Contrastive Learning for Event Sequences with Self-Supervision on multiple domains

Gleb Mazanov [1] **Nikolay Kotoyants** [1] **Ivan Gurev** [1] **Anna Iliushina** [1] **Viacheslav Naumov** [1]

Abstract

We address the problem of self-supervised learning on discrete event sequences generated by real-world users. Self-supervised learning incorporates complex information from the raw data in low-dimensional fixed-length vector representations that could be easily applied in various downstream machine learning tasks. In this project, we review a new method "CoLES", which adapts contrastive learning, previously used for audio and computer vision domains, to the discrete event sequences domain in a self-supervised setting. In order to estimate method performance, we have considered binary classification task taken from Data Fusion Contest 2022

Github repo: https://github.com/fiestaxxl/ML-project

1. Introduction

Not every sequential discrete data features high mutual information between a single item and its immediate neighborhood. For example, log entries, IoT telemetry, industrial maintenance, user behavior, travel patterns, transactional data, and other industrial and financial event sequences typically consist of interleaved relatively independent substreams. The most state-of-the-art representation learning methods for token and sequence embedding from the NLP or CV are not guaranteed to capture the peculiarities of such financial data, which exhibits customer behavior of a certain type and constitutes valuable information for the fraud prevention and development of efficient financial products. The article (Babaev et al., 2022) proposes a novel self-supervised method for embedding discrete event sequences, called Contrastive Learning for Event Sequences (CoLES), which is based on contrastive learning, with a special data augmentation strategy. The use of CoLES embeddings significantly improves the performance of existing models in downstream tasks and provides significant financial benefits. It is necessary to compare CoLES on several public datasets of event sequences and prove that CoLES representations consistently outperform other methods in various tasks.

2. Problem statement

While the method proposed in this study could be studied in different domains, we focus on discrete sequences of events. Assume there are some entities e and that each entity's lifetime activity is observed as a sequence of events $x_e = \{x_e(t)\}_{t=1}^{T_e}$. Entities could be people or organizations or some other abstractions. Events $x_e(t)$ may have any nature and structure (e.g., transactions of a client, click logs of a user), and their components may contain numerical, categorical, and textual fields. According to theoretical framework of contrastive learning proposed in [(Babaev et al., 2022)], each entity e is a latent class, which is associated with a distribution P_e over its possible samples (event sequences). However, unlike the problem setting of (Saunshi et al., 2019), we have no positive pairs, i.e. pairs of event sequences representing the same entity e. Instead, only one sequence (x_e) is available for the entity e. Formally, each entity e is associated with a latent stochastic process $X_e(t) = X_e(t)_{t\geq 1}$ and we observe only a single finite realisation $x_e = x_e(t)_{t=1}^{T_e}$ of it. Our goal is to learn an $encoder M$ that maps event sequences into a feature space R^d in such a way that the obtained $embedding$ $x_e \to c_e = M(x_e) \in R^d$ encodes the essential properties of e and disregards irrelevant noise contained in the sequence. That is, the embeddings $M(x^{'})$ and $M(x^{''})$ should be close to each other, if $x^{'}$ and $x^{''}$ were paths generated by the same process $X_e(t)$, and further apart, if generated by distinct processes. After obtaining embeddings we train a model $f : M(X) \to \{0, 1\}$.

3. Related work

Contrastive learning has proven to be effective in creating low-dimensional representations (embeddings) for different objects, including images, texts, and audio recordings. However, these studies aim to recognize the object based

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Gleb Mazanov <Gleb.Mazanov@skoltech.ru>.

Final Projects of the Machine Learning 2023 Course, Skoltech, Moscow, Russian Federation, 2023.

on its sample, and their supervised approaches cannot be used in our scenario. This is because their training datasets explicitly include multiple independent samples of each object, forming positive pairs that are crucial for learning. There are papers focused on supervised learning for discrete event sequences, such as [(Dmitrii Babaev & Umerenkov, 2019), (Tobback, 2019)], self-supervised pretraining is not utilized in those works. Other articles have proposed different methods for encoding, similar to the one described in our paper. However, none of these articles, except [(Babaev et al., 2022)] worked with the data type that was used in our paper.

CoLES method were tested and compared to other approaches to self-supervised embedding such as SOP (sequence order prediction), NSP (next sentence prediction), RTD (replaced token detection), CPC (contrastive predictive coding). Proposed method has shown competitive quality rate on different prediction tasks such as age group prediction, churn prediction, assessment prediction and scoring. Our task was to test CoLES effectiveness for the discrete task of education level prediction. Machine learning based methods for prediction of author's educational level were discussed in [(Gomzin, 2018)],[(Florea & Roman, 2021)], but most of proposed solutions were based on the SVM classifier and TF-IDF feature extractor parameters tuning and the dataset were prepared for text analysis, as opposed to clickstream events and transactions we use in our tests. Also, the literature on applying a self-supervised contrastive learning for similar datasets is still rather limited in spite of the obvious research potential of such applications.

