified all the data as non-fraud, we would come at a very high accuracy score. but that way we would be wrongly classifying all fraud data as non-fraud. Hence skewed data has to be handled differently. There are various techniques which are applied on skewed data. Which are illustrated below:

Techniques applied at data level

These are kind of pre-processing techniques, which are applied before trying to fit the data-set in any hypothesis class.
1. Under Sampling: Removing the majority class at random Data may be redundant but removal of data randomly is unsupervised and involves risk of removing relevant observations
2. Oversampling: Involves duplicating the minority class in random fashion. Risk of over-fitting the minority class and increases the training time.
3. SMOTE: It is combination of oversampling and under-sampling. etc.

Techniques applied at Algorithmic level

1. MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data etc.[9]

3.1.2 Knowing the Underlying Distribution

As described by Andrea Dal Pozzolo in his paper [**?**] that randomly under sampling introduces a bias in the data-set. And we lose the underlying distribution of the data-set, when it is randomly sampled. If sampling is done, such that it accurately represents the population, then we are expected to get better results, in comparison to randomly under sampling the data-set. Hence while under sampling 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. We have used kmeans clustering to cluster the non-fraud data. It was observed that the best results were at forming 3 clusters. Assuming that the underlying distribution is clusters. (Later we will verify our assumption that the underlying distribution is cluster using inter cluster and intra-cluster co-variance).

Once the clusters are formed, under sampling of the data is done from the clusters itself. Depending on the size of clusters, the data points are picked from the clusters, corresponding to similar ratio.
e.g. cluster 1 contains 35% of data
cluster 2 contains 60% of data and
cluster 3 contains 5% of data

so, while under sampling we will pick from the clusters in same ratio. if we have to choose 100 records for under sampling
35 data points are chosen from cluster 1
60 data points are chosen from cluster 2
and 5 data points are chosen from cluster 3
the above selection is done randomly, now once the data is under sampled properly, (based on underlying distribution) we apply various methods and use different hypothesis classes to train the model, and compare results. Randomly under sampling and sampling based on some underlying distribution gives different results, this can be seen in the below comparisons.

3.2 Verifying Kmeans as underlying distribution

Total Size of Non Fraud Data is 284315. After apllying K-means clustering the non fraud data is distributed into 3 clusters. The complete code is done in python using python supported library cv2

	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

Clearly from the variance it can be seen that data-set forms clusters as low intra-cluster variance means the clusters are closely packed and vice verse. And variance in cluster A and cluster C are clearly low. Hence the underlying distribution of the non-fraud data can be taken as clusters. After under-sampling is

done, we apply various standard algorithms and compare the results of the algorithm. (comparison is done between randomly under-sampled data and data sampled using kmeans clustering

3.3 Algorithm and its results

Logistic Regression: Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. It is an estimation of the logit function. The logit of a number P between 0 and 1 is given by the formula[10]:

$$logit(P) = ln\frac{P}{1-P}$$

Support Vector Machine(SVM): An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in[11][12].

Decision Tree learning: A decision tree is a decision support tool[13] that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). The goal is to create a model that predicts the value of a target variable based on several input variables[13].

Random Forest: Random Forest is a method of generating multiple decision trees(by default we have used 500 decision trees for one forest). Random decision forests correct for decision trees' habit of overfitting to their training set[14]. In Random Forests the idea is to decorrelate the several trees which are generated on the different bootstrapped samples from training Data. And then we reduce the Variance in the Trees by averaging them[15], this in turn reduces overfitting.

Multivariate Normal Distribution: It is the generalization of the one-dimensional (univariate) normal distribution to higher dimensions. It is the distribution of vectors of correlated variables each element of which has a univariate normal distribution. The underlying distribution should follow gaussian distribution[16]. It works in two parts, firstly, it finds the covariance matrix then the probability density function. Based on these two factors, it then classifies if an element is anomalous or not.

