Application of PCA in Credit Card Fraud Detection

Huang Ke Tao
1024040812
Nanjing University of Posts and Telecommunications
School of Computer Science
NanJing China

Abstract

PCA dimensionality reduction is a commonly used technique, which can effectively reduce the size of data sets and speed up training. This paper mainly explores the impact of PCA on the training speed and evaluation indicators of credit card fraud data sets on different models.The results show that PCA dimensionality reduction has a certain promotion effect on different models.

1 Introduction

Credit card fraud detection is an essential aspect of modern financial systems, as fraudulent activities not only cause significant financial losses but also undermine the trust in digital payment systems. Over the years, fraud detection systems have become more sophisticated, moving away from simple rule-based methods to more complex machine learning models. However, one of the major challenges in this domain is the high-dimensionality of the transaction data, which can lead to issues such as overfitting and computational inefficiencies[1]. With the ever-increasing volume of credit card transactions and more complex fraudulent schemes, relying solely on traditional methods is no longer adequate to capture the underlying patterns of fraud.

Principal Component Analysis (PCA), a technique used for dimensionality reduction, has proven to be effective in addressing these challenges. By transforming the high-dimensional data into a lower-dimensional space while retaining the maximum variance, PCA simplifies the complexity of the data, making it easier for machine learning algorithms to identify fraudulent patterns[2]. Moreover, PCA reduces the risk of overfitting by eliminating noisy and redundant features, which can enhance the generalization capability of fraud detection models. This reduction in dimensionality leads to improvements in both the speed and accuracy of models, allowing for real-time detection of fraud with minimal computational overhead.

In this paper, we investigate the application of PCA in credit card fraud detection, focusing on its role in enhancing the performance of machine learning models. Specifically, we demonstrate how PCA can be applied to reduce the dimensionality of transaction data and improve the efficiency of fraud detection systems. We compare the performance of models with and without PCA preprocessing to assess its impact on detection precision and recall, highlighting the advantages of integrating PCA in fraud detection workflows. The reminder of this paper is structured as follows.

2 Related work

In the past, the initial strategies for detecting credit card fraud placed significant emphasis on rule-based mechanisms and statistical techniques. These approaches employed preset limits to identify transactions as potentially fraudulent according to specific characteristics, like the sum of the transaction, the place where it occurred, and how often similar transactions took place. Although these traditional means were easy to put into practice, they lacked the ability to adapt and frequently couldn't keep up with the intricate and constantly changing aspects of fraud patterns. Consequently, these former methods were progressively superseded by more sophisticated machine learning algorithms.

Emmanuel[3] et al., developed a feature selection algorithm that was based on the genetic algorithm (GA). Within the fitness function of this algorithm, the RF method was utilized. This particular algorithm was then applied to a fabricated credit card fraud data set. The outcome of this implementation was an Area Under the Curve (AUC) value reaching 1, accompanied by a 100

Khatri[[4] et al. conducted a performance analysis on ML (Machine Learning) techniques used for credit card fraud detection. In this study, the authors took the following ML methods into consideration: Decision Tree (DT), k-Nearest Neighbors (KNN), Logistic Regression (LR), Random Forest (RF), and Naive Bayes (NB). To evaluate the performance of each ML method, the authors utilized a highly imbalanced data set generated from European cardholders. One of the main performance metrics employed in the experiment was the precision obtained by each classifier. The experimental results demonstrated that the precisions of DT, KNN, LR, RF, and NB were 85.11

Benchaji[5] et al, combines the Long Short Term Memory (LSTM) networks, the uniform manifold approximation and projection (UMAP), and the attention mechanism to enable the combined model to both focus on local information in the sample and capture long-term dependencies in the transaction sequence.Their model was tested on a data set called Dataset-1 and eventually achieved a precision of 0.9885 and a recall of 0.9191.However, they use a recursion network when processing sequences, which results in high memory overhead.

Fiore[6] et al, Fiore used a generative adversarial network to solve the sample imbalance of the credit card fraud data set. Specifically, they trained a GAN to output simulated minority class examples, and then merged it with the training data into an enhanced training set in order to improve the effectiveness of the classifier. Their experiment was on a publicly available data set. Compared to the Smote method, the GAN can effectively improve the classification sensitivity of the data set, but the specificity is slightly reduced.

3 Methods and materials

3.1 Dataset

Many banks are reluctant to disclose the data set related to credit cards. The data set of this experiment is sourced from the Kaggle website[7].

