Modelling Session Activity with Neural Embedding

Oren Barkan
Microsoft, Israel

Yael Brumer
Microsoft, Israel

Noam Koenigstein
Microsoft, Israel

ABSTRACT

Neural embedding techniques are being applied in a growing number of machine learning applications. In this work, we demonstrate a neural embedding technique to model users' session activity. Specifically, we consider a dataset collected from Microsoft's App Store consisting of user sessions that include sequential click actions and item purchases. Our goal is to learn a latent manifold that captures users' session activity and can be utilized for contextual recommendations in an online app store.

Keywords

Skip-Gram, Collaborative Filtering, Recommender Systems

1. INTRODUCTION

Neural embedding models have significantly advanced state-of-the-art in the field of Natural Language Processing [1], [2]. In Recommender Systems research, neural networks have been applied in Collaborative Filtering (CF) [3], and basket completion [4]. Specifically, [5] presented a neural model for embedding items in a latent manifold that encodes CF information. These early works have been published very recently and indicate a growing interest in neural embedding techniques for recommendations.

In this work, we take a different direction and utilize neural embedding techniques to model users' session activity. Specifically, we consider a dataset collected from Microsoft's App Store consisting of user sessions that include sequential click actions and item purchases. Our goal is to learn a latent manifold that captures users' session activity and can be utilized for contextual recommendations of apps in an online app store.

Most prominent CF techniques such as Matrix Factorization [6] do not take into account the sequential order of user actions prior to purchasing an item. Recently, there is a lot of interest around user behavior modeling to predict purchases. One of the latest competition "Tmall Recommendation Prize" requires to predict future user purchases on Tmall website [7]. While they build user profiles to predict purchases, we try to model session behavior regardless the user profile. Another approach [8] uses LSTM-BiRNN to learn sequence clicks made at the same session to predict all purchases associated with this session, while we try to predict the next purchased item given a click action that made only in a pre-defined window before the purchase.

The underlying assumption in this work is that users consider several items prior to their ultimate decision to purchase. Hence, we model users' session activity as a sequence of click events on item detail pages and purchase events. For example, (C1, C2, C3, C4, C5, P5) denotes a user session consisting of 5 click event on 5 different items followed by a single purchase event. Note that an item purchase event is always preceded by a click event on the same item. By learning to predict these sequences, one can improve the overall user experience by recommending the items that the user is most likely to purchase.

2. DATASET

Our dataset collected from Microsoft's App Store consisting of user sessions that include sequential click actions and item purchases and is based on a sample of activity sessions from March to June 2016. Each action, whether it is click or purchase, is uniquely identified by the session id, timestamp and item id. From this sample, all sessions with less than two different clicked items, or without purchase event are removed. The effective dataset consists of 8,785,295 distinct sessions that contains 43,956,340 clicks and 18,838,796 purchases. On average, each session is associated with 5.003 clicks and 2.144 purchases, while the total number of distinct items is 22,139. In every session, the actions are ordered by their timestamps.

3. NEURAL ACTION EMBEDDING

Our model is inspired by Skip-Gram with Negative Sampling (SGNS) also known as Word2vec [2]. As explained above, we wish to model the user actions in a dataset $\mathcal{D} = \{s_i\}_{i=1}^{K}$ of K ordered user activity sequences where the i'th sequence is $s_i = (a_{i,1}, a_{i,2}, \dots, a_{i,L_i})$, and L_i is its length. The set of all possible actions is denoted by $\mathcal{A}$ and includes in our case click and purchase events of different items from the items catalog. We further define a function $A: \mathcal{A} \rightarrow \{C, P\}$ that maps between an action to its type (click or purchase).

Our objective is to maximize the following term:

$$\frac{1}{K}\sum_{i=1}^{K}\sum_{(k,j)\in I_i}\log p\left(a_{i,k}\middle|a_{i,j}\right). \qquad (1)$$

where, $I_i \subseteq \{(k,j): 0 \leq j < k \leq L_i\}$ is a set that contains tuples of sequential actions. The probability $p\left(a_{i,k}\middle|a_{i,j}\right)$ is defined by

$$p\left(a_{i,k}\middle|a_{i,j}\right) = \sigma\left(u_{a_{i,j}}^{T} v_{a_{i,k}}\right)\prod_{n=1}^{N}\sigma\left(-u_{a_{i,j}}^{T} v_{a_n}\right)$$

where $\sigma(x) = 1/1 + \exp(-x)$, $u_a \in U(\subset \mathbb{R}^m)$ and $v_a \in V(\subset \mathbb{R}^m)$ are latent vectors corresponding to the target and context representation of action a. The parameter m is chosen empirically through cross-validation. N is a parameter that determines the number of negative examples to be drawn per a positive example. A negative action a is sampled from a distribution that is proportional to the frequency of the item that is associated with a . In this work, we use the unigram distribution raised to the 3/4rd power.

In order to mitigate the effect of popularity and produce better modeling for unpopular items, we subsample the sessions. Specifically, we discard each action a with the probability $p(discard|a) = 1 - \sqrt{\rho/f(y(a))}$ where $y(a)$ is the item that is associated with the action a, $f(x)$ is the frequency of the item x and ρ is a parameter that controls how aggressive is the subsampling. Finally, the latent vectors are estimated by applying a stochastic gradient ascent with respect to the objective in Eq. (1).

