|  |
| --- |
| **Unsupervised Machine Learning for Risky Bank Clients Identification** |

**Bo Xiao Shuyang Li**

CCID: CCID:

[bxiao2@ualberta.ca](mailto:bxiao2@ualberta.ca) shuyangli@ualberta.ca

**Yuntian Zhang Han Wang**

CCID:CCID: han13

[yuntian1@ualberta.ca](mailto:yuntian1@ualberta.ca)han13@ualberta.ca

**Abstract**

Banking operations are complicated and include a lot of periodic activities and transactions performed by employees and customers. The monitoring of banking operations or historical records can avoid banks affected by risky events performed by customers. Anomaly detection problem has been more and more popular in the banking industry recently. Historical data have been collected and processed by machine learning algorithms. KNN has been used for calculating anomaly score based on their distance. And K-means groups 'k' similar clusters of data. In this project, We have proposed a four steps unsupervised methodology in this project, which are feature engineering, dimensionality reduction, model selection, and evaluation, with the purpose of categorizing customers into two groups: risky and non-risky. In the given dataset, there exist 40,000 customers and each customer has been described by 1276 features. Four unsupervised algorithms have been tested in this dataset, and compared based on two different evaluated metrics. The evaluation results showed the GMM-EM method has a better performance than others. Also, the number of distribution initialized affects the performance of the model. Based on the results, we suggest that GMM-EM should be used for further application and the number of components should be set up to three, which means customers should be grouped by three categories instead of two.

**1 Introduction**

Anomaly detection refers to the task of identifying those entries in a dataset that is different than the majority of the data. These unusual patterns are often referred to as anomalies, outliers noise, or exceptions. Anomaly detection is applicable in a variety of domains, such as intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, and detecting ecosystem disturbances [1]. In the US, the value of payment card fraud losses is as high as 9.1 billion. Naturally, anomaly detection customer behavior archetype can identify and significantly such risky behavior. There is a rich history that applied machine learning techniques to detect risky customers. K-nearest neighbor has been used for detecting an anomaly, which computes the anomaly score based on their distance [2]. However, this method is hard to select an appropriate threshold. Furthermore, K-means [3] is a widely used clustering algorithm. It groups 'k' similar clusters of data points. Data instances that fall outside of these groups are usually marked as anomalies.

There are several challenges raise. Firstly, the data contains noise which might be similar to abnormal behavior, because the boundary between normal and abnormal behavior is often not precise. For example, freezing a credit card after a risky behavior may prevent future money loss but too many false positive translates to terrible customer experience [4]. Secondly, the definition of abnormal or normal may frequently change, as fraud committed constantly adapt themselves for financial gains. History statistic provides little or no when a new fraud mechanism emerges. Thirdly, an interesting object is often not a rare object in the statistical definition. Many outlier detection methods (in particular unsupervised methods) will fail on such data. Instead, a cluster analysis algorithm may be able to detect the microclusters [5].

In this project, the main goal is to detect risky customers based on their historical behavior dataset, which consists of 40K entries of customers without true labels. The input for this task is 1276 features collected from historical behaviors of the individual customer, while the output is to indicate if this customer will do risky behaviors in the future or not. We have proposed a four steps unsupervised methodology in this project, which are feature engineering, dimensionality reduction, model selection, and evaluation. Missing data and normalization have been handled in the feature engineering. And we have applied PCA [6] to reduce the dimensionality of the dataset from 1276 to 100. Two widely adopted models, probabilistic model, and cluster model, has been tested on our dataset. The dataset has be split into a training set and test set with the proportion of 80% and 20% separately. 5-fold cross-validation [7] has been applied to the training set to select the best parameters for each model. We have decided to use the silhouette coefficient and log likelihood to evaluate cluster model and probabilistic model respectively. The trained model has been tested on the test set for the final comparison. GMM-EM algorithm [8] has shown the best performance of detecting bank anomaly in this project.

**2 Related work**

There is not much published research on using an unsupervised model to detect risky bank customers, but there is some work sharing a similar task or similar techniques with our project. We provide a short summary of some of this work.