3.2 Data Mining

Before officially starting data handling, you first need to perform descriptive statistics on the original data source in order to better understand the situation of the data and prepare for the next data handling.This often leads to unexpected gains about the original data source.

The shape of this data is (284807, 31). The first number represents the number of samples, and the second number represents the dimension. Each sample represents a credit card transaction record. In terms of dimensions, the first dimension is timestamp (e.g. 0, 1, 2). Dimensions 2-29 are the main features of the credit card data, but it is unknown what each feature represents. The 30th dimension represents the amount involved in each credit card transaction sample. The last dimension is the category of the credit card sample, where 1 represents a fraudulent transaction and 0 represents a non-fraudulent transaction.

All samples have no missing values. After counting the proportion of positive and negative samples, it is found that the number of samples belonging to the fraud category is only 492, accounting for 0.17

The time of the transaction should be a factor worth exploring. Therefore, the experiment provides a visual description of the time of normal and fraudulent transactions. Since there may be more frequencies on a timestamp, the frequencies are logarithmic. The final result is shown in Fig.1. The upper half represents the normal transaction credit card data set sample, and the lower half represents the fraudulent transaction sample. It can be seen that the normal transaction time has a certain periodicity, and they tend to be more concentrated in two time periods. The fraudulent credit card sample tends to be more uniform distribution and does not show strong periodicity.This is a very interesting mining of information from the original data source.

Secondly, I continue to analyze the relationship between the amount of transactions and the label. A boxplot is a statistical chart used to show the distribution of data, which is concise and intuitive. Therefore, I used it to plot the relationship between the two transaction categories and the amount. The result is shown in Fig.2. It can be inferred that although both transaction categories are densely distributed at the lower amount, the amount of fraudulent transactions is generally relatively small,

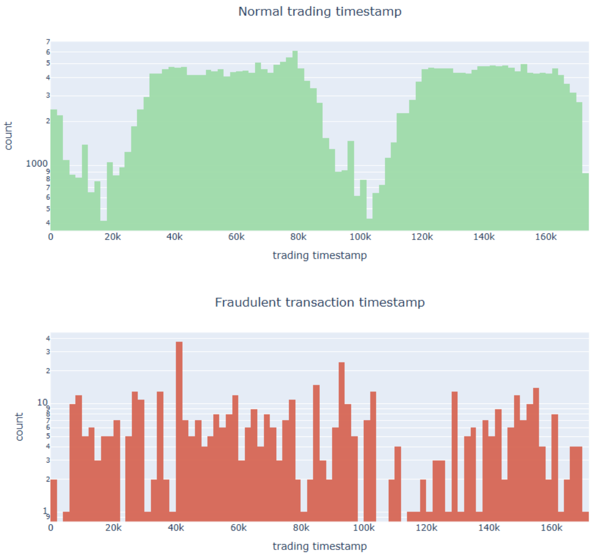

Figure 1: The relationship between the time of the transaction and the label

with a normal maximum of more than 2,000, while the amount of normal transactions is generally larger.

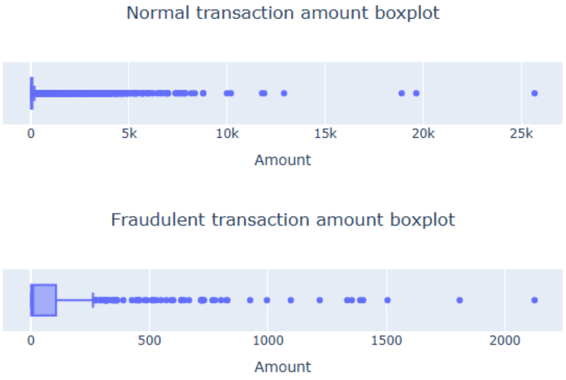

Figure 2: The relationship between the amount of the transaction and the label

Finally, I drew a scatter plot to explore the relationship between transaction amounts and time. As mentioned above, fraudulent transactions tend to be non-cyclical, fragmented, and involve low amounts of money. Therefore, as shown in Fig.3, it is obvious that the lowest level of data points has the largest number and is evenly distributed. The normal transaction sample, due to the large number of samples, actually does not see any obvious distribution in this graph, but the number of outliers is very small, which is obvious.

3.3 Feature dimensionality reduction

Feature dimensionality reduction refers to the process of transforming the original high-dimensional feature space into a low-dimensional feature space through some mathematical transformation or mapping method, while retaining the important information and structure in the original data source as much as possible. It can reduce the amount of computation and storage space, remove noise and redundant information, and reduce over-fitting.

