Improved Session based Recommendation using Graph-based Item Embedding

Madiraju Srilakshmi
Indian Institute of Technology
Kharagpur, India
sreelakshmi@iitkgp.ac.in

Gourab Chowdhury
Indian Institute of Technology
Kharagpur, India
gourab@iitkgp.ac.in

Sudeshna Sarkar
Indian Institute of Technology
Kharagpur, India
sudeshna@cse.iitkgp.ac.in

ABSTRACT

Recommendation systems greatly benefit from good user and item representations. We consider the task of session-based recommendation where users are anonymous and thus the representation of items plays a key role in the effectiveness of recommendations. We propose an enhanced item embedding method that captures the relationships between items by taking into account item pair co-occurrence across sessions using a graph-based embedding approach. Specifically, we construct a heterogeneous graph of items and sessions, where two items are connected through a session if they co-occurred within the session. We learn embeddings for items and sessions by considering the triplets of the form (Previous item, Session, Next item) using a knowledge graph embedding approach called TransE. We use a neural attention based Recommendation model and apply it on two benchmark datasets. We show that the learned item representations improve the performance of our recommendation model and achieve state-of-the-art results. Combining the graph-based embedding with standard Prod2Vec embedding gives further improvement.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Information Systems → Information retrieval**; *Recommender Systems*; • **Computing methodologies** → Neural networks.

KEYWORDS

Session-based Recommendation, Sequential Behaviors, Item Embedding, Knowledge Graph Embedding

ACM Reference Format:
Madiraju Srilakshmi, Gourab Chowdhury, and Sudeshna Sarkar. 2020. Improved Session based Recommendation using Graph-based Item Embedding . In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'20)*. ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

An e-commerce portal typically hosts a large number of products, and recommendation systems provide a very useful service to the users by recommending products. We consider the task of Session-based Recommendation Systems (SRS) where users are unknown and we only have access to the user interaction sequences organized by sessions. The task is to predict a user's next action based on his/her behavior in the current session. The limited data available makes this task challenging.

Recurrent neural networks (GRU and LSTM) based models [5, 14, 4] have been effectively used for SRS and have been shown to outperform traditional collaborative filtering approaches. Hidasi et al. [5] propose to use RNN for this task and Tan et al. [14] enhances this model by using data augmentation and temporal shift of user behaviour. Gui et al. [4] denotes items with Prod2Vec embeddings [3] and utilize them to make recommendations with RNN. Further, augmenting these models with attention mechanism improves the performance significantly [9, 7].

Though the above methods show promising improvements over traditional methods, they have not emphasized on effective item representations. Most of these methods use 1-to-n encoding [5, 14] or add an embedding layer in their neural architecture [7, 9]. One-hot encoding takes more time to optimize and is not scalable for datasets with millions of items. On the other hand, the addition of an extra embedding layer that jointly trained with the whole network may lead to performance deterioration due to the acquirement of bad features [4, 5]. More importantly, neither of these methods capture the sequential similarities of items implied in the session sequences effectively. Gui et al. [4] denotes items with Prod2Vec embedding [3] to overcome this limitation. However, Prod2Vec ignores the item co-occurrences across different sessions. For instance, they do not capture the relationship between two items which co-occur with the same item in separate sessions.

To address this limitation, we propose a graph-based item embedding method which captures sequential behaviour of items within a session as well as item pair co-occurrence across sessions. For this, we construct a heterogeneous directed graph of items and sessions. In this graph, two items (item_ids) are connected through the session (session_id) in which they co-occur. We find the embedding of items and sessions by considering the triples of the form (Previous item, Session, Next item). We use a Neural Attention based Recommendation model with Embedding (NARE) and apply it on two benchmark datasets. The results reveal that the graph-based item embedding enhances the performance of our simple recommendation model and achieves comparable results with the state-of-the-art methods. Moreover, we find that combining the graph-based embedding with Prod2Vec embedding gives further improvement.