**2.1 Related work sharing a similar task**

An unsupervised shell company identification model is proposed using only bank transactions [9]. All incoming and outgoing transactions from a particular bank account are used as its various attributes, and use anomaly detection techniques to identify the accounts that pertain to shell companies. Three unsupervised algorithms are compared, including RRS, a near linear time approximation algorithm (FastVOA) and Local Outlier Factor (LOF), the result shows the performance of the LOF algorithm clearly outperforms the other two. A developed bankruptcy Neural Network prediction model in reference [10] introduces some novel indicators extracted from the stock price of the firm to predict bankruptcy. In [11], it constructed the consumer loan default predicting model through conducting the empirical analysis on the customers of unsecured consumer loan from a certain financial institution in Taiwan, and adopt the borrower’s demographic variables and money attitude as the real-time discriminant information. The results show that DEA–DA and NN are possessed better-predicting capability and they are the optimal predicting model that this study longing for.

**2.2 Related work sharing similar techniques**

In [12], it proposes a supervised evaluation approach for bank loan default classification models based on multiple criteria decision making (MCDM) methods. A set of performance metrics is utilized to measure a selection of statistical and machine-learning default models. The technique for order preference by similarity to ideal solution (TOPSIS), an MCDM method, takes the performances of default classification models on multiple performance metrics as inputs to generate a ranking of default risk models. In [13], 19 different unsupervised anomaly detection algorithms are evaluated on 10 different datasets from multiple application domains. Additionally, this evaluation reveals the strengths and weaknesses of the different approaches for the first time. Besides the anomaly detection performance, computational effort, the impact of parameter settings as well as the global/local anomaly detection behavior is outlined. The reference [14] proposes a novel ensemble approach called SELECT for anomaly detection, which automatically and systematically selects the results from constituent detectors to combine in a fully unsupervised fashion. Gaussian mixture models are a probabilistic model for representing [normally distributed](https://brilliant.org/wiki/multivariate-normal-distribution/) subpopulations within an overall population [15]. [Mixture](https://brilliant.org/wiki/mixture-model/) models don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of [unsupervised learning](https://brilliant.org/wiki/unsupervised-learning/). If the number of components is known, [expectation maximization](https://brilliant.org/wiki/expectation-maximization-algorithm/) (EM) is the technique most commonly used to estimate the mixture model's parameters. Expectation maximization is an iterative algorithm and has the convenient property that the maximum likelihood of the data strictly increases with each subsequent iteration, meaning it is guaranteed to approach a [local maximum](https://brilliant.org/wiki/extrema/) or [saddle point](https://brilliant.org/wiki/saddle-point/). In [16], it presents a data clustering method named BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) and demonstrates that it is especially suitable for very large databases. BIRCH incrementally and dynamically clusters incoming multi-dimensional metric data points to try to produce the best quality clustering with the available resources (i.e., available memory and time constraints). BIRCH can typically find a good clustering with a single scan of the data, and improve the quality further with a few additional scans.   

**3. Methodology**

In this section, the methodology proposed in this project will be introduced in details. In order to group customers into two categories, risky and non-risky, based on our non-labeled dataset, a four-steps methodology has been proposed, as shown in Figure 1, which includes feature engineering, dimensionality reduction, model selection, and evaluation.

|  |
| --- |
|  |

Figure 1: The pipeline of the proposed methodology

**3.1 Feature Engineering**

The whole dataset contains about 40k customers each customer has 1276 features. It is noticed that there exist a lot of missing data (NaN) in the given dataset. In this step, features are filtered and refined. First of all, the NaN values in this dataset are filled with -2. Instead of replacing NaN with mean value in the column, we think missing data still have some insights. For example, NaN in the feature “AGE\_MOST\_RECENT\_DEROGATORY” column means that this customer never had a derogatory.  To emphasize the role of NaN in such columns, the NaN values are replaced by -2. Then, we have converted non-number, such as postcodes, entries to numbers. This is done by converting each character in the string to bytes which guarantees a unique mapping from string to number. At the last step, normalization is used to help further processes. The columns converted from non-number have large values because we use bytes to represent a character. So, it is hard to apply normalization directly. The formula (1) has been applied on the dataset to do with those columns, which have large values. Then, tanh function has been applied on the dataset, and the value of each entry is in the range of -1 to 1 after feature engineering.