4 Results and Observation

Evaluation metric used: We will be using "confusion matrix" and from that we will 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 : the proportion of the total number of predictions that were correct.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

2. Precision : the proportion of positive cases that were correctly identified.

$$Precision = \frac{TP}{TP + FP}$$

3. Recall : the proportion of actual positive cases which are correctly identified.

$$Recall = \frac{TP}{TP + FN}$$

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

$$F1Score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

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 the above table, we can see that for all models except MVG, K-means applied on the data performs better than the 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 data set, but are also maintaining the underlying distribution of the original data.

Multivariate Gaussian Distribution(MVG) behaved better on the data set 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 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 original data set. In our data set, we some of our attributes did not follow gaussian very well, also, since our fraud and non fraud was highly similar, we had too many false positives. This is the

reason why MVG did not behave as good as the other models.

We also observe that when precision is high, recall is comparatively low, and vise versa. As noted from Wikipedia "Often, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other." This is because "precision for a class is the number of true positives divided by the total number of elements labeled 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 observation and the data set we used, we recommend the following:
1. Using "Logistic Regression with K-means" if computation is not an issue and the technique followed is undersampling using clustering.
2. If the data is not very skewed, and all the underlying distribution 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 data set available is quite large like for a few months, then we suggest using "SVM or random forest" on a randomly sampled data.
Also, since our data is highly unbalanced, we recommend the following options for overcoming the skewness of data set: Collect more data Synthesizing the data which is in minority Over-sampling, ie adding copies of the minority class Under-sampling, select all instance of the minority class and equal number of instances from the majority. The above methods will give us a sample which has approximately 50-50% of the positive and the negative class instances.

5 Conclusion

From the above experimentation and results we can say that, fraudulent data and non-fraudulent follows different distribution. Also randomly under-sampling, would fail to correctly represent the underlying distribution, and bias could be introduced in the under-sampled data. From the intra-variance between clusters, it can be seen that the non-fraudulent data follows a clustering structure. And hence when data are sampled according to its distribution, it gives better results than randomly under-sampling, in all cases. Also logistic regression have proved to be a best hypothesis for this data-set. As for logistic regression, accuracy, precision, recall and F1 score all are highest.

References

[1] Pwc.com, "Global economic crime survey," Mar. 2016. [Online]. Available: https://www.pwc.com/gx/en/services/advisory/forensics/economic-crime-survey.html

[2] 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.

[3] P. K. Chan, W. Fan, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems*, vol. 14, pp. 67–74, 2010.

[4] P. du Jardin, "Predicting bankruptcy using neural networks and other classification methods," *Neurocomputing*, vol. 73, no. 10–12, pp. 2047–2060, 2010.

[5] K. Fanning, K. Cogger, and R. Srivastava, "Detection of management fraud: a neural network approach," *International Journal of Intelligent Systems in Accounting, Finance Management*, vol. 4, no. 2, pp. 113–26, 1995.

[6] ——, "Neural network detection of management fraud using published financial data," *International Journal of Intelligent Systems in Accounting, Finance Management*, vol. 7, no. 1, pp. 21–24, 1998.

[7] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," Dec. 2015.

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

[9] Y. He, H. Tan, W. Luo, S. Feng, and J. Fan, "Mr-dbscan: a scalable mapreduce-based dbscan algorithm for heavily skewed data," *Springer.*, vol. 8, p. 83–99, Feb. 2014.

[10] J. S. Cramer, *The origins and development of the logit model.* Cambridge University Press, 2003.

[11] Cristianini, Nello;, and S.-T. John, *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[12] S. Ray, "Understanding support vector machine algorithm," Sep. 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code

[13] Rokach, Lior, and O. Maimon, *Data mining with decision trees: theory and applications.* World Scientific Pub Co Inc., 2008.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer., 2008.

[15] datascienceplus, "Random forests introduction," Jul. 2017. [Online]. Available: https://datascienceplus.com/random-forests-in-r/

[16] Mathworks, "Multivariate normal distribution." [Online]. Available: https://in.mathworks.com/help/stats/multivariate-normal-distribution.html