Modeling Mobile User Actions for Purchase Recommendation using Deep Memory Networks

Djordje Gligorijevic*
Temple University
Philadelphia, PA, USA
gligorijevic@temple.edu

Jelena Gligorijevic*
Temple University
Philadelphia, PA, USA
jelena.stojanovic@temple.edu

Aravindan Raghuveer*
Facebook Inc.
Menlo Park, CA, USA
aravindr@fb.com

Mihajlo Grbovic*
Airbnb
San Francisco, CA, USA
mihajlo.grbovic@airbnb.com

Zoran Obradovic
Temple University
Philadelphia, PA, USA
zoran.obradovic@temple.edu

ABSTRACT

Rapid expansion of mobile devices has brought an unprecedented opportunity for mobile operators and content publishers to reach many users at any point in time. Understanding usage patterns of mobile applications (apps) is an integral task that precedes advertising efforts of providing relevant recommendations to users. However, this task can be very arduous due to the unstructured nature of app data, with sparseness in available information. This study proposes a novel approach to learn representations of mobile user actions using Deep Memory Networks. We validate the proposed approach on millions of app usage sessions built from large scale feeds of mobile app events and mobile purchase receipts. The empirical study demonstrates that the proposed approach performed better compared to several competitive baselines in terms of recommendation precision quality. To the best of our knowledge this is the first study analyzing app usage patterns for purchase recommendation.

ACM Reference Format:
Djordje Gligorijevic, Jelena Gligorijevic, Aravindan Raghuveer, Mihajlo Grbovic, and Zoran Obradovic. 2018. Modeling Mobile User Actions for Purchase Recommendation using Deep Memory Networks. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3209978.3210138

1 INTRODUCTION

We have been witnessing rapid growth in the number of smart mobile devices recently, followed by the increase of mobile data traffic of 74% in 2015 and 63% in 2016[1], and it is further estimated that in the near future the majority of the overall Internet consumption will be mobile. Such climate of ubiquitous mobile usage provides a growing market opportunity for mobile advertising with greater reach than more traditional online advertising [1].

In order to continuously improve advertising efforts, a better understanding of mobile usage patterns is necessary, especially those usage actions which lead to a mobile purchase. Mining users' *mobile app actions* and *purchasing habits* is a fundamental task for enabling better context-aware design and delivery of advertisements and recommendations in the form of relevant apps or services that could be of interest to the user.

In this light, monitoring users' mobile activities and generating ***app event logs*** from users who opted for anonymous advertising studies serves as a rich source of data. Another relevant piece of information comes from users' purchase habits, which can be found in ***app store purchase receipts***. In our study, the data will ultimately consist of registered users that made purchases on a mobile app store and use one or more apps registered on proprietary advertisement SDK. The usefulness of the app event signals for predicting mobile purchase can be taken from the fact that approximately 50% of purchases were preceded by more than ten signals from apps registered to a proprietary SDK in a one-hour window. Coupling the two data sources would result in sessions as shown in Figure 1. App event logs are thus perceived as *context* of purchased items. In our example we observe a user over a specific time window, spending a period of time communicating with friends and listening to music, followed by playing a mobile game (Angry Birds in our example), after which he purchased coins for the game. The example session depicts a user whose main intent at that time was to be entertained, which often results in purchasing of in-app goods for a complete experience.

Predicting users' purchase intent is a difficult problem primarily due to large universe of products and services (items) that a user can buy. Furthermore, app events are traditionally defined by the app developers who provide a free-form textual description of the event, therefore lacking uniformity over the universe of apps, which poses a challenge of using this information.

The goal of this study is to propose a model capable of coping with the aforementioned challenges and capturing useful patterns from app event sequences that can be used to anticipate users' purchase intents, improving ad targeting and app recommendation capabilities. Modeling user actions as sessions and learning low-dimensional distributed representations of events in a sequence

*This work was conducted while the authors were at Yahoo! Research
[1]http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, acc. June 2017.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210138

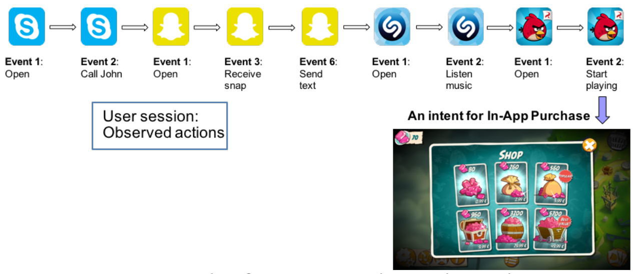

Figure 1: Toy example of user session log with purchase intent.

has recently brought benefits over more traditional approaches such as collaborative filtering or clustering. However, these approaches are primarily unsupervised, or rather they do not model supervised information explicitly. We aim to employ a family of models that can exploit the power of neural embedding models, while retaining the ability to explicitly model supervised information given a sequence. Namely, we exploit the recently proposed Deep Memory Networks models [5], capable of explicitly modeling memory relevant for predicting current events. For example, users that made an in-app purchase for a mobile game or bought songs within a mobile store are likely to do so again. Our results show that the proposed approach outperforms other approaches on mobile purchase prediction while learning meaningful representations of noisy events.

2 RELATED WORK

Majority of the research on mining mobile app data focused on classifying mobile apps in a set of categories used in different app marketplaces using data collected from session logs of users and other available app metadata. Approaches consisted of variety of topic modeling techniques [8], collaborative filtering [7] and neural embeddings [3]. Aside from [3], mentioned approaches mostly conducted experiments on a limited number of users (< 500) and on a relatively small number of apps, which leaves scalability as an open question. Additionally, taking into account that different users may have different preferences still remains as an open research question given that in most cases users' context was ignored. These issues are tackled in this study. Empirical evidence shows benefits of using our approach over baselines that reflect studies mentioned above.