4. Algorithms and Models

4.1. Prepocessing

In this paper we are not dealing with data collection process due to the focus on comparison of CoLES and other methods sequences. The data provided by the organizers of the Data Fusion Contest 2022 [(Science, ODS.ai)] were used to complete this task. A total of 3 datasets were used. The first one is a list of user transactions. This dataset contains information about the buyer number, time of payment and other parameters(user id, event time, mcc code, currency mk, transaction atm). The number of rows in this file is 19821910.

The other dataset is clickstream, which is a sequence of hyperlinks through which users go to a given site. Both of these csv files had a lot of weight and no sorting. As a result of preprocessing work, the final datasets were created, which are lists of dictionaries, where the keys are the features. Additionally, all data are converted into tensor format, to speed up work with large amounts of data.

The result of the training is a model for predicting whether customers have higher education at the selected site. Therefore, a dataset with target functions inside is used, where the label of whether clients have higher education is 0 and 1.

Then we encode data in three different ways: using Contrastive Learning for Event Sequences, Random encoder, Aggregation baseline.

4.2. CoLES

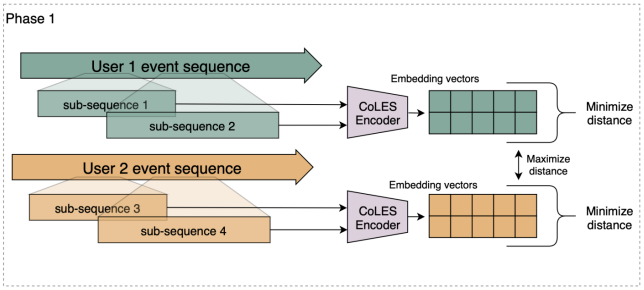

Figure 1. General framework. Phase 1: Self-supervised training.

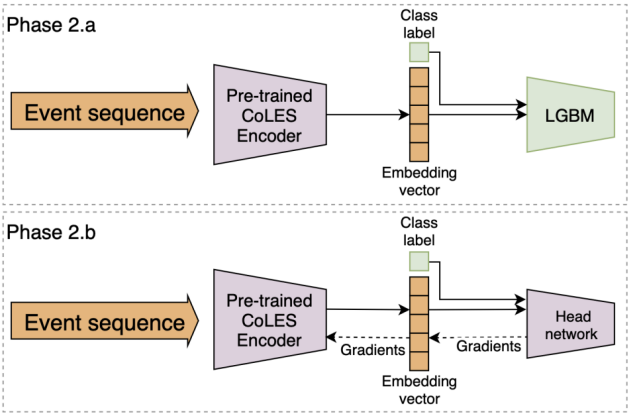

Figure 2. General framework. Phase 2.a Self-supervised embeddings as features for supevised model. Phase 2.b: Pre-trained encoder fine-tuning.

CoLES uses a novel augmentation algorithm, which generates sub-sequences of observed event sequences and uses them as different high-dimensional views of the object (sequence) for contrastive learning. Contrastive learning aims to learn a representation $x \to M(x)$, which brings positive pairs, i.e. semantically similar objects, closer to each other in the embedding space, while negative pairs, i.e. dissimilar objects, further apart. The proposed generative process is specifically designed to address the observed interleaved periodicity in financial transaction event sequences, which is the primary application of our method. Representations

learnt by CoLES can be used as feature vectors in supervised domain-related tasks, e.g. fraud detection or scoring tasks based on transaction history, or they can be fine-tuned for out-of-domain tasks.

The quality of representations can be examined by downstream tasks in the two ways: (1) c_e can be used as a feature vector for a task–specific model (see Figure 2, Phase 2a), (2) encoder M can also be (jointly) fine-tuned (see Figure 2, Phase 2b).

Encoder architecture

The composite encoder model trained in an end-to-end manner to minimize the contrastive loss. The event encoder takes a set of attributes of each event and outputs its intermediate representation in R^d. This encoder consists of several linear layers, for embedding onehot encoded categorical attributes, and batch normalization layers, applied to numerical attributes of events. Outputs of these layers are concatenated to produce the event embedding The concatenated to produce the event embedding. The sequence encoder takes the intermediate representations of the events and outputs the representation of their sequence up to the time. In our experiments, we use GRU, [5], a recurrent network which demonstrates robust performance on the sequential data

4.3. Random encoder

This is our first baseline model, which we use to evaluate the results of the CoLES. The network architecture is the same as CoLEs but with random weights. The CoLES results are supposed to be at least as good as the random encoder results.

4.4. Aggregation baseline

This is our second baseline model. The categorical features are arranged in One-Hot Encoder style and the numerical features are arranged in the resulting columns. But the significant feature of the method is that the order of the transactions is not taken into account. The CoLES results also supposed to be better.

5. Experiments and Results

We provided experiments for 3 encoders on two datasets. 6 experiments at total. We used Random Forest Classifier as a classification model. 20% of the dataset was taken for testing. The hyperparameters were taken from the article [1]. The following two tables show the results for different encoders in five different metrics (accuracy score, precision score,f1 score,Recall score,roc auc score). Let's define accuracy score, if $\hat{y}_i$ is the predicted value of the i-th sample and is the corresponding true value, then the fraction of correct predictions over n samples is defined as

$$accuracy(y, \hat{y}_i) = \frac{1}{n}\sum_{n=1}^{n} 1(\hat{y}_i = y_i)$$

Definition of F1 score

$$F1 = \frac{2(precision \cdot recall)}{precision + recall}$$

.