The credit card data set used in the experiment has 30 features, leaving 28 main features in addition to timestamps and amounts. These features are not described in the original website, but the distribution of each feature must be closely related to the final label. Therefore, I plotted the

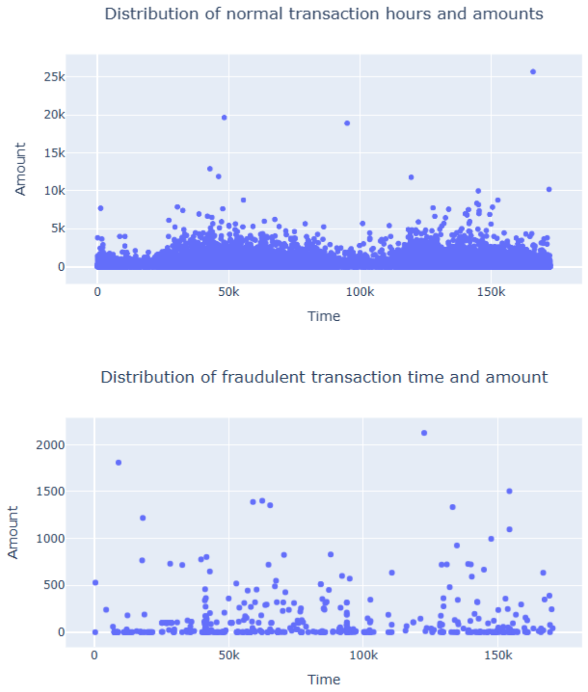

Figure 3: Relationship between sample timestamp and amount

distribution of each feature under different labels in order to visually compare the distribution of each feature in different labels. The plotted histogram will have two elements, one is the specific value range of the corresponding feature (that is, the abscissa coordinate), and the other is the frequency of the feature value in the corresponding category (that is, the longitudinal coordinate).Then it is conceivable that the final rendered histogram can be classified into four categories, that is, under this feature, the value range of positive and negative samples is similar (or not), and the frequency of each sub-value range is similar (or not).

The actual plotted result is also roughly in line with the above guess. As shown in Fig.4, it is the histogram of features V1 and V28. Under these two features, the distribution of positive and negative samples is roughly similar, but the V1 feature can still distinguish between positive and negative samples in terms of frequency, while the performance of the V28 feature in this regard may be slightly weaker.

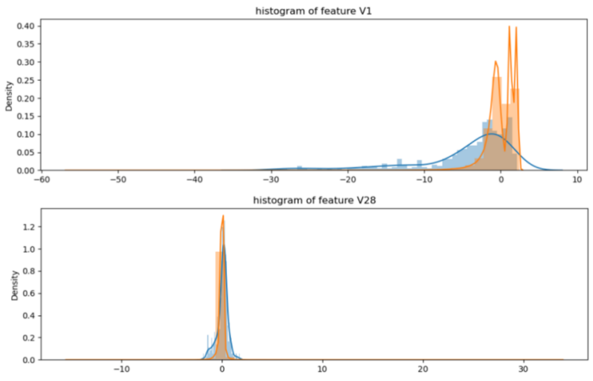

Figure 4: V1 and V28 feature histograms

Correspondingly, there are also features with different interval distributions but roughly the same frequency. As shown in Fig.5, feature V11 and V12 are in line with this situation. Under these two features, the interval distribution of positive and negative samples is different, but in terms of frequency, the difference between positive and negative samples of feature V11 is significantly lower than that of V12.

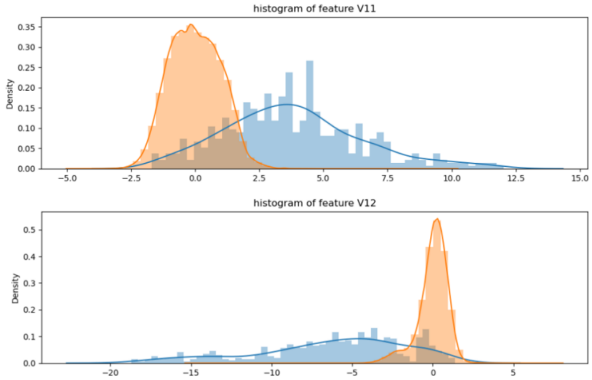

Figure 5: V11 and V12 feature histograms

If under a feature, the data interval distribution and frequency of the positive and negative samples are very similar, then it can be considered that the feature is extremely unhelpful to distinguish between positive and negative samples, so it can be deleted. This is the first dimensionality reduction operation, and a total of 6 features are deleted, which are V13, V15, V22, V24, V25, V28. Next, perform the PCA dimensionality reduction operation.

