Robust Sequence Embedding for Recommendation

Rongzhi Zhang[1], Shuzi Niu[2(✉)], and Yucheng Li[2]

[1] University of Chinese Academy of Sciences, Beijing, China
zhangrongzhi17@mails.ucas.edu.cn
[2] Institute of Software, Chinese Academy of Sciences, Beijing, China
{shuzi,yucheng}@iscas.ac.cn

Abstract. Sequential recommendation is a significant task that predicts the next items given user historical transaction sequences. It is often reduced to a multi-classification task with the historical sequence as the input, and the next item as the output class label. Sequence representation learning in the multi-classification task is of our main concern. The item frequency usually follows the long tail distribution in recommendation systems, which will lead to the imbalanced classification problem. This item imbalance poses a great challenge for sequence representation learning. In this paper, we propose a **Ro**bust **S**equence **E**mbedding method for the recommendation, RoSE for short. RoSE improves the recommendation performance from two perspectives. We propose a balanced k-plet sampling strategy to make each training batch balanced at the data level and propose the triplet constraint for each training sequence to make sure of balance and robust distribution in feature space at the algorithmic level. Comprehensive experiments are conducted on three benchmark datasets and RoSE shows promising results in the face of item imbalance.

Keywords: Sequence embedding · Imbalance · Recommendation systems

1 Introduction

Sequential recommendation is a fundamental problem in real world. It attempts to predict the next items based on user historical transaction sequences. At each time step, it treats the user sequences before this time step as input instances, and the current items as the output class labels. In this sense, sequential recommendation is reduced to a multi-class classification task at each time step.

Many sequential recommendation approaches [1,11,12] have been proposed, among which neural sequence models [1] are becoming the main stream. They roughly consist of the following two steps. First, recurrent hidden units or its variants are utilized to learn the users' sequential features from input sequence instances. Second, users' sequential features, or sequence representation, and items' features are combined to feed the final layer to determine which items will

G. Li et al. (Eds.): KSEM 2020, LNAI 12275, pp. 114–122, 2020.
https://doi.org/10.1007/978-3-030-55393-7_11

be bought next. In this paper, we mainly focus on the sequence representation learning in this classification task.

In the recommendation system, most items in the tail of the distribution are consumed only several times, which belong to the minority category. On the contrary, a few items are consumed many times, which belong to the majority category. Such sequential recommendation methods fail in face of the long tail distribution data.

In traditional classification task, sampling based methods [9] and cost sensitive learning methods [4] are two major kinds of techniques to deal with class imbalance problem. Therefore, both kinds of solutions are not able to be applied to sequence embedding in imbalanced classification tasks. How to learn robust sequence embedding against the item imbalance becomes our main concern.

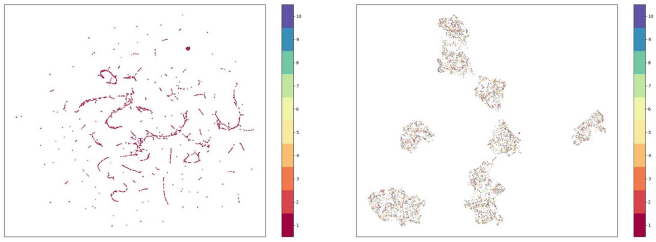
In this paper, we propose a **Ro**bust **S**equence **E**mbedding method against the item imbalance problem in the sequential recommendation task. From the data perspective, we propose a balanced k-plet sampling strategy to generate a training batch for both tasks in an alternative manner. It aims to uniformly sample sequences with balanced target class label. From the algorithmic perspective, we introduce a triplet constraint that intra-class sequence distance is larger than the inter-class sequence distance as the auxiliary task to the basic RNN, referred to as triplet constrained RNN. The stability of the triplet constraint contributes to balance representation between the input sequences with the same and different target items. We train the sequential recommendation task and its auxiliary task in a multi-task framework.

To investigate the effectiveness of sequence representations learned from RoSE, we conduct comprehensive experiments on three benchmark recommendation data sets. Experimental results that RoSE can achieve better performance compared with other baselines. Our main contributions are summarized as follows: (1) We first investigate how the sequence representation changes when the predicted item distribution is imbalanced. (2) To solve this problem from the algorithmic perspective, we propose triplet constrained RNN to help separate the minority class of sequences. (3) To solve this problem from the data perspective, we propose balanced k-plet sampling strategy to obtain a balanced training batch.

