Continuous Embedding Spaces for Bank Transaction Data

Ali Batuhan Dayioglugil[1] and Yusuf Sinan Akgul[2]

[1]Cybersoft R&D Center, İstanbul, Turkey
alibatuhandayioglugil@gmail.com
[2]Gebze Technical University, Gebze, Kocaeli 41400, Turkey
akgul@gtu.edu.tr

Abstract. In the finance world, customer behavior prediction is an important concern that requires discovering hidden patterns in large amounts of registered customer transactions. The purpose of this paper is to utilize this customer transaction data for the sake of customer behavior prediction without any manual labeling of the data. To achieve this goal, elements of the banking transaction data are automatically represented in a high dimensional embedding space as continuous vectors. In this new space, the distances between the vector positions are smaller for the elements with similar financial meaning. Likewise, the distances between the unrelated elements are larger, which is very useful in automatically capturing the relationships between the financial transaction elements without any manual intervention.

Although similar embedding space work has been used in the other fields such as natural language processing, our work introduces novel ideas in the application of continuous word representations technology for the financial sector. Overall, we find the initial results very encouraging and, as the future work, we plan to apply the introduced ideas for the abnormal financial customer behavior detection, fraud detection, new banking product design, and making relevant product offers to the bank customers.

Keywords: Feature embedding space, word representation, neural networks, segmentation, deep learning, machine learning

1 Introduction

Financial institutions are required by law to keep their data (customer, account, credit, etc.) in a structured form. This form is sufficient for basic daily operations like transactions and report generations. However, when the company needs to extract extra information from the data, domain knowledge becomes essential. Currently, human experts provide most of the domain knowledge into the information extraction process in the financial data, which makes this process time consuming, subjective, difficult to keep updated, and prone to human errors. Hence, automatically discovering hidden

adfa, p. 1, 2011.

patterns in the observed customer data is essential for several financial tasks such as fraud detection, new product offers, and customer behavior analysis.

The availability of raw customer transaction data made it very suitable for the application of machine learning techniques for this purpose. Many of these methods need numeric inputs and numeric output labels. Although financial data includes frequently non-numeric data, clustering, ranking and dummification [11] techniques can be used to produce digitized input and output data. However, obtaining the numeric or non-numeric output data is not trivial because it mostly requires hand labeling of the data by the experts, which is very expensive and time consuming for big scaled finance data.

The most popular techniques for financial data extraction are rule based approaches and other supervised machine learning methods. When rule based approaches are considered, keeping all the rules up to date and regularly adding the new ones (due to new customers, new scenarios or changing habits) are inevitable and also a constraint for a system's performance. Supervised machine learning methods are also powerful for detecting patterns in financial data and they are widely used in modelling customer behavior and fraud detection fields [1, 3, 7, 12], but they remain incapable of detecting deep relations and they require hand labeled data.

Recently, continuous word representations in high dimensional spaces brought a great impact in Natural Language Processing (NLP) community by their ability to unsupervisedly capture syntactic and semantic relations between words, phrases [8, 9] and even complete documents [13]. Employment of these representations produced very promising results with the help of available large text bases in the fields of language modeling and language translation.

Motivated from the success of continuous word representations in the NLP world, this work proposes to represent the financial transaction data in a continuous embedding space to take advantage of the large unlabeled financial data. To the best of our knowledge, our system is the first to propose the continuous embedding space methods for the finance data. The resulting vector representations of the transactions are similar for the semantically similar financial concepts. We argue that, by employing these vector representations one can automatically make information extraction from the raw financial data. We performed experiments to show the benefits of these representations.

This paper is organized as follows; in Section 2 data properties are given, accordingly parameters and details to create word vectors are explained. Section 3 gives the experiments with the proposed model. Section 4 discusses the obtained results and argues possible financial applications of this method as future works.

2 Embedding Spaces for Financial Data

Raw transaction data consists of timewise ordered individual transactions. We argue that this data is very similar to natural language sentences. Researchers of NLP domain recently achieved very favorable language modeling results by projecting the words and sentences to a continuous embedding space. We propose to use a similar idea for the raw transaction data. Representation of the transaction data in a new space can lead us to demonstrate semantic and syntactic relations among transaction elements.