The embedding can be used with other recommendation models and is shown to improve the performance. We apply the graph-based embeddings in a popular baseline method called NARM [7] and achieve better results than standard NARM.

SIGIR eCom'20, July 30, 2020, Virtual Event, China

2 RELATED WORK

2.1 Session-based Recommendations using RNN

Hidasi et al. [5] used Recurrent Neural Networks for recommendation and achieved significant improvements over traditional methods. Tan et al. [14] improved this approach by adapting to data augmentation technique which splits the session into many subsessions for training. Gui et al [4] used Prod2Vec embeddings to represent items and utilize them with RNN based model.

NARM [7] used hybrid encoder with a bi-linear matching method for recommendation. The global encoder encodes the full session information using GRU whereas the local encoder is expected to learn user's main purpose in the session using GRU with attention. The output embeddings of both are combined to represent the session. The scores of items are computed by using a bi-linear similarity function between learned session embedding and item embedding.

STAMP [9] recommends items using a tri-linear composition method. The scores of items are computed as inner product of item embedding and weighted user embedding. The user embedding is a bi-linear composition of over-all preferences (Average of all item embeddings in the session) and current interest (last-clicked item embedding). The attention weights are learned using a simple feed forward network on session sequences.

SR-GNN [17] models the session sequences as graph and learn item and session representations by using Graph Neural Networks (GNN). The item scores are computed by multiplying the item embedding with session embedding.

We use a standard LSTM with attention model for recommendation and we represent items with the learned graph-embeddings.

2.2 Knowledge Graph Embedding based Recommendations

In recent times, knowledge graph embeddings have shown to be beneficial for recommendation systems. The basic idea is to represent the available data in the form of a graph, learn embeddings for entities using Knowledge graph embedding methods [16] and incorporate them into recommendation.

Collaborative Knowledge base Embedding (CKE) uses TransR [8] to learn items structural representations and combine them with visual and textual embeddings. Deep Knowledge-aware Network (DKN) [15] learns entity embeddings using TransD [6] and and designs a CNN framework by combining them with word embeddings for news recommendation. Ai et al. [1] learn embedding of users and items by the method of TransE [2] and the recommendation is based on user-item similarity score in the projected space.

The major difference between our work and previous works is graph representation and recommendation framework. Since the users are anonymous in our setting, the graph contains only sessions and items. We jointly learn the embeddings for sessions and items using TransE [2]. We choose TransE because of its simplicity and efficiency in modelling multi-relational data [10]. We use the learned item embeddings for recommendation using LSTM with attention framework.

2.3 Prod2Vec Embedding

Given a sequence of items in a session, Prod2Vec [3] objective is to find a d-dimensional representation such that the items which co-occur together in the session are close in the resulting vector space. The embeddings are learned by minimizing the weighted cross entropy between the empirical and the modelled conditional distributions of context items given the target items.

We compare graph-based embeddings with Prod2Vec embeddings in our recommendation model.

3 METHOD

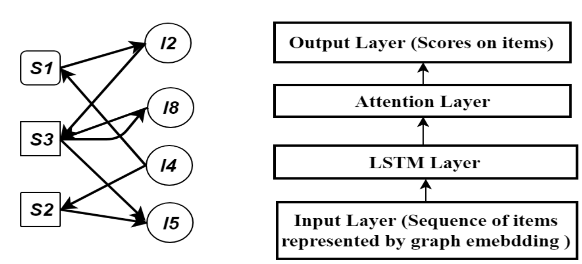
In this section, we first formally define the task and present an overview of the proposed model. We then describe the main components of our approach in detail, i.e. graph-based item embedding generation and recommendation framework using attentive LSTM model.