Definition of precision score

$$precision = \frac{Truepositive}{Truepositive + Falsepositive}$$

.

Definition of F1 score

$$precision = \frac{Truepositive}{Truepositive + FalseNegative}$$

.

And "roc auc score" - Area Under the Receiver Operating Characteristic Curve.

The Table 1 shows the results for the Transaction dataset.

Table 1. Evaluation of the quality of forecasts for different encoders for Transaction dataset

ENCODER	AC	PR	F1	R	ROC AUC
COLES	0.76	0.8	0.85	0.91	0.64
RANDOM ENC	0.73	0.74	0.84	0.97	0.5
AGG BASELINE	0.78	0.79	0.86	0.95	0.62

Table 2. Evaluation of the quality of forecasts for different encoders for Clickstream dataset

ENCODER	AC	PR	F1	R	ROC AUC
COLES	0.72	0.72	0.84	0.99	0.5
RANDOM ENC	0.69	0.72	0.81	0.92	0.49
AGG BASELINE	0.64	0.73	0.76	0.8	0.51

The Table 2 shows the results for the Clickstream dataset. From Table 1 we can see that results for CoLES encoder is better than for Random encoder. Also CoLES encoder has similar results as Aggregation baseline encoder. From Table 2 we can see that CoLES encoder has the best results in most metrics.

6. Conclusion

In our work, we have accomplished all the tasks we set out to do. We revised the article and reproduced its contents. We prepocessed the datasets in the format required for the task. We obtained embeddings for two different datasets using three encoders. We compared the embeddings on the classification problem to a random forest model with a tagged formation and proved that CoLES got better results compared to the baseline models.

References

Babaev, D., Ovsov, N., Kireev, I., Ivanova, M., Gusev, G., Nazarov, I., and Tuzhilin, A. Coles: Contrastive learning for event sequences with self-supervision. In *Proceedings of the 2022 International Conference on Management of Data*, pp. 1190–1199, 2022.

Dmitrii Babaev, Maxim Savchenko, A. T. and Umerenkov, D. Applying deep learning to credit loan applications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 2019.

Florea, A. R. and Roman, M. Artificial neural networks applied for predicting and explaining the education level of twitter users. volume 11, pp. 1–12. Springer, 2021.

Gomzin, A., L. A. S. V. . T. D. Detection of author's educational level and age based on comments analysis. 2018.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pp. 5628–5637. PMLR, 2019.

Science(ODS.ai), O. D. Data fusion contest 2022. education. URL https://ods.ai/competitions/data-fusion2022-education/dataset.

Tobback, M. Retail credit scoring using fine-grained payment data. In *Journal of The Royal Statistical Society Series A-statistics in Society*, pp. 1227–1246, 2019.

A. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Gleb Mazanov (20% of work)

- Coding the main algorithm
- Experimenting with model parameters on clickstreams dataset
- Preparing the Sections Introduction, Random encoder, Aggregation baseline, Conslusion and the main structure of this report

Nikolay Kotoyants (20% of work)

- Coding the main algorithm
- Experimenting with model parameters on transactions dataset
- Preparing the Section CoLES of this report

Ivan Gurev (20% of work)

- Coding the preprocessing of the datasets
- Preparing the GitHub Repo
- Preparing the Sections Abstract and Problem statement of this report

Anna Iliushina (20% of work)

- Reviewing literate on the topic ([3] papers)
- Preparing the presentation
- Preparing the Sections Related work and Preprocessing of this report

Viacheslav Naumov (20% of work)

- Reviewing literate on the topic ([4] papers)
- Preparing the presentation
- Preparing the Section Experiments and Results of this report

B. Reproducibility checklist

Answer the questions of following reproducibility checklist. If necessary, you may leave a comment.

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: We used the off-the-shelf CoLES architecture to reproduce the result from the article, but we did the preprocessing of the data and evaluation of the classification results ourselves

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: All of this is described in *Problem statement* and *Algorithms and Models* sections

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: You can see it and the beginning of the report.

4. A complete description of the data collection process, including sample size, is included in the report.

   ☐ Yes.
   ☑ No.
   ☐ Not applicable.

   Students' comment: We use pre-collected dataset from source [6] and the process of data collection is not considered

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: Link[6] indicates the downloadable version of the data set on the competition site

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: You can see it at "Experiments and results"

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   ☐ Yes.
   ☐ No.
   ☑ Not applicable.

   Students' comment: We use predefined hyperparameters from the article and the ready code to reimplemented the results.

9. The exact number of evaluation runs is included.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: Provided experiments for 3 encoders on two datasets. 6 experiments at total

10. A description of how experiments have been conducted is included.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: Used Random Forest classifier and made 6 experiments.

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: You can see it at "Experiments and results"

12. Clearly defined error bars are included in the report.

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: You can see it at "Experiments and results"

13. A description of the computing infrastructure used is included in the report.

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: metric formulas are added and experiments are described in the report