2 Related Work

2.1 Sequential Recommendation

Existing sequential recommendation methods capture the sequential pattern based on either Markov Chains or Neural Sequence Models. Markov Chain based models [11,12] utilize the sequence as a state transform process and capture the local feature through neighbor relationship. In light of the basic Markov Chain model [12], FPMC [11] puts forward a novel concept that leverages a personalized transmission graph matrix for each user. Neural Sequence models [1,5,8] can encode the whole item sequences while decode item representations with maximum probability in the next step.



(a) Pareto Distribution (b) Uniform Distribution

Fig. 1. UMAP of sequence representations learned from data with item frequency follows (a) Pareto Distribution and (b) Uniform Distribution respectively.

2.2 Learning from Imbalanced Data

Learning from imbalanced data is still a focus of intense research for decades [6]. Traditional studies solve this problem with data level, algorithm level and hybrid methods. **Data level methods** mainly concentrate on how to adjust the training set to balance the distribution through sampling. Oversampling and undersampling are two major kind of data-level methods. **Algorithm level methods** directly modify learning algorithms to alleviate bias towards the majority classes or correct the data distribution. The most popular branch is cost-sensitive learning. This kind of methods set a higher cost to the minority class and we boost its importance during the learning process. **Hybrid methods** combine the advantages of two previous groups. Quintuplet instance sampling and the associated triple-header hinge loss [3] are ensured to learn a robust and discriminative representation in standard deep learning framework for vision tasks.

Most existing studies on imbalanced data are for the typical classification task. In this paper, we put more emphasis on sequence representation learning from imbalanced data.

3 Sequence Embedding from Imbalanced Data

According to our survey, rare studies focus on how learned representations change in face of imbalanced data. Supervised embedding methods encode the label information in the input representation. Assuming that the item frequency follows nearly a uniform distribution, the label information does make the representation learning more discriminative. A more natural phenomenon is that the class frequency follows the long-tail distribution such as Pareto distribution in a social or scientific scenario. How the learned representations will be changed from the imbalanced data is attracting our attention.

To investigate this problem, we exploit the basic Recurrent Neural Network [1] with one layer of 10 hidden units to do comparative experiments on two synthetic datasets. Both datasets contain $11,000$ sequences with length 100, including 900 for training, 100 for validation and $10,000$ for test. Each item in

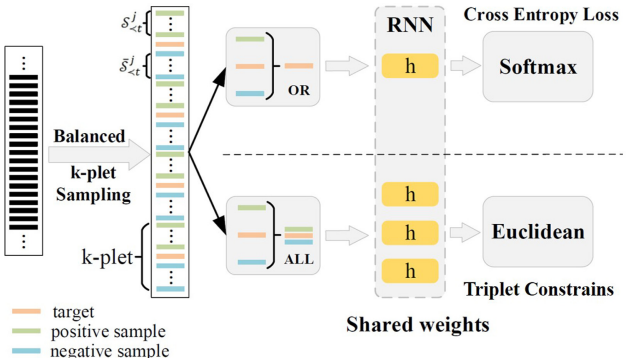


Fig. 2. Architecture of RoSE.

the sequence is sampled from $\{i\}_{i=1}^{10}$. The only difference between two datasets lies in the item frequency distribution. One is from the Pareto distribution, the other is from the Uniform distribution. We train the recurrent neural network model on both sets and obtain the sequence representations on test data shown in Fig. 1.

It is obvious that the sequence point separation is more clear in Fig. 1(b) than in Fig. 1(a). In this sense, sequence representations learned from uniform data are better than those learned from Pareto data. Another observation is that points labeled with item 1, 2, 3 almost dominate the whole dataset and other points even cannot be picked out in Fig. 1(a). These confused representations will lead to cascade errors in the following classifier layer. Thus the imbalanced data makes the sequence embedding mixed together and pose great challenge to learn robust sequence representation in this scenario.

4 Robust Sequence Embedding

To learn a robust sequence representation against item imbalance, we propose a **Ro**bust **S**equence **E**mbedding approach, referred to as RoSE. As shown in Fig. 2, the network architecture is divided into two components: (1)balanced k-plet sampling strategy; (2) Triplet Constrained RNN. First, we generate each training batch by balanced K-plet sampling strategy. Then, the training batch is fed to the triplet constrained RNN.