**(1)**

**3.2 Dimensionality Reduction**

It is necessary to compress and filter noise of our data by reducing the feature dimension of our data because there are 1276 features. Dimensionality reduction will help to simplify our problem and improve the efficiency of our model. Principal component analysis (PCA) has been chosen in this project. Compared with other dimensionality reduction techniques, such as autoencoder, PCA is efficient in computation. It is a multivariate technique that analyzes a data table in which observations are described by several inter‐correlated quantitative dependent variables. Its goal is to extract the important information from the table, to represent it as a set of new orthogonal variables called principal components, and to display the pattern of similarity of the observations and of the variables as points in maps.

**3.3 Model Selection**

In this study, we are aiming to group the customers into two groups: risky and non-risky. This task can be defined as a clustering problem, which generates two clusters based on the given data. Two cluster models, K-means and Birch have been selected in this project. Considering there are 40,000 data points in our dataset, computational speed is important in this study. K-means and Birch are efficient and perform well in large dataset. However, the clustering model is a hard cluster method, which means each data sample will be categorized as 0 or 1, and we have only limited information from the clustering results. The probabilistic model assumes all data points are generated from some numbers of Gaussian distributions. The advantages of the probabilistic model are that it provides probability as output.

**3.4 Evaluation**

The dataset has been split into two subsets, which are training set and testing set. The training set contains 80% of data, while the testing set has 20%. 5-fold cross validation has been conducted in the training set to get the best parameters for each model. Then, the whole training set has been used to train the model with the setup parameters and then test on the testing data for the comparison of the results. It is hard to compare clustering methods and probabilistic methods under the same criteria. It is decided to use the log likelihood to evaluate probabilistic methods and silhouette coefficient to evaluate clustering methods.

**3.4.1 Log likelihood to evaluate probabilistic methods**

The probability of a data point x located in a probabilistic model is the sum of probability of x located in the kth component, which has been shown in the formula (2). The Likelihood function is a production of the probability of each instance located in this model as demonstrated in the formula (3). The score of this model is the average of  L over N instances and a larger L implies a higher overall probability that the instances are located in this model.

(2)

(3)

**3.4.2 Silhouette Coefficient to evaluate clustering model [17]**

Assume the data have been clustered into k clusters. For each data point  (data point me in

the cluster Ci), let

(4)

be the mean distance between a sample and all other points in the same class, where d(i,j) is the distance between data points i and j in the cluster Ci. We can interpret as a measure of how well point i is assigned to its cluster (the smaller the value, the better the assignment).

For each data point, we now define

(5)

to be the smallest average distance of I to all points in any other cluster. We now define a silhouette (value) of one data point i

(6)

And .

From the above definition it is clear that

(7)

The average s(i) overall points of a cluster is a measure of how tightly grouped all the points in the cluster are. Thus the average s(i) overall data of the entire dataset is a measure of how appropriately the data have been clustered, the average s(i) is closer to 1 the better the clustering result is.

|  |
| --- |
|  |

Figure 2: The interpretation of Silhouette Coefficient parameters

**4. Results and Discussion**

The overall performance can be found in Table 1 below. It can be found the GMM-EM performs better than Variational Bayesian GMM-EM. The log likelihood for GMM-EM is 30.69, which is slightly higher than 29.85. For clustering methods, K-means performs better than Birch, which has a silhouette coefficient of 0.63. To sum up, GMM-EM is recommended in the probabilistic category and K-means is recommended in the cluster category.