3.1 Problem Definition

Session based Recommendation is the task of predicting a user's next interaction based on his/her behaviour in the current session and given all earlier session sequences. Let $I = \{i_1, i_2, \ldots, i_m\}$ be the set of items (item_ids) and $S = \{s_1, s_2, \ldots, s_n\}$ be the sessions(session_ids) available. Let S_{seq} is the set of all train session sequences. Each session s_i has a sequence of user interactions $(x_1^i, x_2^i, \ldots, x_{T_i}^i)$ where x_t^i denotes the item interacted at time-step t in session s_i. Given all train session sequences (S_{seq}) and a target session with the user interactions $(x_1^i, x_2^i, \ldots, x_t^i)$, the task is to predict the next possible interaction x_{t+1}^i.

3.2 Overview

We create a graph-based item embedding from a graph created by items that co-occur in a session. This item representation is used in an attention based LSTM model. The output of the model is a ranking list over all items from which top-k items are recommended. Figure 1 gives an overview of our approach and the details are presented in Subsections 3.3 and 3.4.



(a) Graph representation of items and sessions

(b) General framework of our recommendation model

Figure 1: Overview of our approach: (a) Graph representation of session sequences;(b) Recommendation framework and data-flow of our system

3.3 Item embedding generation

The representation of items have a significant impact on the outcome of the recommendation model. We aim to generate item representations that can capture the relationships between items by considering item pair sequential behavior as well as co-occurrences across sessions.

We model session sequences in the form of a directed heterogeneous graph. The nodes of the graph are sessions (session_ids) and items (item_ids). Two items (item_ids) are connected through the session (session_id) in which they co-occur. The direction of the edges is based on the order of the occurrence of the items in the session so as to capture the sequence of item interaction. The graph is formed as set of triplets $T = (i_1, s, i_2)$, where i_1, i_2 are item nodes and s is the session node through which they are connected.

For example, consider sessions $s_1 = \{i_4, i_2\}, s_2 = \{i_4, i_5\}, s_3 = \{i_2, i_8, i_5\}$. The triplets for this graph are as follows: $T = \{(i_4, s_1, i_2), (i_4, s_2, i_5), (i_2, s_3, i_8), (i_2, s_3, i_5), (i_8, s_3, i_5)\}$. The graph representation for the sessions is as shown in Figure 1. We observe that the items i_4 and i_8 occur in different sessions, but there is a path between them in the graph as they co-occur with the item i_2. Since these items are connected in the graph, the embedding may be able to capture the relationship between them.

After we obtain the triplets, we apply the method of TransE [2] to learn d-dimensional embeddings for items and sessions. To be compatible with the representation of TransE, we consider our item nodes as entities (I) and session nodes as relations (S). Given a triplet (i_1, s, i_2), the relation is interpreted as a translation vector s so that the learned entities i_1 and i_2 can be connected by s with minimal error, i.e. $i_1' + s' \approx i_2'$ when (i_1, s, i_2) holds, where i_1', i_2' and s' are the corresponding representation vectors of i_1, i_2 and s. Figure 2 gives an illustration of this approach. The embeddings are learned through Stochastic Gradient Descent (SGD) and TransE learning algorithm [2] is shown in Algorithm 1.

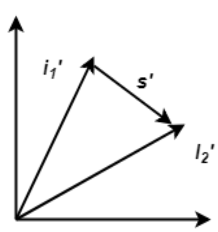


Figure 2: Items and session in lower dimensional space in *TransE*

3.4 Recommendation Framework

We use Long Short Term Memory model with attention mechanism for recommendation. Let h_t be the LSTM hidden unit and y_t be the output at t-th time step. For every session sequence $(x_1^i, x_2^i, \dots, x_t^i) \in S_{seq}$ where S_{seq} is the set of session sequences, we perform the following steps to train the LSTM model.

