Краткие сообщения

Short communications

УДК 004.852

ФУНКЦИЯ ПОТЕРЬ, УЧИТЫВАЮЩАЯ СЕМАНТИКУ ПРОСТРАНСТВА, ДЛЯ СИНТЕЗА ЭМБЕДДИНГОВ НА ТРАНЗАКЦИОННЫХ ДАННЫХ

М. Е. ВАТКИН[1)]***, Д. А. ВОРОБЕЙ***[1)]***, М. В. ЯКОВЛЕВ***[1)]***, М. Г. КРИВОВА***[1)]

[1)]*ОАО «Сбер Банк», бул. Мулявина, 6, 220005, г. Минск, Беларусь*

Популярные в банковской сфере транзакционные данные часто представляются в виде разреженных (с большим количеством признаков) векторов. Использование разреженных векторов в задачах глубинного обучения является неэффективным и может вести к переобучению. Для извлечения полезных признаков в пространстве меньшей размерности широко применяют автокодировщики. В настоящей работе предлагается новая функция потерь, которая основана на метрике, оценивающей качество отображения исходных табличных данных в пространство эмбеддингов. Эта функция служит для преобразования снижения размерности и позволяет сохранить структуру отношений объектов исходного пространства. Полученные результаты показывают улучшение качества получаемых эмбеддингов посредством использования новой функции потерь в комбинации с традиционной средней квадратической ошибкой функции.

Ключевые слова: данные; эмбеддинг; вектор; функция потерь; автокодировщик.

Образец цитирования:
Ваткин МЕ, Воробей ДА, Яковлев МВ, Кривова МГ. Функция потерь, учитывающая семантику пространства, для синтеза эмбеддингов на транзакционных данных. *Журнал Белорусского государственного университета. Математика. Информатика.* 2022;1:97–102 (на англ.).
https://doi.org/10.33581/2520-6508-2022-1-97-102

For citation:
Vatkin ME, Vorobey DA, Yakovlev MV, Krivova MG. Space semantic aware loss function for embedding creation in case of transaction data. *Journal of the Belarusian State University. Mathematics and Informatics.* 2022;1:97–102.
https://doi.org/10.33581/2520-6508-2022-1-97-102

Авторы:
Максим Евгеньевич Ваткин – главный специалист по данным.
Дмитрий Александрович Воробей – специалист по данным.
Максим Вадимович Яковлев – специалист по данным.
Марина Григорьевна Кривова – специалист по данным.

Authors:
Maksim E. Vatkin, chief data scientist.
mevatkin@bps-sberbank.by
https://orcid.org/0000-0002-6923-9998
Dmitry A. Vorobey, data scientist.
davorobey@bps-sberbank.by
https://orcid.org/0000-0001-9063-6077
Maksim V. Yakovlev, data scientist.
mvyakovlev@bps-sberbank.by
https://orcid.org/0000-0003-4722-9753
Marina G. Krivova, data scientist.
mgkrivova@bps-sberbank.by
https://orcid.org/0000-0003-0345-4828

Журнал Белорусского государственного университета. Математика. Информатика. 2022;1:97–102
Journal of the Belarusian State University. Mathematics and Informatics. 2022;1:97–102

SPACE SEMANTIC AWARE LOSS FUNCTION FOR EMBEDDING CREATION IN CASE OF TRANSACTION DATA

***M. E. VATKIN*[a], *D. A. VOROBEY*[a], *M. V. YAKOVLEV*[a], *M. G. KRIVOVA*[a]**

[a]*Sber Bank, 6 Muliavina Boulevard, Minsk 220005, Belarus*
Corresponding author: M. E. Vatkin (mevatkin@bps-sberbank.by)

Transaction data are the most popular data type of bank domain, they are often represented as sparse vectors with a large number of features. Using sparse vectors in deep learning tasks is computationally inefficient and may lead to overfitting. Autoencoders are widely applied to extract new useful features in a lower dimensional space. In this paper we propose to use a novel loss function based on the metric that estimates the quality of mapping the semantic structure of the original tabular data to the embedded space. The proposed loss function allows preserving the item relation structure of the original space during the dimension reduction transformation. The obtained results show the improvement of the resulting embedding properties while using the combination of the new loss function and the traditional mean squared error one.

Keywords: data; embedding; vector; loss function; autoencoder.

