Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks

Bartłomiej Twardowski
Warsaw University of Technology
Institute of Computer Science
Nowowiejska 15/19, 00-665 Warsaw, Poland
B.Twardowski@ii.pw.edu.pl

ABSTRACT

Preparing recommendations for unknown users or such that correctly respond to the short-term needs of a particular user is one of the fundamental problems for e-commerce. Most of the common Recommender Systems assume that user identification must be explicit. In this paper a Session-Aware Recommender System approach is presented where no straightforward user information is required. The recommendation process is based only on user activity within a single session, defined as a sequence of events. This information is incorporated in the recommendation process by explicit context modeling with factorization methods and a novel approach with Recurrent Neural Network (RNN). Compared to the session modeling approach, RNN directly models the dependency of user observed sequential behavior throughout its recurrent structure. The evaluation discusses the results based on sessions from real-life system with ephemeral items (identified only by the set of their attributes) for the task of top-n best recommendations.

Keywords

recommender system; matrix factorization; recurrent neural network; session-aware recommendations

1. INTRODUCTION

The Session-Aware Recommender System is a type of a Context-Aware Recommender Systems (CARS) where additional information about the user session is incorporated in the prediction process. Such systems can understand user short-term goals and fit better to the changing users behaviour in time. Particularly, in this work, the user is not identified at all, only characterized by the last actions in the current session.

The problem of session-aware recommendations can be even more troublesome, when the recommended items are ephemeral i.e. the item life-cycle is too short or the availability is too dynamic to identify it only by unique id, e.g. news, online auctions. In such settings, the Content-Based filtering is a standard technique. However, neglecting the session information results in loosing lots of information. In this paper novel ways to cope with ephemeral items and to recommend them in the right session context are presented.

The attractiveness of session-aware recommendations for online businesses can be confirmed by the fact that identification of the user can be hard and the same account can be shared among others, e.g. family members. For common e-commerce more than half (57.06%) of all sessions are non-logged users.[1] Only 2.53% of all sessions converts to transaction. Most sessions are *window-shopping* ones, where users are only looking for the product availability, price, and information. The 20.98% of all page views are interactions with the search engine. Out of all sessions, 35.80% used the search engine to find the right offer. User interactions with the search engine is the rich source of their preferences. Yet, the traditional recommender systems do not take this into account directly.

2. RELATED WORK

In most of RS, the information about the time of user actions is omitted (e.g. most of the Collaborative-Filtering algorithms) or is used directly in computing the cost of the model, e.g. in a form of time dependent bias [4]. Different approach is taken in Context-Aware Recommender Systems, where time can be used as a contextual variable. Although, time should be discretized, e.g. to time of a day, day of the week, etc.. Another possibility is to represent time dependencies between actions as a new variables in CARS models, e.g. after viewing particular items, which item was bought/seen next[3].

In work[5] an approach for detecting a topic of the user's session is proposed. The topic is described by the set of item attributes, as in this work. For catching the user's goal (topic of the session) a factored Markov Decision Process is used. In order to train such a model, a strong assumption of attribute independence is being made. This results in efficient optimization of many independent models.

Usage of RNN for session based recommendations was recently proposed in [2]. The GRU units are used to predict the next item in the session. The network input is one-hot-encoded current clicked item id. The output is scoring for fixed number of items. This paper uses similar evaluation

[1]Presented statistics are based on Polish e-marketplace allegro.pl for 3228 M page views sample in January of 2016, where 310 M sessions was identified by HTTP cookie or mobile device hash.

RecSys '16, September 15-19, 2016, Boston , MA, USA

DOI: http://dx.doi.org/10.1145/2959100.2959162

methodology, but only considering unseen items as valid recommendations.

3. USER SESSION DATA

User sessions in this work are defined as uninterrupted sequences of activity in the system. The session ends when the user is inactive for more than a predefined number of minutes [1]. This method of setting the session boundary allows to represent all user actions grouped in sessions: $S = \{s_1, \ldots, s_m\}$ where each session is represented as a set of events ordered in time: $s_m = \{e_m^{(1)}, \ldots, e_m^{(t)}\}$, where t is the time of the event occurrence in session m. In turn, each event is described by contextual information: $e_m^{(t)} \in C_1^E \times C_2^E \times \cdots \times C_k^E$, where number of attributes k depends on collected event $e_m^{(t)}$ type.