Unlike traditional PCA methods, there are currently many variants of PCA, such as Kernel PCA[8], Sparse PCA[9], Incremental PCA[10], and Autoencoder-based PCA[11]. Sparse PCA and Incremental PCA are suitable for high-dimensional data sets and large-scale data sets, respectively, while Autoencoder-based PCA is commonly used for unsupervised machine learning. I use Kernel PCA as my main method for the next step to feature dimensionality reduction.

Kernel PCA is a non-linear principal component analysis method, which introduces kernel function to map the original data source to the high-dimensional feature space, and then performs principal component analysis in the high-dimensional feature space to achieve non-linear dimensionality reduction of the original data source[8].

The kernel function I chose is the radial basis function, and set gamma = 15, n_components = 15, that is, a total of 15 principal component features are retained. After dimensionality reduction, in order to verify whether the principal components can effectively distinguish the labels of the sample, I used 2 of the principal components as the basis to draw the scatter plot. As shown in Fig.6. I selected the 2nd and 5th principal components, and selected 200 fraudulent samples and 5000 non-fraudulent samples as the sample for the plot. The fraudulent label is a red dot, while the non-fraudulent label is blue. It can be seen that the non-fraudulent labels are distributed in the left half of the graph, and there is a certain distribution law, while the fraudulent labels are distributed in the right half of the graph, and they are far from each other. This shows that it is easier to distinguish between positive and negative samples on these two principal components.

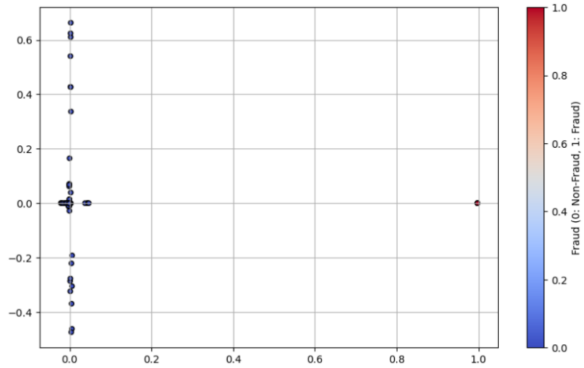

Figure 6: Sample performance on principal components 2 and 5

3.4 Sample imbalance processing

As mentioned in 3.2, the data set is highly unbalanced between positive and negative samples. From the perspective of the data itself, common sample imbalance strategies include oversampling, undersampling, and comprehensive sampling[12]. From the perspective of the model itself, such as in the deep learning domain, there are corrections for the loss function, such as the very famous focal loss[13]. In this experiment, I first perform SMOTETomek on raw datasets to obtain the new data set B. This data set is also the basis for modeling and decision-making in the next chapter 4.

But in order to try the effect of focal loss, I used the xgboost model with focal loss to make decisions on the original data source after feature engineering, and compared it with the xgboost model that did not use focal loss but was trained with the data after SMOTETomek sampling.

The sample imbalance strategy is for the training set, so first, I divide the original training set and test set in a ratio of 7:3. After performing the SMOTETomek operation on the training set, the positive and negative sample ratio of the data is 1:1, and the total number of samples reaches 398040.

4 Experiments

As mentioned in the 3.4, we divide the training set and test set into 7:3. There are a total of 3 experiments conducted. The first experiment is mainly to compare the prediction effect of different models on the data after using kernal pca dimensionality reduction. The models used are xgboost, decision tree, svm and BP neural network. The evaluation metrics used are accuracy and recall. where precision was the number of true positives (TP) divided by the number of all positive results, i.e., true positives (TP) plus false positives (FP); while recovery was the number of true positives (TP) divided by the number of all tests that should have been positive, that is, true positives (TP) plus false negatives (FN). The second experiment is to compare the training time and prediction results of the models before and after using Kernal PCA. The third experiment is to verify the effect of focal loss, and the model used is xgboost. The following equations relate to the classification functions previously described.