2.1 Transaction Data

Throughout this work, a darkened transaction data on a medium-sized Turkish bank is used. The transaction data $TRX= \{T_i\}$, $i=1..n$, is an ordered set of individual transactions $T_{i,}$ where n is the total number of transactions that can reach up to hundreds of millions. Each transaction $T_i= \{t_{i,j}\}$, $j=1..m$, is a set of m transaction elements. In our case, we took m=10 after eliminating irrelevant elements (such as date and voucher id) and elements including private (Name, security number etc.) or missing data. Each transaction element is a variable that can take either numerical or categorical values. The two numerical transaction elements are *age* and *amount*. Categorical elements with their corresponding possible number of values are *gender* (3), *housing status* (6), *marital status* (8), *education level* (9), *business segment* (15), *customer type* (19), *profession* (10), and *transaction process group code* (56)[1]. We eliminated the transactions that were initiated by the bank because these types of transactions do not provide any significant information on customer behavior. Thus, our dataset contains only customer based transactions. Since the number of transaction element values are much bigger for the numerical elements, we coarsely clustered age and amount transaction element values as shown in Table 1. We have 126 different transaction element values for only categorical elements and 11 different clustered elements values, which makes a total of 137 possible element values.

As mentioned before, we propose to use continuous vector representations for the data *TRX* as it is done for the natural language words. If we like to draw more concrete parallels between transaction data and the natural languages; our transaction words can be considered as natural language words and individual transactions can be seen as natural language sentences. The total number of possible element values can be compared to the number of unique words in a natural language. As expected, since the possible number of transaction element values are much smaller than the natural language vocabulary sizes, our embedding space dimensions will be much smaller than the frequently used dimension sizes in the NLP domain.

Table 1. Categorical value assignments for numeric values

Transaction Element Name	Assigned Cluster Values	Number of Values
Transaction Amount (Turkish Liras)	Amnt1(Amount<1), Amnt2(1<Amount<100), Amnt3(100<Amount<200), Amnt4(200<Amount<1000), Amnt5(1000<Amount<3000), Amnt6(3000<Amount<10000), Amnt7(10000<Amount)	7
Age	Young(age<=25), Young_Adult(25<age<=40), Middle_Age(40<age<=55), Old(55<age), No_Age (age is Null)	4

[1] All feature names are translated from Turkish to English

2.2 Transaction Elements Vectors

Generating word embedding from context dependent corpora using neural networks is successfully applied by feed-forward networks [2] and later by recurrent networks (RNN) with higher accuracies [10, 11]. RNN is a special form of ANN which has self-recurrent connections on hidden nodes that can keep the short time memory of word relations with respect to a predefined window size. Other types of methods, such as [8, 9] were proposed with varying efficiencies and precision. In this work, we adopted word2vec library [8],to create our transaction element value vectors(Figure 1)

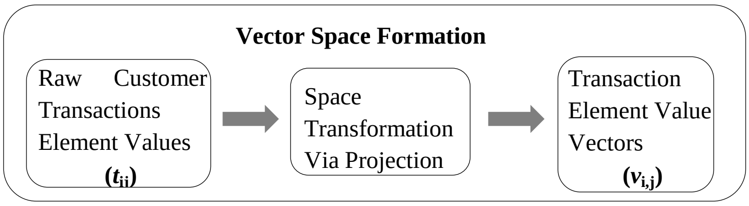

Vector Space Formation
Raw Customer Transactions Element Values
(t_{ij})
Space Transformation Via Projection
Transaction Element Value Vectors
($v_{i,j}$)

Fig. 1. Proposed vector space formation model

We selected window size as the total number of transaction elements in a transaction (w=10). This is similar to choosing the window size as the average sentence length in NLP applications. Different dimensions of vectors (100, 50 and 20) were tested. For each of these dimensions, we measured the distances between semantically similar transaction element values such as “small farmer” and “medium farmer”. The tested dimensions produced almost the same similarity values for the same element values. In order to reduce the complexity of the network with a small vocabulary of 137 values, we chose an embedding space with 20 dimensions.

For the vector similarities, we used the cosine similarity (Equation 1) which is one of the most popular metrics used for this type of comparison.

$$sim(A,B) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{d} AiBi}{\sqrt{\sum_{i=1}^{d} A_i^2}\sqrt{\sum_{i=1}^{d} B_i^2}}, \qquad (1)$$

where A and B are the vectors of dimension d, A_i and B_i are the elements of these vectors, respectively.