4.1 Balanced K-Plet Sampling Strategy

Balanced k-plet sampling strategy employs k-plet structure to make sure uniformly sample the training sequences for each item. Details are listed as follows. It generates a balanced training batch. The batch includes b balanced k-plets. Each balanced k-plet is consist of one pivot sequence, $k/2$ sequences with the

same label as the pivot sequence and $k/2$ sequences with different labels from the pivot.

To sample a balanced k-plet, we first randomly choose an item label i uniformly from all items. Then, the pivot sequence $S^u_{\prec t}$ and its $k/2$ sequences $\mathcal{S}^j_{\prec t}$ are randomly chosen from all the sequences labeled with i through inverted index. Finally, we uniformly choose $k/2$ items from all the items except i, randomly choose a sequence from all the sequences with each chosen label e through inverted index. Thus we obtain $k/2$ dissimilar sequences denoted as $\bar{\mathcal{S}}^u_{\prec t}$. So far a k-plet $(S^u_{\prec t}, \mathcal{S}^u_{\prec t}, \bar{\mathcal{S}}^u_{\prec t})$ is sampled, and a batch of such k-plets are sampled through our proposed sampling strategy. Different from traditional k-plet sampling strategies [7]. Our proposed sampling methods not only focuses on sequences with the same label, but also sequences with different labels in order to make training batch balanced.

4.2 Triplet Constrained RNN

Each item is represented as a one-hot vector in $\mathbb{R}^N$ is fed into the recurrent layer. For user's historical sequence $S^u_{\prec t}$, we use the Long Short Term Memory (LSTM) block as the basic unit to obtain the dynamic sequence embedding denoted as $h(S^u_{\prec t}; W)$ at time $t-1$. Parameters in this recurrent layer is denoted as W.

In order to predict which items will occur at time step t, the sequence embedding $h(S^u_{\prec t})$ at time step $t-1$ will be passed through the softmax layer with N output units. The i-th unit is represented as the probability that item i appears at time step t shown as $P(i|S^j_{\prec t}) = \frac{\exp(f_1(h(S^u_{\prec t};W),i;W_s))}{\sum_j \exp(f_1(h(S^u_{\prec t};W),j;W_s))}$, where W_s is the parameter of the output layer.

Through the balanced k-plet sampling strategy, we obtain the batch $\mathcal{B}$ and organize it into the training instances for sequential recommendation task denoted as $B_s = \{(S^u_{\prec t}, s^u_t)\}$. Each training instance is a input sequence $S^u_{\prec t}$ with its target item s^u_t. Suppose these instances are independent, sequential recommendation task is to optimize the cross entropy objective function in Eq. (1).

$$\mathcal{L}_1(B_s) = \frac{1}{|B_s|} \sum_{(S^j_{\prec t}, s^j_t) \in B_s} \log P(s^j_t | S^j_{\prec t}). \tag{1}$$

To make the learned sequence embedding robust against item imbalance, we propose a triplet constrain to the sequence embedding. Triplet constrain says that the distance between sequence a and b of the same label should be smaller than the distance between sequence a and c of different labels as $d(h(a;W), h(b;W)) < d(h(a;W), h(c;W))$. $d(\cdot,\cdot)$ is the distance between two sequences. This constrain will push the sequence embedding of a and b together, while pulling the sequence embedding of a apart from c. In other words, the explicit separation constrain solves the problem that sequence representations are mixed together in face of item imbalance. In this sense, the triplet constrain will make the sequence embedding robust.

There are various ways to define the sequence distance, such as Euclidean distance and cosine distance so on. However, whether the sequence distance is