(1) Each item in the session sequence is represented with learned graph-based embedding.

$$v_t^i = Graph - based\ embedding(x_t^i) \tag{1}$$

Input: Training set $T = (i_1, s, i_2)$, Entity set I, Relation set S
Randomly initialize the embeddings for $s \in S, i \in I$
for (i_1, s, i_2) *in* T **do**
  Corrupt the triplet by changing i_1 or i_2 and add it to T
  $T \leftarrow T \cup (\hat{i}_1, s, \hat{i}_2)$
  Update embeddings w.r.t the loss L i.e.
  $\sum_{(\hat{i}_1,s,\hat{i}_2),(i_1,s,i_2)\in T} \Delta[\gamma + d(i_1 + s, i_2) - d(\hat{i}_1 + s, \hat{i}_2)]$
  Dissimilarity measure d can be either the $L1$ or the $L2$ norm
end

Algorithm 1: Learning TransE embeddings

(2) We update each hidden state h_t by the previous hidden state h_{t-1} and the current item embedding v_t^i.

$$h_t = LSTM(h_{t-1}, v_t^i) \tag{2}$$

(3) The output of hidden states $(H = h_1, h_2, \dots, h_t)$ is given as input to attention layer to find attention weights and the weights for each time-step are learnt i.e. $A = (a_1, a_2, \dots, a_t)$.

$$A = softmax(w^T * \tanh(H)) \tag{3}$$

(4) We input the weighted sum of the hidden states (M) into a dense layer (D). The number of neurons in this layer is equal to the total number of unique items and softmax is used as the activation function.

$$M = A^T * H \tag{4}$$

$$p_t \mid v^i_{1\le i\le t-1} = softmax(D[M]) \tag{5}$$

(5) The loss function used for optimization is the categorical cross-entropy loss. The loss (L) is calculated as shown below.

$$L = \sum_{c=1}^{m} y_c \log(p_c) \tag{6}$$

where m is the number of unique items, $Y \in \mathbb{R}^m$ is is the one-hot represented ground truth and $P \in \mathbb{R}^m$ is the estimated probability for each class by softmax.

The trained LSTM with attention model is used for the next-item prediction. Each item in the target session sequence is represented by the graph-based embedding. If any item in the sequence is not seen in the training set, we represent it with the zeros.

We input the sequence into the trained LSTM model. The outputs of the final layer in the model are the probabilities of all items in the catalogue. The item with the highest probability is recommended to the user. This architecture is illustrated in Figure 3.

4 EXPERIMENTS

In this section, we first describe the details of the dataset, the methods used for comparison and the evaluation metrics employed in our experiments. Then we present the results of the proposed framework to show the role of the item embedding.

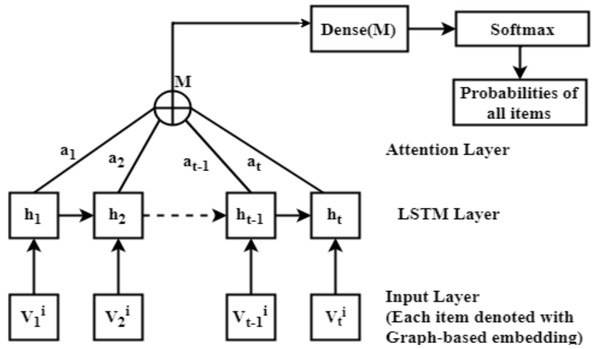


Figure 3: LSTM model with attention for recommendation

Table 1: Data statistics after pre-processing

Statistics	Yoochose 1/64	Diginetica
# Total clicks	557248	982961
# Train sequences	369859	719470
# Test sequences	55898	60858
# Total items	16766	43097
Average length	6.16	5.12

4.1 Dataset Details

We evaluate the proposed model on Yoochoose 1/64 [1] dataset, which was released by RecSys' 15 Challenge and Diginetica [2] dataset, which was obtained from CIKM Cup 2016, where only its trasactional data is used. We use same preprocessing method and evaluation metrics as used [9, 7].

Yoochoose dataset consists of six months of click-stream data, where the training set only contains session events. Following [9, 7], we remove the sessions of length 1 and items that appear less than 5 times. The test set of Yoochoose data consists of the sessions of subsequent days with respect to the training set, whereas Diginitica test set contains the sessions of subsequent week. We filter out clicks (items) from the test set that did not appear in the training set. We use the recent 1/64 fraction of training sequences of Yoochoose as training set. The statistics of the datasets is shown in Table 1.