Introduction

Every day, a large number of payment transactions using bank cards are performed, which details are collected on the bank's servers. Such data contains information about the bank's client's behaviour, which can later be used by the bank in various forecast models. At the same time, if we want to use such data, it must be presented in a certain format, for example via transformation into a fixed-length vector, where each coordinate acts as a counter for the number of transactions of a particular type in a certain time period (e. g., the first coordinate shows the number of transactions at gas stations, the second one does it for public catering places, etc.). Such vectors can describe the client's behaviour at certain time intervals, however, due to a large number of possible transactional activity categories, these vectors have a large number of coordinates, many of which are zero, in other words, clients are described by sparse vectors.

The use and storage of such vectors is computationally inefficient and may also lead to overfitting of predictive models. A popular solution to this problem is to use autoencoders, which first encode data into a smaller space, and then reconstruct the original data from it. The process of training the model can be described as reducing the reconstruction error by updating the model's weights, and as a result, we get a mapping of the source data into a space of smaller dimension, preserving the maximum of the original information about objects.

The quality of the obtained representations is measured by the recovery error, however, it was shown in [1] that this metric is not a reliable indicator of their applicability to the final problem. In addition, it does not give us an idea of how the relationships of representations are arranged, which reduces the confidence in the results obtained. Thus, we would like to have representations that contain as much information about the original objects as possible, and also preserve their semantic relationships in the embedding space. In other words, we want the representation of objects in the new space to be formed so that, using the vector additional operation, we can move from one representation to another and this transition is meaningful. This corresponds to the analogies presented by T. Mikolov [2], where we have pairs $x : y$ and $a : b$ in the original space, where x is semantically connected with y, as well as a with b, for example «man» : «king» and «woman» : «queen», so we want this connection to be reflected in the embedding space. To fulfill this condition, we propose a modification of the loss function, which will allow not only compressing information into a space of smaller dimension, but also building the resulting vectors so that they reflect semantic relationships, which is possible due to the structure of the source data itself.

General description of problem

Let us introduce the notation: mapping $f : R^n \to R^m$ $(m < n)$ to be called embedding (encoder) and mapping $g : R^m \to R^n$ to be called decoder. An object is represented by a vector $X \in R^n$, where each coordinate displays the number of transactions executed for a product or service of a certain category.

The goal is to build an embedding that reflects the semantic relationships between objects. More strictly, this can be formulated as follows:

$$g\big(f(x) + f(y)\big) = z, \tag{1}$$

where z is an object corresponding to the union of the meanings of objects x and y. Referring to the source data, you can see that each object itself reflects the client's behaviour, moreover, there is already a meaningful addition operation in the source space. Indeed, if combination the two original vectors results as a third vector that describes the third client which behaviour corresponded to the union of the first two. Then it is possible rewrite (1) as

$$g\left(f\left(x\right)+f\left(y\right)\right)=x+y.$$

Because it is assumed that the reconstruction error of autoencoder is zero, as well as the f is an injective function, then $g\left(f\left(x+y\right)\right)=x+y$ and thus $f\left(x\right)+f\left(y\right)=f\left(x+y\right)$. So actually equation (1) says that with some additional conditions f is a homomorphism. It seems that using equation (1) can transform our autoencoder into principal component analysis since it is a linear map, and the autoencoder with a linear encoder resembles principal component analysis projection [3]. This could be the case when autoencoder has a zero reconstruction error which is impossible due to different sizes of the embedding space and the objects space.

An experiment methodology

Data description. Used dataset contains information about 284.807 transactions performed during two days by European cardholders in September 2013 with 492 fraudulent transactions [4]. The reason to use this dataset is to implement the method of fraud detection described in another article [5], namely to fit the autoencoder on non-fraud data and, assuming that fraudulent transactions will be recovered by an autoencoder with a larger recovery error, use reconstruction error to verify if the new transaction is fraudulent or not. Thus the task is to compare different models. To do this, the dataset was separated into the train part that consists of 80 % of the transactions, all of them being normal, and the test part that contains the remaining 20 % with all 492 fraudulent transactions. Since to use our approach we need to be able to add the vectors of the original space, we will only use the attributes from $V1$ to $V28$.

Training metrics. For the training process evaluation, we use two metrics.