We observed that the skip-gram method captures stronger relations between the transaction element value vectors with similar financial meanings within the embedding space. These relations are given statistically in Table 2. We use cosine similarity to measure the distance between the transformed transaction elements. Note that cosine similarity can be at most 1.0. A quick analysis of Table 2 reveals that most similar transaction elements for a given transaction element value are very similar in terms of financial semantics. For example, the most similar transaction element value to “Young Adult” is the “Middle Age” value. Note that these vector assignments were obtained fully automatically without any human supervision.

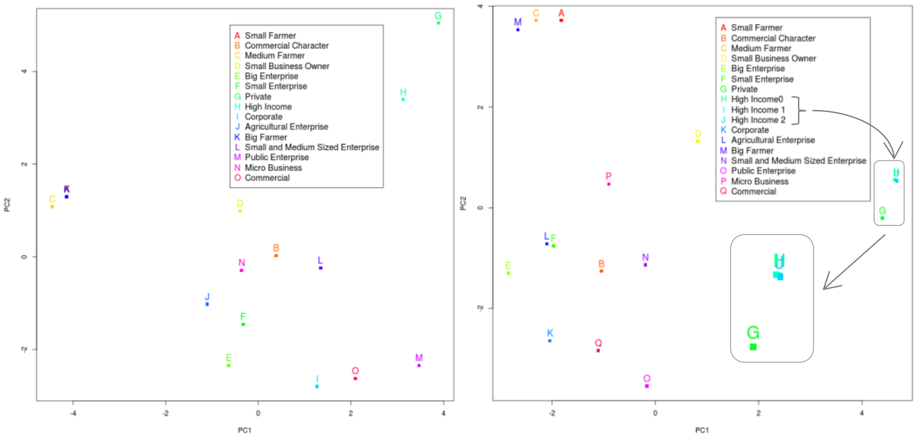
To visualize all the segment vectors in a reduced 2 dimensional space, Principal Component Analysis (PCA) is applied on the obtained embedded vectors and the calculated relations are shown graphically in Figure 2 *(a)* for the *business segment* element values. As can be seen in this figure, semantically close values are assigned closer vector positions such as all 3 farmer types (small, medium, and large farmers).

Table 2. Closest neighbours of three transaction element values (Age, business segment and profession respectively) with respect to cosine similarities

Young Adult		**Small Farmer**		**Student**	
Middle Age	0.958798	Medium Farmer	0.995632	Unemployed	0.868379
Old	0.879234	Big Farmer	0.944026	Housewife	0.833956
Young	0.833032	Small Business Owner	0.829680	Public Sector Employee	0.766121
TGC32[2]	0.490597	Micro Business	0.774104	Private Sector Employee	0.737213
No Age	0.460723	Agricultural Enterprise	0.768568	Retired	0.692351
		Private	0.748798	No Profession	0.652012
		SME[3]	0.707125	Retired Working	0.589541
		Corporate	0.652397	Self-employment	0.581086

2.3 Representing Transactions as Vectors

After the transaction element value vectors are estimated, a complete transaction T_i can be represented in the vector form by concatenating the transaction element value vectors of $v_{i,j}$ into the vector V_i. Since we use 20 dimensional transaction element vectors and we have 10 transaction elements in a transaction, the resulting transaction vector V_i would be 200 dimensional.



(a) ***(b)***

Fig. 2. PCA of b*usiness segment* element value embedding vectors *(a)* and embedding vectors of same element values with artificially divided 'High-income' value *(b)*

3 Experiments

We performed two types of experiments for the validation of the estimated embedding vectors of financial data. For the first experiment, we took all the transactions that contains the "high income" element value as the business segment. We artificially

[2] Transaction Process Group Code 32

[3] Small and Medium Sized Enterprise

replaced the original business segment values with 3 different dummy values (High_Income_0, High_Income_1, and High_Income_2) for these transactions in a random manner. We then estimated the vectors for this modified transaction data as described in Section 2. The obtained element value vectors are shown in Figure 2 (b) in reduced dimensions. As the figure shows, the artificially modified element value vectors turn out to be very close in this space, which shows that the proposed model is able to capture the semantic relations in the finance data.

For the second experiment, we like to explore the idea of using semantically closer representations of feature labels may successfully lead to recognition of hidden patterns in data. For this purpose, the transaction vectors ($\bar{V}_i$) without the *business segment* element vectors are used as inputs of an ANN. The *business segment* vectors (B_i) is used as the output variable. In other words, the input to the ANN is a 180 dimensional vector that does not include any *business segment* information. The output is a 20 dimensional vector that we expect to produce vectors positions close to *business segment* element value vectors.