Table. 1 shows the evaluation metrics between different models. Since precision and recall are often more important for credit card fraud problems, these two metrics are used. Overall, precision is higher than recall, with the highest precision being the decision tree, but recall is also the lowest. Overall, I think xgboost works best.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

	Precision	Recall
Xgboost	0.85	0.73
SVM	0.79	0.69
BP neural network	0.83	0.70
Decision tree	0.87	0.62

Table 1: Evaluation metrics for different models

As shown in Table. 2, the training time of each model before and after using Kernal PCA is shown. Among them, dimensionality reduction is not obvious for the improvement of SVM training speed, and it still has a little improvement for the rest of the models. In addition, it can be seen that decision trees and SVMs are trained faster, while neural network training is slower, which is very common.

The evaluation index can be seen from Table. 3. Before and after dimensionality reduction, The precision rate and recall rate of SVM have almost no significant change, but the effect of decision tree is very significant, indicating that Kernal PCA has a great influence on it. For the remaining two models, it can be seen that Kernal PCA also has a certain promotion effect on them, but the improvement in effect is not very obvious.

	Time(Before)	Time(After)
Xgboost	37.1(s)	22.9(s)
SVM	15.8(s)	14.3(s)
BP neural network	396(s)	284(s)
Decision tree	12.2(s)	9.7(s)

Table 2: Running Time for different models before and after Kernal PCA

	Precision(Before)	Recall(Before)
Xgboost	0.81	0.72
SVM	0.80	0.69
BP neural network	0.80	0.65
Decision tree	0.78	0.59

Table 3: Evaluation metrics for different models After Kernal PCA

5 Conclusion

This experiment mainly explores the performance of Kernal PCA on the credit card fraud data set. The experiment mainly uses four models: BP neural network, SVM, decision tree, and xgboost. The final experimental results show that Kernal PCA has a certain degree of improvement on the training speed and evaluation indicators of the four models, but the effect on different models is still different. The experiment also tries the performance of focal loss on the imbalanced dataset set. The experimental results show that on the xgboost model, the recall performance of this data set improves greatly.

References

[1] Phua C .A Comprehensive Survey of Data Mining-based Fraud Detection Research[J].Artificial Intelligence Review, 2010, abs/1009.6119.DOI:10.1016/j.chb.2012.01.002.

[2] Jolliffe, I,Jolliffe, N.A.Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed[J]. 2002.

[3] Ileberi E , Sun Y , Wang Z .A machine learning based credit card fraud detection using the GA algorithm for feature selection[J].Journal of Big Data, 2022, 9(1):1-17.DOI:10.1186/s40537-022-00573-8.

[4] Khatri S , Arora A , Agrawal A P .Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison[C]//2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence).IEEE, 2020.DOI:10.1109/Confluence47617.2020.9057851.

[5] Benchaji I , Douzi S , El Ouahidi B ,et al.Enhanced credit card fraud detection based on attention mechanism and LSTM deep model[J].Journal of Big Data, 2021, 8(1):1-21.DOI:10.1186/s40537-021-00541-8.

[6] Fiore U , Santis A D , Perla F ,et al.Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection[J].Information ences, 2017:448-455.DOI:10.1016/j.ins.2017.12.030.

[7] The Credit card fraud [Online].https://www.kaggle.com/ mlg-ulb/creditcardfraud

[8] Schölkopf, Bernhard, Smola A ,Müller, KlausRobert.Kernel principal component analysis BT - Artificial Neural Networks-ICANN'97[J]. 1997.

[9] Zou,H.,Hastie,T.,& Tibshirani,R.(2006).Sparse Principal Component Analysis.Journal of Computational and Graphical Statistics,15(2), 265–286. https://doi.org/10.1198/106186006X113430

[10] Zhao H, Yuen P C,Kwok J T .A novel incremental principal component analysis and its application for face recognition[J].IEEE Transactions on Systems Man & Cybernetics Part B,2006,36(4):873-886.DOI:10.1109/TSMCB.2006.870645.

[11] Chen Z , Yeo C K , Lee B S ,et al.Autoencoder-based network anomaly detection[C]//2018 Wireless Telecommunications Symposium (WTS).2018.DOI:10.1109/WTS.2018.8363930.

[12] Krawczyk B .Learning from imbalanced data: Open challenges and future directions[J].Progress in Artificial Intelligence, 2016, 5(4).DOI:10.1007/s13748-016-0094-0.

[13] Lin T Y,Goyal P,Girshick R ,et al.Focal Loss for Dense Object Detection[J].IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP(99):2999-3007.DOI:10.1109/TPAMI.2018.2858826.