One of the event context information C_k^E is an association with item $I = \{i_1, \ldots, i_n\}$. Each item is described by a set of defined attributes $i_n \in C_1^I \times C_2^I \times \cdots \times C_p^I$, where the number of all the item attributes is p.

Item and Event Encoding

All methods presented in this work require items and events being represented by real-valued vectors. This is a well-known approach from Content-Base filtering, where every items data before being compared has to be vectorized first. There are two entities to encode items and events. For items, the encoding function $C_1^I \times C_2^I \times \cdots \times C_p^I \rightarrow \mathbf{x}_n$, $\mathbf{x}_n \in \mathbb{R}^{d_I}$ should exist, where d_I is the number of real-values in the encoded item representation. Similarly, the session event $C_1^E \times C_2^E \times \cdots \times C_k^E \rightarrow \mathbf{x}_e$, $\mathbf{x}_e \in \mathbb{R}^{d_E}$, where d_E is the dimension of the encoded event vector.

In this work a unified vectorization method for contextual information is assumed. If the context information is textual or can be treated as a bag-of-words (e.g. offer category path information) the data is one-hot encoded. To constrain the size of the output, the minimum term frequency and maximum vocabulary size are defined. Any categorical information are also one-hot encoded. Numeric attributes are scaled to $[0, 1]$. For events encoding, the encoded contextual information is concatenated with associated encoded item information.

4. SESSION-AWARE RECOMMENDATIONS

4.1 Session Modeling with Matrix Factorization

This method is a simplified version of Factorization Machines[7], where only interactions between variables of the session and the item are allowed for final estimation. The item is represented by the encoded, vectorized representation $\mathbf{x}_i$. The session vector $\mathbf{x_s}$ aggregates variables from all events within it. Due to the fact that some assumptions have to be made about how all events information should be encoded into single session vector this method is considered as an *explicit session modelling*. One solution, which is giving good results and is used in this work, is to aggregate all events data in a time decaying way $\mathbf{x}_s = \sum_{j=1}^{t} \frac{1}{1+t-j} \mathbf{x}_e^{(j)}$, where session vector size $d_S = d_E$ and t event sequence length. The final estimation is:

$$\hat{y}(\mathbf{x_s}, \mathbf{x_i}) = (\mathbf{x_s Q})(\mathbf{x_i P})^\intercal + \mathbf{x_s}\mathbf{b_p}^\intercal + \mathbf{x_i}\mathbf{b_q}^\intercal$$

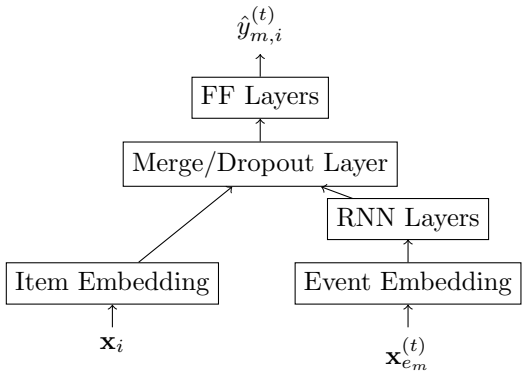

Figure 1: Neural Network Layers Architecture.

where $\mathbf{Q} \in \mathbb{R}^{d_S \times d}$ and $\mathbf{P} \in \mathbb{R}^{d_I \times d}$ are matrices with d-dimensional latent features for session and items variables respectively.

As a loss function the pairwise method is used in form of: Bayesian Personalized Ranking (BPR)[6], TOP-K as defined in work [2] and Weighted Approximated-Ranking Pairwise (WARP)[8] loss.