suitable for this scenario remains unknown. Therefore, we learn the sequence distance measure as the auxiliary task. First, we organize the training sequence batch $\mathcal{B}$ into a triplet set B_m, and $B_m = \{(S^j_{\prec t}, C^p_{\prec t_c}, \bar{C}^n_{\prec t_{\bar{c}}}) | C^p_{\prec t_c} \in \mathcal{S}^j_{\prec t}, \bar{C}^n_{\prec t_{\bar{c}}} \in \bar{\mathcal{S}}^j_{\prec t}, (S^j_{\prec t}, \mathcal{S}^j_{\prec t}, \bar{\mathcal{S}}^j_{\prec t}) \in \mathcal{B}\}$. For each triplet $(S^j_{\prec t}, C^p_{\prec t_c}, \bar{C}^n_{\prec t_{\bar{c}}})$, there is a pivot sequence $S^j_{\prec t}$, sequence with the same label as $S^j_{\prec t}$ is the positive sample $C^p_{\prec t_c}$, and sequence with a different label is the negative sample $\bar{C}^n_{\prec t_{\bar{c}}}$. Then we use a ratio measure [2] to classify the samples into two class. In order to satisfy the triplet constrain, mean square error is defined as the loss function in Eq. (2). The loss aims to maximize the probability of positive samples in local neighborhood.

$$\begin{aligned} \mathcal{L}_2(B_m) &= \sum_{(x,x^+,x^-)\in B_m} \|(d_+, d_- - 1)\|_2^2 = const * d_+ \\ \text{where, } d_+ &= \frac{e^{\|h(x)-h(x^+)\|_2}}{e^{\|h(x)-h(x^+)\|_2} + e^{\|h(x)-h(x^-)\|_2}}, \\ d_- &= \frac{e^{\|h(x)-h(x^-)\|_2}}{e^{\|h(x)-h(x^-)\|_2} + e^{\|h(x)-h(x^+)\|_2}} \end{aligned} \tag{2}$$

Finally, we learn both sequential recommendation task and the auxiliary task satisfying these triplet constrains in a multi-task framework simultaneously. Here we define the whole network structure as the Triplet Constrained RNN. We share the weight W of two tasks in the hidden layer and train two tasks by updating the weight matrix alternately. We use the training parameter α to choose the order.

5 Experiment

To investigate the proposed RoSE, we conduct comprehensive experiments on three benchmark datasets.

5.1 Experimental Setting

We conduct the experiment on three benchmark dataset, MovieLens-1M, Tmall and Amazon Movies. In our experiments, MovieLens-1M is a subset of the Movielens dataset with $6.0K$ users, $3.7K$ movies and $1.0M$ ratings. Amazon-Movies has $3.2K$ users and $24.3K$ items, and $451.8K$ ratings. Tmall is a user-purchase dataset obtained from IJCAI 2015 competition which has $182.8K$ transactions, $0.8K$ user and $4.5K$ brands. We remove users with the number of ratings less than 50 in all datasets.

We compare our model with the following baselines. **POP**: The most popular items are recommended. **UKNN**: It predicts the next items based on consumed items of the target user's k neighbors, where user neighbors are defined based on cosine distance or Euclidean distance. **BPRMF** [10]: It is a matrix factorization method and directly optimize the ranking loss. **FPMC** [11]: Factorized Personalized Markov Chains stacks state transition matrixes into a cube which satisfies the Markov property. **RNN** [1]: Recurrent Neural Network encodes transactions as item sequences into recurrent neural network and predicts the next behavior.

Table 1. Performance comparison on three datasets.

Dataset	Metric@10	POP	UKNN	BPRMF	FPMC	RNN	RoSE
MovieLens-1M	sps	0.0502	0.1237	0.0140	0.0154	0.2788	**0.2866**
	recall	0.0391	0.0552	0.0105	0.0468	0.0756	**0.0764**
	precision	0.2360	0.2305	0.0699	0.2120	0.3078	**0.3109**
	F1-measure	0.0671	0.0889	0.0183	0.0767	0.1214	**0.1227**
	NDCG	0.2450	0.2370	0.0790	0.2228	0.3229	**0.3263**
	U_cov	0.7077	0.7546	0.4230	0.7657	**0.8766**	0.8764
Tmall	sps	0.0465	0.0465	0.0100	0.0235	0.0233	**0.0581**
	recall	0.0125	0.0230	0.0042	0.0076	0.0207	**0.0265**
	precision	**0.0860**	0.0651	0.0349	0.0209	0.0605	0.0825
	F1-measure	0.0218	0.0339	0.0074	0.0111	0.0308	**0.0401**
	NDCG	0.0949	0.0718	0.0388	0.0216	0.0765	**0.0983**
	U_cov	0.5233	0.4535	0.2674	0.1860	0.4186	**0.5465**
Amazon Movies	sps	0.0065	0.0265	0.0038	0.0220	0.0228	**0.0531**
	recall	0.0078	**0.0141**	0.0031	0.0083	0.0109	0.0137
	precision	0.0498	0.0701	0.0199	0.0476	0.0649	**0.0761**
	F1-measure	0.0135	**0.0234**	0.0053	0.0141	0.0186	0.0232
	NDCG	0.0508	0.0713	0.0222	0.0504	0.0677	**0.0835**
	U_cov	0.3138	**0.4431**	0.1630	0.2810	0.3399	0.4285