1. The mean squared error (MSE) of objects and the autoencoder's predictions to check the reconstruction abilities of our autoencoder:

$$\frac{\sum_{i}^{N}\left(g\left(f\left(x_i\right)\right)-x_i\right)^2}{N}.$$

2. Custom metric called as mean semantic preserving error (MSPE):

$$\frac{\sum_{i}^{N}\sum_{j}^{M}\left(g\left(f\left(x_i\right)+f\left(y_j\right)\right)-\left(x_i+y_j\right)\right)^2}{NM},$$

where $x, y \in X$. However, it is computationally inefficient to choose all y objects from X, so for the performance reason the $M=100$ random y objects were chosen from X for every object x to be used for computation of given loss function.

Models. Autoencoders with the same architecture were used for each model: InputLayer(shape=(28)) – Dense(21, activation='elu') – Dense(14, activation='elu') – Dense(7, activation='elu') – Dense(14, activation='elu') – Dense(21, activation='elu') – Dense(28) but different losses:

- the sum of the MSE and the MSPE (model 1);
- the MSE as a model loss (model 2);
- the MSPE as a model loss (model 3).

The batch size equals 1000, the number of epochs is 50, the optimiser is Adam with the default hyperparams. As predictions for the fraud detection task, we compute the mean squared error between the input vector and its reconstruction.

Classification metrics. Since our task is essentially a binary classification task with imbalanced classes, we will use three metrics.

1. *Maximum F1-score.* F1-score is a harmonic mean of precision and recall:

$$\text{F1-score}=\frac{2\cdot\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}},$$

where precision is the number of objects classified by the algorithm as 1 (fraud) while they really belong to class 1, divided by the number of objects classified by the algorithm as 1, which shows whether we can rely on

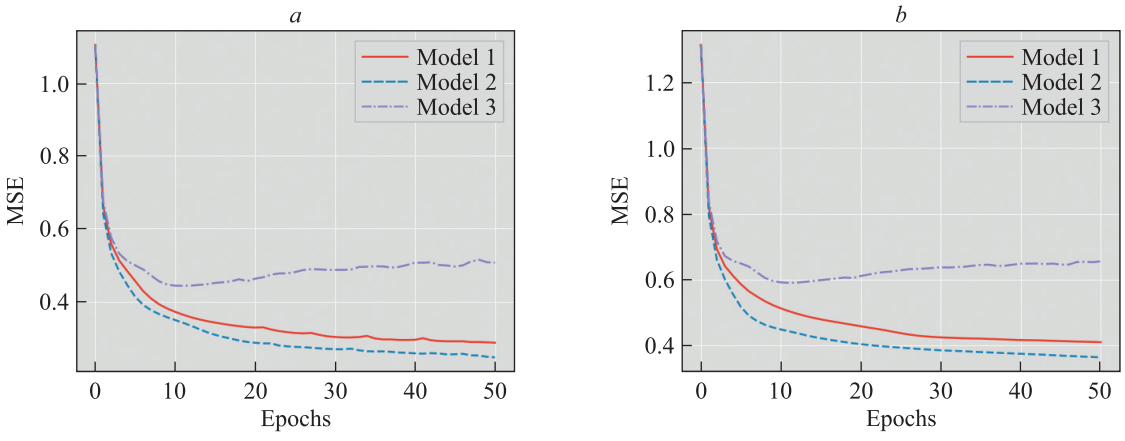

Fig. 1. The reconstruction loss (model 1 uses MSE and MSPE, model 2 uses MSE only, model 3 uses only MSPE as the loss): *a* – train part; *b* – test part