4.2 Session-Aware Recommendations with a Neural Network

The presented novel method assumes usage of two types of neural networks. Both, Recurrent Neural Network (RNN) and Feed Forward Neural Network (FFNN) are used to predict Top-N recommendations for the session. This combination is the key concept of handling event and item data represented only by their attributes. The RNN is used to capture data dependency between session events in time. It uses hidden state as the memory to handle variable length data. In this case, the sequence of events in session. The FFNN is used as a ranking score estimator. It uses the representation of session context returned by RNN and the new items data as an input. The simplified layer architecture of the network is presented in figure 1.

The input for the network is $\mathbf{x}_{e_m}^{(t)}$, the session event data for the time t and the item data to estimate $\mathbf{x}_i$. Before the input data is passed to next layers, the Embedding Layers map it to continuous, lower-dimensional space. The embedded events data is used in RNN Layers. The session events are passed sequentially. If the session ends, the RNN hidden state is reset for the new session. In RNN Layers different units can be used (Standard RNN, GRU or LSTM). The item data $\mathbf{x}_i$ do not have to be processed by RNN. Thus, after passing the embedding item data is gathered together with a RNN session representation in FFNN Layers. But before that, the dropout is applied as a regularization for the model. The FFNN Layers consists of multilayer perceptrons. It uses the non linear activation function to produce the output $\hat{y}_{m,i}^{(t)}$ - scoring for the current session m at time t of the item i.

In order to learn Top-N recommendation ranking, the pairwise approach for loss function is applied. For every time t the network is minimizing the loss in a form of BPR/TOP-K/WARP. Thus, for every Stochastic Gradient Descent up-

dataset	ALLEGRO	AVITO
items	24360	4374
sessions	20904	31826
events	535871	767550
searches	366874	110204
density %	0.028	0.464
s. len. mean	25.634	24.117
s. len. std	30.282	14.877

Table 1: Dataset statistics

date, a single event $\mathbf{x}_{e,t}$ is passed to calculate RNN Layers output, then multiple (i.e. in BPR/TOP-K method two) positive and negative item examples are passed to estimate $\hat{y}$ and calculate the loss.

5. EXPERIMENTS

5.1 Datasets

In evaluation, two real-life datasets are used: AVITO and ALLEGRO. In both, the session boundary is set by user inactivity for more than 30 minutes. Moreover, additional pre-filtering is made to constraint dataset size (computational purposes). Dataset statistics are shown in Table 1.

The ALLEGRO dataset is based on users' actions in Polish biggest e-marketplace[2]. In this dataset user actions (*search, view, watch/cart add/remove, buy, letter send*) from a single category (beauty products) is taken in a period of two weeks. Only sessions with more than 2 events are taken and events can only be connected with items with more than 10 clicks.

The second dataset, named AVITO, is created from data publicly available on Kaggle platform for context ad click prediction challenge[3]. Only events (*search, view, phone request*) and items from category 43 are used. Pre-filtering includes removing session without search and with less than 4 events. For items, a click cut-off threshold was set to 20.

For further processing, event and items in dataset is encoded. Bag-of-words minimal frequency is set to 10. This results in d_I=473/2710 and d_E=716/5523 and for AVITO/ ALLEGRO datasets respectively.

5.2 Experiments Settings and Model Training

For the reference, two baseline methods are used. First, the naive one, always returns the top N most popular (POP) items in the whole dataset. The second method is Content-Based (CB) filtering, where the cosine similarity is being used. The result is a list ordered by the similarity value calculated between items which have positive interactions with the user in the session and those which have not been seen yet.

The matrix factorization (MF) is done with a latent feature dimension set to 100. The L2 regularization is used with rate 0.005 while training the model. The BPR and TOP-K loss functions are applied. Negative examples are sampled uniformly from all items. Two types of the event representations are checked. First, where only the associated item (I) data is encoded and treated as event data. Second, where the item as well as the event context information (IE) are encoded and used. In both datasets, besides event type itself, the additional contextual information is delivered from search events. The MF model parameters are learned using adagrad optimization.