Table 1. The overall performance in the testing set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Probabilistic Methods** | | **Clustering Model** | |
| GMM | Variational Bayesian GMM | K-Means | Birch |
| Evaluation Metric | Log-likelihood | | Silhouette coefficient | |
| Results | 30.69 | 29.85 | 0.63 | 0.60 |

It is hard to compare probabilistic methods and cluster methods under the same criteria. We calculated the proportion of risky client in the test dataset to do further analysis. Because we set up two groups, the group with a minor number of data points has been identified as risky clients. The K-means has identified 50% of customers as risky, which Birch has identified 40% if customers as risky. This high proportion result shows the clustering methods are not suitable in our task, because clustering methods are based on distance calculation. Two probabilistic methods have identified the risky proportion around 28%, which is more reasonable.

In GMM-EM, a number of components is an important parameter to determine how many groups will data points be categorized. In this project, we have set up it to 2 because we think there exist two groups in the dataset, risky and non-risky. However, there may exist more than two groups in the dataset. We have conducted an experiment to set up a number of the component from 2 to 9 in GMM-EM, and the results have been shown in Figure 3.

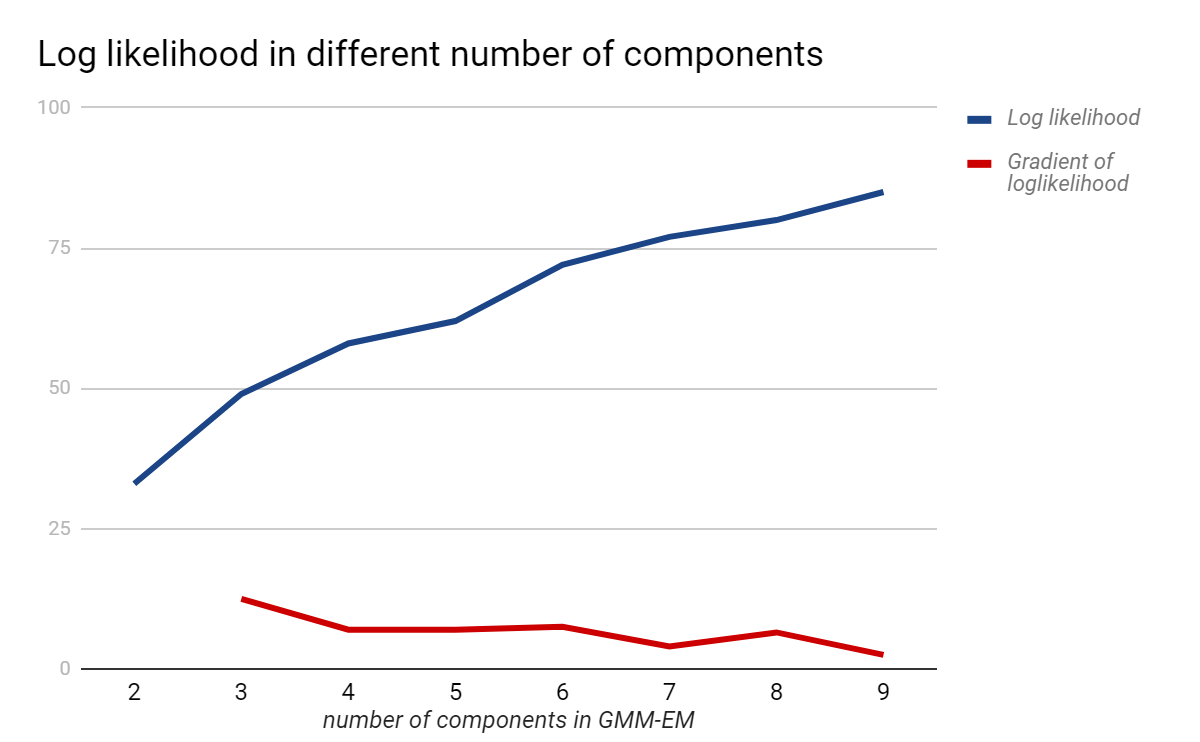


Figure 3. The performance of GMM-EM based on different components

The blue line shows the log-likelihood has increased when the number of components increased, while the red line should the gradient of log likelihood in a different number of components. The gradient of log-likelihood has achieved the maximum. It indicated there may exist three groups in the dataset. Due to the limitation of time, we may conduct more researches in this direction in the future. There are some limitations to this project. In probabilistic models, we initialized with Gaussian distribution. However, other distributions, such as Poisson distribution and binomial distribution may perform better in this dataset. Also, PCA has been used to reduce the number of features to 100 in this project. The performance may be enhanced if we increase the number of features to 200 or higher.