4.2 Evaluation Metrics and Experimental Setup

4.2.1 Evaluation Metrics. We use the following metrics to evaluate our model.

- Recall@20: It is the proportion of cases when the actual item is among the top-20 recommended items among all test cases.
- Mean Reciprocal Rank@20: It is the mean of reciprocal ranks of correctly recommended items. If the actual item is not in the top-20 recommended items, then the rank is set to zero.

4.2.2 Experimental Setup. Our model uses 100-dimensional embeddings to represent items in Prod2Vec [3] as well as graph-based embedding. We use Adam optimizer, and the learning rate is set to 0.001. The batch size is set at 512. There is a dropout layer in between the LSTM layer and the attention layer with 25% dropout. We use one LSTM layer with 100 hidden units.

[1]https://2015.recsyschallenge.com/challege.html
[2]http://cikm2016.cs.iupui.edu/cikm-cup

Table 2: Performance comparison of our approach with baseline methods

Method	Yoochoose		Diginitica	
	Recall@20	MRR@20	Recall@20	MRR@20
POP	6.71	1.65	0.89	0.20
S-POP	30.44	21.81	21.06	13.68
Item-KNN	51.60	15.01	35.75	11.57
BPR-MF	31.31	22.89	5.24	1.98
FPMC	45.62	29.00	26.53	6.95
GRU4Rec	60.64	28.76	29.45	8.33
NARM	68.32	28.63	49.70	16.17
STAMP	68.74	29.67	45.64	14.32
SR-GNN	70.57	30.94	50.73	17.59
NARE (Prod2Vec)	69.15	29.5	43.97	13.72
NARE (Graph-based)	69.71	30.56	46.52	15.02
NARE (Combined)	**71.25**	**30.98**	47.65	16.21
NARM (Prod2Vec)	68.31	28.70	50.90	18.17
NARM (Graph-based)	69.54	29.25	51.96	18.54
NARM (Combined)	69.92	29.44	**52.68**	**18.57**

4.3 Models Compared

We compare our results with traditional recommendation methods (i.e. POP, S-POP, Item-KNN [13], BPR-MF [12] and FPMC [11]), RNN based method (i.e. GRU4Rec [5]) and attention based methods (i.e. STAMP [9], NARM [7] for Session-based Recommendation. We also compare our result with SR-GNN [17] which is the the state-of-the-art method for SRS.

We use NARE with different embeddings. We study the use of pre-trained item embeddings as input to a recommendation system. We consider our proposed graph-based embedding, Prod2Vec embedding and a combination of the two. Further, we use these embeddings as input to the NARM [7] system. The random initials in NARM are replaced with our embeddings.

4.4 Results and Analysis

The results on all baselines and our methods are listed in Table 2 in terms of Recall@20 and MRR@20 on both Yoochoose and Diginetica datasets.

In our experiments, NARE with the combined embedding achieves the best result on Yoochoose dataset while NARM with the combined embeddings produces the best result on Diginetica dataset.

We conclude that the richer item representation from the pre-trained embedding helps improving the performance of the recommendation systems. Prod2Vec embeddings are limited because they can capture sequential behaviour only. But graph-based embeddings are more effective because the items in different sessions are also connected indirectly through other sessions in the graph. This enables the learned embeddings to capture their similarity across sessions. A concatenation of both the embeddings yields even better results since the concatenated embedding captures both perspectives i.e., sequential behaviour of items as well as co-occurrence of items across sessions.

4.4.1 Computational Complexity of embeddings generation. TransE is very simple and efficient knowledge graph embedding technique. The time complexity of TransE is $O(d)$, where d is the embedding dimension. The space complexity of TransE is $O(nd + md)$, where n and m are the number of items and sessions respectively.