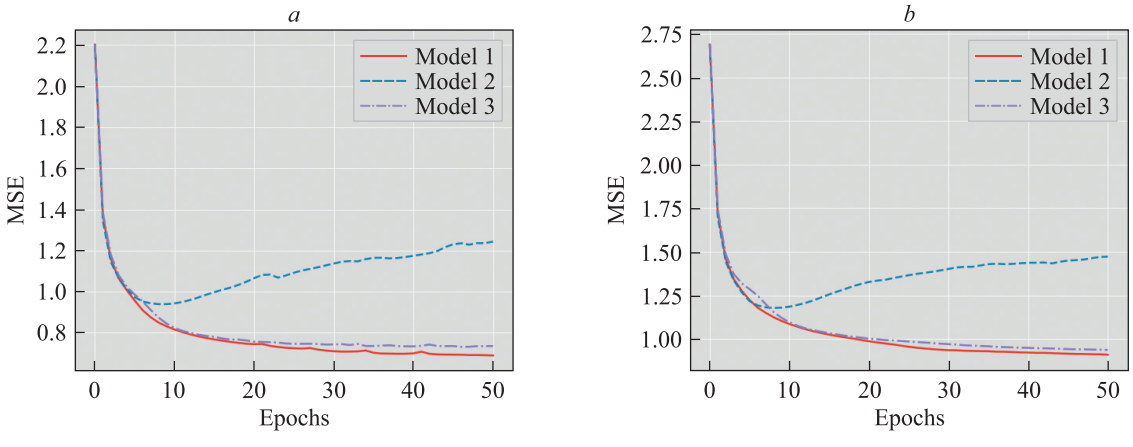

Fig. 2. The MSPE (model 1 uses MSE and MSPE, model 2 uses MSE only, model 3 uses only MSPE): *a* – train part; *b* – test part

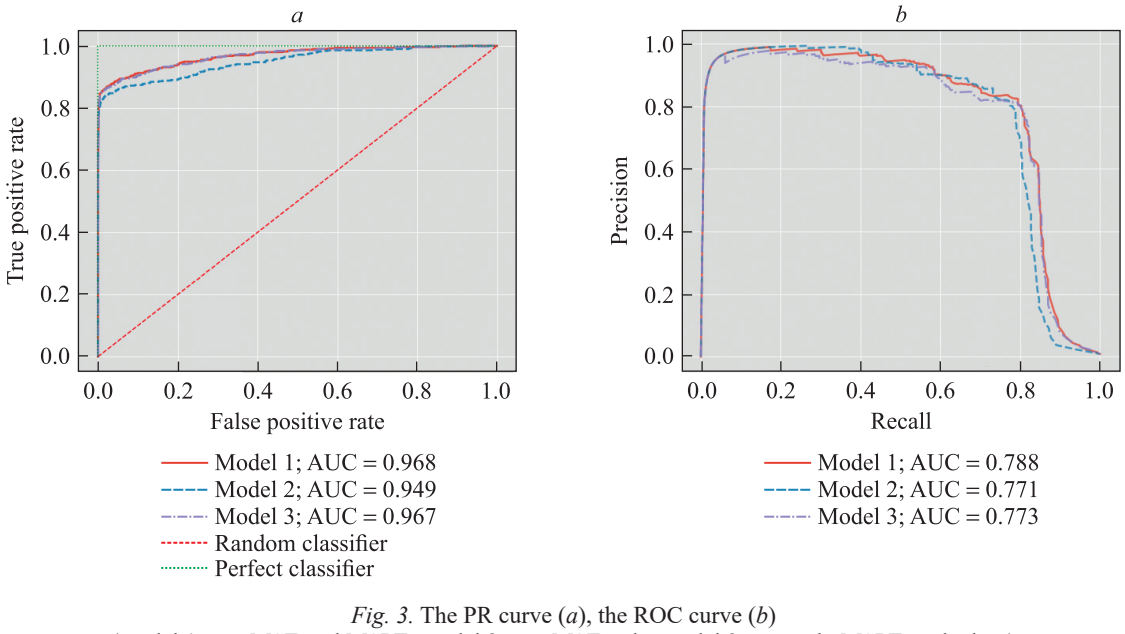

Fig. 3. The PR curve (*a*), the ROC curve (*b*) (model 1 uses MSE and MSPE, model 2 uses MSE only, model 3 uses only MSPE as the loss)

our algorithm when it classifies objects as class 1; recall is the number of objects classified by the algorithm as 1 (fraud) while they really belong to class 1, divided by the number of objects from class 1, which shows how many objects of class 1 it can find among all objects of class 1. However, it is needed to balance these two metrics, and for this purpose F1-score is used. Since it isn't known which threshold to use for the algorithm, and F1-score depends on the threshold, the maximum F1-score within all possible values was used.

2. *PR AUC.* It is an area under the precision recall curve; the closer this metric is to 1, the better the model is.

3. *ROC AUC.* It is an area under the receiver operating characteristic, the closer this metric is to 1, the better the model is. To explain how this metric is computed, we need to know the two other metrics: true positive rate (TPR), which is actually a recall, and false positive rate (FPR), which is the number of objects classified by the algorithm as 1 (fraud) while they really belong to class 0, divided by the number of objects from class 0.