5.2 Performance on Sequential Recommendation

To obtain a whole picture of RoSE's effectiveness, we compare the performance of RoSE with baselines' in terms of short term prediction, long term prediction and generalization. Comparison results on three benchmark datasets are shown in Table 1. **Short Term Prediction**. There are two main observations. (1) Compared with the general recommender, the sps improvement of sequential recommender is over one time on both MovieLens and Amazon Movies. (2) RoSE has a significant improvement in three datasets and shows its remarkable performance in short-term recommendation. For example, RoSE outperforms the best baseline method UKNN by 24.9% on Tmall and 100% on Amazon Movies. Both observations indicate that our model obtains better sequence embedding that makes the prediction results better. **Long Term Prediction**. We observe that UKNN seems to be the best baseline in terms of recall, precision, F1-measure and ndcg. RoSE performs the best among all the baselines on three datasets. For example, the F1-measure and NDCG improvements on Tmall are 30.2% and 28.5% compared with RNN. Generally, RoSE achieves an excellent performance in long term prediction. **Generalization**. We conclude the following two points. (1) We can see that the user coverage of RoSE outperforms the best baseline on Tmall by 4.3% and similar results are obtained on MovieLens. (2) However, things are different on Amazon Movies, and UKNN achieves better performance than RoSE. The reason may lies in that we use the sample of Amazon Movies with uniform distribution and the cosine distance used for uniform

data has strong capacity to represent the global interests. For most datasets, the generalization performance of RoSE achieves the best.

In general, all these observations from Table 1 show RoSE achieves highly superiority on three datasets especially for sps. Meanwhile, these results conform that the learned sequence representations are more discriminative to help separate the predicted items better, which lead to the performance increase in terms of short term and long term prediction.

6 Conclusion

In this paper, we first investigate how sequence representations change with the item imbalance. With the advent of item imbalance, sequence embeddings from minority classes will be mixed together. To solve this problem, we propose a robust representation learning framework RoSE which is composed of balanced k-plet sampling strategy and triplet constrained RNN. We generate a balanced mini-batch through balanced k-plet sampling strategy and define a triplet constrain. The triplet constrains are introduced as an auxiliary task and multi-task framework make representations with better distribution. Experimental results on three benchmark datasets show that our model outperforms baselines against the item imbalance problem.

References

1. Devooght, R., Bersini, H.: Collaborative filtering with recurrent neural networks. arXiv preprint arXiv:1608.07400 (2016)
2. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Feragen, A., Pelillo, M., Loog, M. (eds.) SIMBAD 2015. LNCS, vol. 9370, pp. 84–92. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24261-3_7
3. Huang, C., Li, Y., Change Loy, C., Tang, X.: Learning deep representation for imbalanced classification (2016)
4. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans. Neural Netw. Learn. Syst. **29**(8), 3573–3587 (2018)
5. Ko, Y.J., Maystre, L., Grossglauser, M.: Collaborative recurrent neural networks for dynamic recommender systems. In: JMLR: Workshop and Conference Proceedings, vol. 63 (2016)
6. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell. **5**(4), 221–232 (2016). https://doi.org/10.1007/s13748-016-0094-0
7. Lin, X., Niu, S., Wang, Y., Li, Y.: K-plet recurrent neural networks for sequential recommendation. In: SIGIR, ACM (2018)
8. Niu, S., Zhang, R.: Collaborative sequence prediction for sequential recommender. In: CIKM. ACM (2017)
9. Rendle, S., Freudenthaler, C.: Improving pairwise learning for item recommendation from implicit feedback. In: WSDM (2014)
10. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2009)

11. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: WWW (2010)
12. Zimdars, A., Chickering, D.M., Meek, C.: Using temporal data for making recommendations. In: UAI (2001)