The number of triplets that can be generated from a session of length n is $\binom{n}{2}$. If the session is too lengthy, it can be divided into multiple sub sessions based on time of interaction and as a result each session may be of smaller length. For example $s = \{i_1, i_2, ..., i_{100}\}$ be a user session and the items are ordered by user interaction time. It is very uncommon that i_1 is related to i_{100}. Therefore, we can divide this session into multiple sub sessions of reasonable size (Let us say 20) i.e., $s_1 = \{i_1, i_2, ..., i_{20}\}$, $s_2 = \{i_{21}, i_{22}, ..., i_{40}\}$ etc. Now, triplet generation is also simple and efficient.

5 CONCLUSION

In this paper, we have presented a graph-based representation of user activities in sessions and have shown the effectiveness of graph-based item embedding approach in recommendation. The learned graph-based item embeddings capture more implicit connections between items which helps to improve the quality of recommendation. These embeddings can be used with other recommendation frameworks also. We have demonstrated this by incorporating the embeddings as input to NARM model.

ACKNOWLEDGEMENT

This research was partially supported by a grant from Capillary Technologies International Pte Limited. We thank Dr. Subrat Panda and Mr. Biswa Singh who provided insight and expertise that greatly assisted the research.

REFERENCES

[1] Ai, Qingyao et al. "Learning heterogeneous knowledge base embeddings for explainable recommendation". In: *Algorithms* 11.9 (2018), p. 137.

[2] Bordes, Antoine et al. "Translating embeddings for modeling multi-relational data". In: *Advances in neural information processing systems*. 2013, pp. 2787–2795.

[3] Grbovic, Mihajlo et al. "E-commerce in your inbox: Product recommendations at scale". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015, pp. 1809–1818.

[4] Gui, Yue and Xu, Zhi. "Training Recurrent Neural Network on Distributed Representation Space for Session-based Recommendation". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–6.

[5] Hidasi, Balázs et al. "Session-based recommendations with recurrent neural networks". In: *arXiv preprint arXiv:1511.06939* (2015).

[6] Ji, Guoliang et al. "Knowledge Graph Embedding via Dynamic Mapping Matrix". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2015, pp. 687–696.

[7] Li, Jing et al. "Neural attentive session-based recommendation". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM. 2017, pp. 1419–1428.

[8] Lin, Yankai et al. "Learning entity and relation embeddings for knowledge graph completion". In: *Twenty-ninth AAAI conference on artificial intelligence*. 2015.

[9] Liu, Qiao et al. "STAMP: short-term attention/memory priority model for session-based recommendation". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 2018, pp. 1831–1839.

[10] Palumbo, Enrico et al. "Translational Models for Item Recommendation". In: *European Semantic Web Conference*. Springer. 2018, pp. 478–490.

[11] Rendle, Steffen, Freudenthaler, Christoph, and Schmidt-Thieme, Lars. "Factorizing personalized markov chains for next-basket recommendation". In: *Proceedings of the 19th international conference on World wide web*. ACM. 2010, pp. 811–820.

[12] Rendle, Steffen et al. "BPR: Bayesian personalized ranking from implicit feedback". In: *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press. 2009, pp. 452–461.

[13] Sarwar, Badrul et al. "Item-based collaborative filtering recommendation algorithms". In: *Proceedings of the 10th international conference on World Wide Web*. ACM. 2001, pp. 285–295.

[14] Tan, Yong Kiam, Xu, Xinxing, and Liu, Yong. "Improved recurrent neural networks for session-based recommendations". In: *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM. 2016, pp. 17–22.

[15] Wang, Hongwei et al. "DKN: Deep Knowledge-Aware Network for News Recommendation". In: *arXiv preprint arXiv:1801.08284* (2018).

[16] Wang, Quan et al. "Knowledge graph embedding: A survey of approaches and applications". In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743.

[17] Wu, Shu et al. "Session-based recommendation with graph neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 346–353.