Results of experiment

Figure 1 shows the reconstruction error, fig. 2 shows the mean semantic preserving error for each epoch during the training, fig. 3 shows the ROC and PR curves, and table 1 shows the quality of models in terms of the fraud detection task. As we see from fig. 1, model 2 shows the best reconstruction loss, model 1 shows a slightly worse result, and model 3 performs much worse. However, in fig. 2, models 1 and 3 have similar results while model 2 performs much worse. So, we can make a conclusion that a combination of two losses allows getting good results in terms of both losses while using only one leads to better results in the corresponding metric but much worse in the other. Yet, good quality in those metrics doesn't necessarily lead to good results when trying to apply the models to an external task. Model 1 shows the best reconstruction ability, however it has the worst result in the fraud detection task (table 1), while model 3 with the worst reconstruction ability has better results. Also, the best model doesn't have the best reconstruction ability, but performs well in terms of both metrics. Thus, we can make the following conclusions: the better reconstruction loss in autoencoders does not necessarily lead to better results on an external task, and we can try to combine the reconstruction loss with the mean semantic preserving error in our loss function, which can lead to better results on external tasks. We can also compare the PR AUC of our best model with the results of other models from article [6] since this indicator is better than the ROC AUC for evaluating the performance in tasks with imbalanced data [7; 8]. As we can see from table 2, our best model scores the second, which proves that our approach is comparable to the existing algorithms for solving the fraud detection task.

Table 1

The comparison of the classification metrics

Algorithm	Maximum F1-score	PR AUC	ROC AUC
Model 1 (MSE + MSPE)	0.808	0.788	0.968
Model 3 (MSPE)	0.802	0.773	0.967
Model 2 (MSE)	0.794	0.771	0.949

Table 2

The PR AUC comparison with other algorithms

Algorithm	PR AUC
Bagging	0.825
Model 1 (MSE + MSPE)	0.788
C4.5	0.745
Naive bayes	0.080

Conclusions

The paper considers an actual problem of the embedding construction by applying neural network autoencoder model. In particular, the new semantic preserving error function was introduced as a loss function for autoencoder training process. Given function allows preserving semantic relations between objects in embedding space for tabular data. However, not all properties of the resulting space were considered in details. For example, given approach could be used while training neural networks of other type like in [9]. Mentioned possibilities and properties remain to be explored in the future work.

References

1. Gupta P, Banchs RE, Rosso P. Squeezing bottlenecks: exploring the limits of autoencoder semantic representation capabilities. *Neurocomputing*. 2016;175(PB):1001–1008. DOI: 10.1016/j.neucom.2015.06.091.

2. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Burges CJC, Bottou L, Welling M, editors. *NIPS-2013. Proceedings of the 26th International conference on neural information processing system; 2013 December 5–10; Lake Tahoe, Nevada, USA. Volume 2.* New York: Curran Associates Inc.; 2013. p. 3111–3119.

3. Bourlard H, Kamp Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics.* 1988;59:291–294. DOI: 10.1007/BF00332918.

4. Credit card fraud detection [Internet]. Cambridge: Machine Learning Group; 2018 [cited 2021 March 5]. Available from: https://www.kaggle.com/mlg-ulb/creditcardfraud/data.

5. Al-Shabi MA. Credit card fraud detection using autoencoder model in unbalanced datasets. *Journal of Advances in Mathematics and Computer Science.* 2019;33(5):1–16. DOI: 10.9734/jamcs/2019/v33i530192.

6. Husejinović A. Credit card fraud detection using naive Bayesian and C4.5 decision tree classifiers. *Periodicals of Engineering and Natural Sciences.* 2020;8(1):1–5. DOI: 10.21533/pen.v%25vi%25i.300.

7. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One.* 2015;10(3):e0118432. DOI: 10.1371/journal.pone.0118432.

8. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: Cohen WW, Moore A, editors. *ICML-06. Proceedings of the 23rd International conference on machine learning; 2006 June 25–29; Pittsburgh, USA.* New York: Association for Computing Machinery; 2006. p. 233–240. DOI: 10.1145/1143844.1143874.

9. Marushko EE, Doudkin AA, Zheng X. Identification of Earth's surface objects using ensembles of convolutional neural networks. *Journal of the Belarusian State University. Mathematics and Informatics*. 2021;2:114–123. DOI: 10.33581/2520-6508-2021-2-114-123.

Received 21.10.2021 / revised 01.02.2022 / accepted 14.02.2022.