[2]http://www.allegro.pl/

[3]https://www.kaggle.com/c/avito-context-ad-clicks

	ALLEGRO		AVITO	
	REC@20	MRR@20	REC@20	MRR@20
NN-BPR-IE	**0.4131**±.0051	**0.1328**±.0023	0.1628±.0024	0.0476±.0009
MF-BPR-IE	0.3849±.0031	0.1150±.0003	0.1894±.0008	0.0404±.0001
NN-TOPK-IE	0.3367±.0057	0.0885±.0020	0.1985±.0031	0.0579±.0004
MF-TOPK-IE	0.2862±.0024	0.0773±.0017	**0.2699**±.0018	0.0658±.0005
NN-TOPK-I	0.2863±.0048	0.0709±.0025	0.2003±.0012	0.0579±.0004
MF-BPR-I	0.3080±.0013	0.0864±.0012	0.1883±.0015	0.0408±.0000
MF-TOPK-I	0.2353±.0025	0.0586±.0008	0.2691±.0001	**0.0660**±.0004
CB	0.1858	0.0354	0.1528	0.0243
POP	0.0499	0.0129	0.0193	0.0037

Table 2: Experiment results for Recall and Mean Reciprocal Rank for Top-20 recommendations. A row label describes used algorithm, loss function and contextual information (I - only items data, IE - items and events context). The mean values with 95% CI are given.

In the Neural Network session-aware recommendations the number of hyper-parameters is large: number of RNN/ FFNN layers, size of every hidden layer, type of an activation function, dropout, loss function. It is infeasible to perform full hyper-parameters grid search due to computation time over both datasets. Thus, after the first few experiments - best solutions was chosen for the future investigation and precise parameter tuning.

For the RNN Layers - single layer of the Gated Recurrent Unit is used. The FFN Layers consists of MLP with single hidden layer, where for non-linearity *tanh* activation is applied. The first embedding layers are skipped. For both datasets adding embedding to lower dimension is giving worse results. The same is observed in [2]. Nevertheless, this layer may be necessary for bigger datasets (in the meaning of d_E and d_I) and will be subject of future investigations. The GRU hidden dimension is 200 and for MLP 400 (for both the values: 200, 400, 800 was checked). The dropout at level 0.2-0.3 gives the best results.

The network is trained using adagrad optimization with learning rate 0.05 and 128 examples in a single minibatch. The minibatch is created of parallel sessions. If one of the sessions ends, the hidden state of RNN is reset. From the training dataset 3% random sessions are taken as a hold-out set for detecting overfitting. As a loss function BPR and TOP-K, for K=1 was evaluated.

5.3 Results and Discussion

As an evaluation measures Recall@N (REC@N) and Mean Reciprocal Rank (MRR@N) are chosen. The REC@N is used as an indicator for user engagement and the MRR@N as a measure of how good is the ranking. The test dataset was created from 5% of all sessions with latest starting time. Then, the evaluation measures are calculated for the task of predicting the next event for every subsequent events in the test session. This results in 1592/1046 test sessions which generated 291966/121475 event sequences as test cases for AVITO/ALLEGRO dataset.

Evaluation results for REC@20 and MRR@20 are presented in Table 2. For datasets used and two methods proposed in this work (MF and NN), both outperform baseline