**5. Conclusion**

In this project, we have studied to use the unsupervised methodology to identify customers who may have risky behavior in the future. A large unlabeled dataset has been provided for this project, and we have proposed a four-step methodology to categorize clients into two groups. Four different methods have been tested in our pipeline, which includes probabilistic methods and clustering methods. In order to do the evaluation, cross-validation has been performed in the training dataset, and two different evaluation metrics have been conducted respectively. In this project, we have learned it is very hard to evaluate unsupervised algorithms, especially without true labels. The best performance methods may not perform well in the real world. GMM-EM has shown the best performance in the testing set, and we would suggest initializing with three distributions during training. There also exist some limitations in this project. Due to the large scale of the dataset, we have to choose some computational efficient algorithms. For example, autoencoder may be better than PCA during dimensionality reduction but it requires GPU resources. Changing the number of features during the PCA process may also affect the performance of models. Also, we have applied Gaussian distribution in the probabilistic models and Poisson distribution may have better performance. We will conduct more research in the future.

**References**

*1. Tomek, Ivan (1976). "An Experiment with the Edited Nearest-Neighbor Rule". IEEE Transactions on Systems, Man, and Cybernetics. 6 (6): 448–452.*

*2. Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. IEEE transactions on systems, man, and Cybernetics, (4), 580-585.*

*3. Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651-666.*

*4. Chandola, V.; Banerjee, A.; Kumar, V. (2009). "Anomaly detection: A survey".* [*ACM Computing Surveys*](https://en.wikipedia.org/wiki/ACM_Computing_Surveys)*. 41 (3): 1–58.* [*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier)*:10.1145/1541880.1541882.*

*5. Dokas, Paul; Ertoz, Levent; Kumar, Vipin; Lazarevic, Aleksandar; Srivastava, Jaideep; Tan, Pang-Ning (2002). "Data mining for network intrusion detection"*

*6. Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.*

*7. Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).*

*8. Bilmes, J. A. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. International Computer Science Institute, 4(510), 126.*

*9. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. (n.d.). Retrieved April 15, 2019, from https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0152173*

*10. Atiya, A. F. (2001). Bankruptcy prediction for credit risk using neural networks: A survey and new results. IEEE Transactions on Neural Networks, 12, 929–935.*

*11. Gaussian Mixture Model | Brilliant Math & Science Wiki. (n.d.). Retrieved April 15, 2019, from https://brilliant.org/wiki/gaussian-mixture-model/*

*12. Kou, G., Peng, Y., & Lu, C. (2014). MCDM approach to evaluating bank loan default models. Technological and Economic Development of Economy, 20(2), 292–311. https://doi.org/10.3846/20294913.2014.913275*

*13. Luna, D. K., Palshikar, G. K., Apte, M., & Bhattacharya, A. (2018). Finding Shell Company Accounts Using Anomaly Detection. Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, 167–174. https://doi.org/10.1145/3152494.3152519*

*14. Rayana, S., & Akoglu, L. (2015). Less is More: Building Selective Anomaly Ensembles. ArXiv:1501.01924 [Cs]. Retrieved from http://arxiv.org/abs/1501.01924*

*15.Silhouette (clustering). (2019). In Wikipedia. Retrieved from https://en.wikipedia.org/w/index.php?title=Silhouette\_(clustering)&oldid=88864951314, 2019, from https://www.sciencedirect.com/science/article/pii/S0957417409002577*

*16. Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, 103–114. https://doi.org/10.1145/233269.233324*

*17. The consumer loan default predicting model – An application of DEA–DA and neural network - ScienceDirect. (n.d.). Retrieved April*