4. EVALUATION

In this section, we describe our evaluation of the proposed model. Our prediction task is to predict the next purchased item given a click event. To this end, we split the dataset according to the session order. The first 90% sessions are used as a training set and the remaining 10% are used as a test set. For each test session, we form a set of test (C, P) tuples, where each tuple corresponds to a purchase action that is preceded by a click action. A tuple (C, P) is considered only if C and P distant by at most three other actions. For example, for a given test session (C1, P1, C2, C3, C4, P4), the tuple (C3, P4) can be made because the distance between them is a single action C4. On the other hand, the tuple (C1, P4) cannot be made since the distance between them is four actions. Furthermore, we exclude trivial tuples that consists of click and purchase of the same item. The resulted test set contains ~ 2M tuples.

4.1 Model Variants

We compare three variants of the proposed model. The variants differ by the tuples that the model is trained with. The set of tuples for each model is determined by the choice of I_i in Eq. (1).

The first model dubbed 'CP' comprised of tuples that are created in a similar manner to the test tuples. Specifically, for a given training session s_i, we set
$I_i = \{(k,j): k,j \in \{0..L_i\} \wedge 2 \le k-j \le d \wedge A(a_{i,k}) = P \wedge A(a_{i,j}) = C\}$.
As a result, in this model, U and V are the representations of the clicks and purchases, respectively.

The second model is dubbed 'CC' and comprised of the sequential click events without the purchase events. The reasoning behind this model is the fact that each purchase event is preceded by a click on the same item that was purchased. Hence, by predicting the next item the user will click upon, we are also predicting the next item that would be purchased. Therefore, in this model we set
$I_i = \{(k,j): k,j \in \{0..L_i\} \wedge 1 \le k-j \le d \wedge A(a_{i,k}) = C \wedge A(a_{i,j}) = C\}$.

The third model, dubbed 'PP' is comprised of sequential purchase events (without clicks). Many Collaborative Filtering algorithms are designed to predict the next item a user will purchase, given the items he already purchased. Hence, the 'PP' model was chosen as a baseline that follows a similar approach taken by many contemporary algorithms. In the 'PP' model we use
$I_i = \{(k,j): k,j \in \{0..L_i\} \wedge 1 \le k-j \le d \wedge A(a_{i,k}) = P \wedge A(a_{i,j}) = P\}$.

4.2 Parameters

We used the following parameter configuration: we set the negative to positive ratio N to 15. ρ was set to 1e-3. d was set to 4. All three models were trained for 50 iterations. It is important to clarify that we experimented with different values of $d = 2..10$ and no significant change in the results was observed.

4.3 Evaluation Metrics and Results

Our first evaluation is based on measuring the Percentile Ranks (PR) of the hidden items (purchased items). We report results in terms of Mean Percentile Rank as well as Median Percentile Rank.

Table 1 summarizes the mean and median PR for the different models. The CP model clearly outperforms both the CC and PP models. We therefore conclude that both previous click events as well as previous purchases are relevant in this prediction task. Ignoring each of these signals, undermines the model's ability to detect the hidden item.

A second observation is the fact that the Median PR values are much better than the Mean PR values and the performance difference between the models becomes smaller when the Median PR is considered. This suggests that the Mean PR values are highly affected by a small number of bad examples but in most cases the predictions are much better than the Mean PR values. This behavior characterizes all three models, but more dominant in the CC and CP models.

Table 1. Percentile Rank (PR) of the hidden (purchased) item

	CP	CC	PP
Mean PR	6.49%	11.84%	10.91%
Median PR	0.72%	0.89%	0.89%

Table 2. Precision@K values for difference models

Graphics	CP	CC	PP
K=10	0.21	0.16	0.16
K=25	0.30	0.25	0.26
K=50	0.38	0.33	0.33
K=100	0.46	0.42	0.42

Our second evaluation is based on the Precision@K metric: For each model, we measure the percentage of test examples in which the hidden item was ranked in the top K.

Table 2 presents Precision@K values for different values of K. The results coincide with those of Table 1 where the CP model shows significantly better results across the board. Again, these results emphasize that unlike most present day models, that consider only one type of events, there is significant added value in the CP approach that models click and purchase event simultaneously.

5. CONCLUSION AND FUTURE WORK

In this work, we present and evaluate several variants of neural embedding models for predicting purchases from user activity sessions. The evaluation shows that learning from click-purchase relations in different scales provide better results than learning from either click-click or purchase-purchase relations.

In future, we plan to investigate the contribution of additional hidden layers to the model presented in this paper and compare between our model to sequential neural models such as LSTM [8] for the same prediction task.

6. REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A Neural Probabilistic Language Model," *JMLR*, vol. 3, pp. 1137–1155, Mar. 2003.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *NIPS 26*, 2013, pp. 3111–3119.

[3] F. Strub, J. Mary, and R. Gaudel, "Hybrid Collaborative Filtering with Neural Networks," *CoRR*, vol. abs/1603.00806, 2016.

[4] S. Wan, Y. Lan, P. Wang, J. Guo, J. Xu, and X. Cheng, "Next Basket Recommendation with Neural Networks," in *Poster Proceedings of RecSys 2015.*

[5] O. Barkan and N. Koenigstein, "Item2Vec : Neural Item Embedding for Collaborative Filtering," arXiv preprint arXiv:1603.04259, 2016.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE Computer,* vol. 42, no. 8, pp. 30–37, Aug. 2009.

[7] Zhang, Yuyu, et al. "Large Scale Purchase Prediction with Historical User Actions on B2C Online Retail Platform." arXiv preprint arXiv:1408.6515, 2014

[8] WU, Zhenzhou, et al. Neural Modeling of Buying Behaviour for E-Commerce from Clicking Patterns. In: Proceedings of the 2015 International ACM Recommender Systems Challenge. ACM, 2015. p. 12.