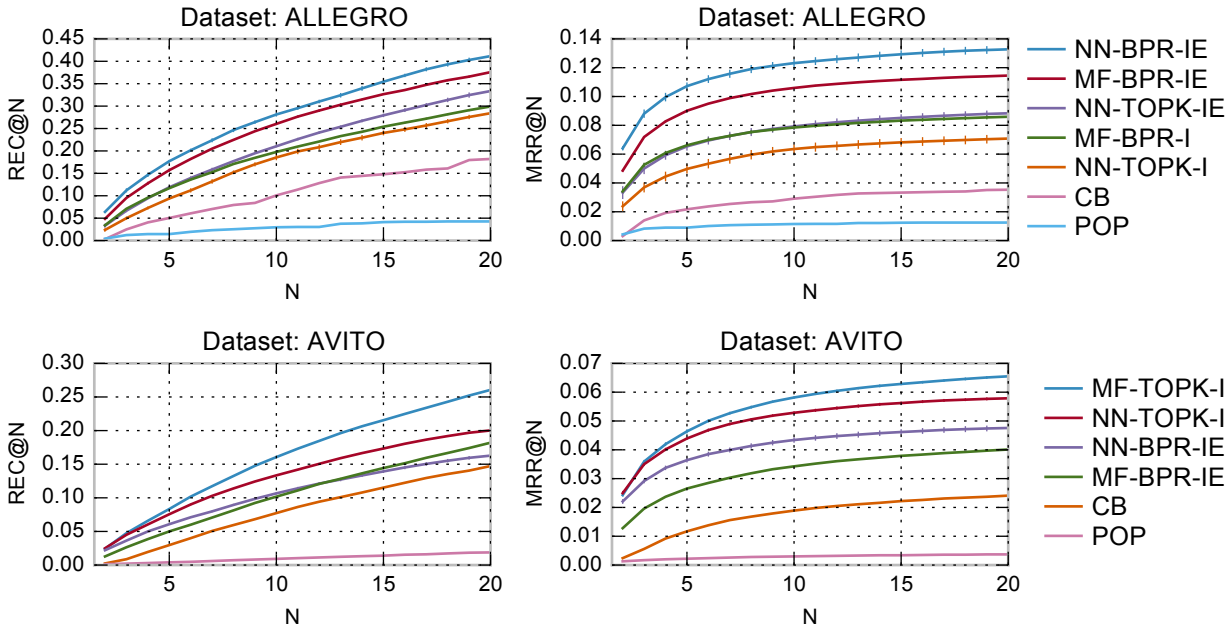

Figure 2: Results in Recall and MRR@Top-N Recommendations (figures share same legends on the right).

methods in the meaning of Recall and MRR. But the more interesting results are presented between the cases where only items data is used (I label suffix) and those where items and events contextual information is used (IE). In every case for the ALLEGRO dataset, where the additional information from event session was incorporated, both methods MF and NN gave better results. The situation is different for the AVITO dataset, where additional event context gives only a slightly better results for MF methods and for NN-TOPK are even lower. On ALLEGRO dataset, the NN significantly outperforms MF methods, while on AVITO, using the same NN configuration results in worst outcome. The difference can be explained by the available attributes of contextual information. The ALLEGRO dataset is rich in terms of items and search contextual information, while AVITO, besides having less items and more events, has less data. After encoding in the same way, both items and events from AVITO dataset have less real-value features than those for ALLEGRO dataset and the same entities. Furthermore, the hyper-parameters setting should be also revised for AVITO, as there is a big space for improvement as the MF-TOPK results present.

Figure 2 presents results of REC@N and MRR@N for N values up to 20. For the sake of readability, only the selected best methods and baselines are presented for each dataset.

6. CONCLUSIONS AND FUTURE WORK

In this paper two new methods for session-aware recommendations are proposed. Methods are presented in the difficult setting, where items are ephemeral (represented only by a set of attributes) and users are not identified between two separate sessions. First method uses matrix factorization technique and explicit session context modeling. The second, applies Recurrent Neural Network to automatic session context modeling. Both outperform significantly baseline approaches in task of top-N recommendations verified by measuring REC@N/MRR@N. However, the usage of RNN is considered more attractive and flexible when no session modeling assumptions is made.

In the future work, more research of using neural network in the field of recommender systems is planed. One research area will involve using RNN to model cross-session/long-term user goals. More experiments over different and bigger datasets are planed. Therefore, the problem of searching the best hyper-parameters and then effective training NN will be investigated.

7. REFERENCES

[1] D. Gayo-Avello. A survey on session detection methods in query logs and a proposal for future evaluation. *Information Sciences*, 179(12), 2009.

[2] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based Recommendations with Recurrent Neural Networks. *International Conference on Learning Representations*, 2016.

[3] B. Hidasi and D. Tikk. General factorization framework for context-aware recommendations. *Data Mining and Knowledge Discovery*, 30, 2016.

[4] Y. Koren. Collaborative filtering with temporal dynamics. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, 2009.

[5] U. B. Maryam Tavakol. Factored MDPs for detecting topics of user sessions. *Proceedings of the 8th ACM Conference on Recommender systems*, 2014.

[6] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-thieme. BPR : Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009.

[7] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, 2011.

[8] J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. *IJCAI International Joint Conference on Artificial Intelligence*, 2011.