Our dataset contains around 1.8M transactions (A four week day period). In order to have exact scalar values for output nodes, the activation function is not implemented in the output layer of the ANN. The ANN uses cosine similarity (1-cosine distance) as it is statistically more robust, to calculate the differences (error) between model output vectors and ground truth vectors. Stochastic Gradient Descent (SGD) optimizer is adopted with different learning rates and for different hidden layer node counts using hyperbolic tangent (*tanh*) as the activation function in hidden layer. In the training part of the network, the best results are obtained by an ANN which contains a single hidden layer with 60 nodes with learning rate of 0,018 where batch size was 100 for error backpropagation.

In order to avoid overfitting, k-fold cross validation (k=4) method is used during training and testing. After cross-validation is completed, we have predicted output vectors ($\hat{B}_i$) for each transaction in input data. These 20 dimensional predicted vectors are compared with true element value vectors using cosine similarities. Each possible element value vector is assigned a similarity score, which are sorted to produce the top-1, top-3, and top-5 most likely predictions. Table 3 lists the final scores obtained from this experiment. Although around 70% business segment prediction accuracy seems not very high, we should note that these results are obtained without any supervision. Furthermore, considering only top 5 matching results, one can easily conclude that these unsupervised predictions can be used as reliable cues for a fraud detection system.

Table 3. Proposed model accuracies

	Top 1 Match	Top 3 Matches	Top 5 Matches
Accuracy (%)	**72,4**	**90,4**	**93,8**

4 Discussion and Future Work

In the financial world, decision making process needs reliable arguments which must always be up-to-date in order to keep up with dynamic nature of the domain. We believe that the interpretation of large scaled structured data with the implementation of newest methods in NLP, such as embedding vectors, can produce strong results. We observed that increasing the epoch number does not enhance the overall accuracy but, in order to obtain better results, one might claim that using larger amounts of transaction data (months, years) with much more features would yield higher accuracies.

The competition between banks and other financial institutions increases as technology advances. They have to register and process higher volumes of raw data day by day. While having such big data brings many risks to manage, it also gives opportunities to create relevant products and also ensure customer content by analyzing customer behavior. Achieving these targets by traditional systems is not efficient anymore. The proposed model in this work has the capability to handle these hot topic subjects in financial world. As the future work, we plan to extend the proposed method to detect fraud attacks, define abnormal customer activities, and provide valuable knowledge to create new products.

Acknowledgments

We would like to thank Tuğba Halıcı and Sait Şimşek for their valuable technical contribution to this work.

References

1. Ando, Y., Hidehito, G., Hidehiko, T.: Detecting Fraudulent Behavior Using Recurrent Neural Networks (2016)
2. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research, pp. 1137-1155 (2003)
3. Busta, B., Weinberg, R.: Using Benford's law and neural networks as a review procedure, Managerial Auditing Journal 13 (6), pp. 356–366 (1998). doi:10.1108/02686909810222375
4. Hermans, M., Schrauwen, B.: Training and analysing deep recurrent neural networks. In Advances in Neural Information Processing Systems, 190-198 (2013)
5. Kirkos, E., Spathis, C., Manolopoulos, Y.: Data mining techniques for the detection of fraudulent financial statement, Expert Systems with Applications 32, pp. 995–1003 (2007)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. (2013)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pp. 3111-3119 (2013)
8. Mikolov, T., Zweig, G.: Context dependent recurrent neural network language model. In SLT, pp. 234-239 (2012). doi:10.1109/slt.2012.6424228
9. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S.: Recurrent neural network based language model. In Interspeech (Vol. 2, p. 3) (2010)
10. Sohl, J.E., Venkatachalam, A.R.: A neural network approach to forecasting model selection, Information & Management 29 (6), pp. 297–303 (1995)
11. Wiese, B.J.: Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks. Diss. Department of Computer Science, University of the Western Cape (2007). doi: 10.1007/978-3-642-04003-0_10
12. Zou, W. Y., Socher, R., Cer, D. M., Manning, C. D.: Bilingual Word Embeddings for Phrase-Based Machine Translation. In EMNLP, pp. 1393-1398 (2013)