3 DATA

As discussed in the introduction, our data comes from two data sources: mobile purchase receipts data and applications events feed.

Purchase receipts data: Data set of purchase receipts from users was constructed while anonymizing user IDs for privacy protection reasons and were analyzed by automated systems. Product information were extracted from receipts using an in-house extraction tool. We collected more than 2.5 million app store purchase receipts over three months, received by a subset of users. Users vastly prefer to purchase single items ($\sim 70\%$), which tells about a direct intent from users to purchase a particular item of interest, rather than buying in bundles. As mobile advertising efforts allow for a limited number of ads due to screen size (usually up to 5 ads if application does not have a scrollable screen), it is very important to advertise items that are relevant to the user in a given time point, rather than placing more generic ads. Predicting the purchase of mobile items will be tackled using signals from mobile actions collected from a proprietary mobile advertising SDK.

Flurry SDK application events data: Mobile events data are collected from the mobile analytics, monetization, and advertising platform, Flurry, that allows users to register their smart device applications and receive rich insights while advertising within them. The data consists primarily of a feed of user app event activities, logged in tens of billions a day. For the purpose of this study, activities are sampled only for mobile users who voluntarily opted-in for such studies, and their anonymized IDs were joined with purchased receipts data. In this dataset, we observed that approximately 75% of purchases were preceded by more than one app event, 50% by more than ten events and 10% by more than 20 events. We use these events feeds as the context of a mobile purchase, with the assumption that usage activities reveal users' purchase intent.

Prediction task: Our goal is to estimate what item the user is most likely to purchase given their mobile app events session, such that the ad/recommender system can anticipate and react by providing timely recommendations for items of interest to the user.

4 METHODOLOGY

We first provide formulation of our prediction task, after which, we propose the methodology for tackling the formulated problem.

4.1 Problem set-up

Let us assume that we are given a set of mobile app events from app universe $\mathcal{A}$, and a set of app store purchase items from item universe $\mathcal{T}$. Further, we consider a set of app events $\mathcal{X}_n \subset \mathcal{A}$, associated with a timestamp, that occurred in an arbitrarily chosen time window (in our experiments one hour) prior to an app store receipt containing set of items $\mathcal{Y}_n \subset \mathcal{T}$, representing the purchases of a user for each independent session x_n, where $x_n = \{\mathcal{X}_n, \mathcal{Y}_n\} = \{\{a_1, a_2, a_3 \ldots\}, \{t_1, t_2, \ldots\}\}$. We consider the task of app store items recommendations, namely, given all app events $\mathcal{X}_n$ a user took in a time window, we wish to predict items that the user is most likely to purchase. By retrieving the most likely or K most likely items, we are providing recommendations to the user.

4.2 Proposed Approach

Here, we propose to use and extend a Deep Memory Networks (DMN) model [5] for the task of mobile purchase recommendation. DMN builds upon previously proposed distributed embedding approaches [6] to which we compare experimentally. Compared to these distributed embedding approaches which only model a given sequence of actions, DMNs are capable of modeling provided external set of actions (external memory) as a context to current actions and use such information together with the current action sequence for target prediction. DMNs are particularly interesting models for applications where previous actions matter and can be stored as an external memory (i.e., DMN models have been successfully applied to several tasks such as question answering tasks [5]).

We adapt and develop these models for mining context-rich mobile sessions. Their purpose is to learn representations of previous app events and purchased app store items as external memory

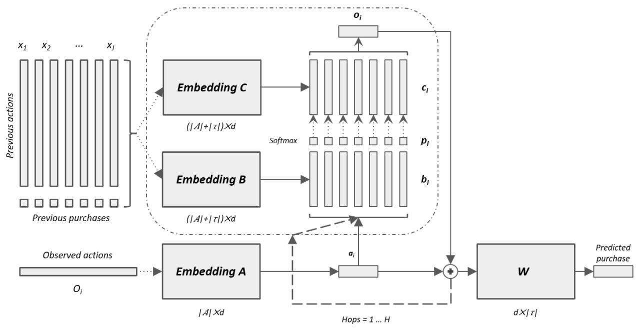

Figure 2: Graphical representation of Deep Memory Network for predicting mobile purchases based on observed actions

(useful for modeling users' preferences), in addition to learning representation of current app events with the final goal of predicting the following purchased item.

4.2.1 Deep Memory Networks for Purchase Prediction on Mobile Devices. External memory is formulated as a set of sessions S_i preceding i^{th} session x_i, ordered by time of purchases and for personalized recommendations, grouped for each user then ordered by time of purchase. This way, the DMN model keeps memory of historical users' action sessions and purchase that follow them. DMN, thus, has a task of capturing patterns of long- and short-term actions mobile users have prior to the mobile store purchase and applying learned rules on currently observed app event actions to predict items most likely to be purchased. In order to make predictions, DMNs take two different inputs, the first input is a *set* of historically observed sessions stored as a "memory" (the size of the memory J is chosen by the user), and the second input is the currently observed mobile app events session. The model finally outputs the most likely item to be purchased from universe $\mathcal{T}$.

The graphical representation of the forward propagation steps of the DMN model are depicted in Figure 2. The first step is to embed observed app events session (embedding matrix **A**) and previous sessions into continuous vector representations (embedding matrices embedding matrix **B** and embedding matrix **C**). Observed app events are embedded into a vector a_i. a_i is then used to learn relevance scores for each event in the memory using attention mechanism (embedding **B** with Softmax layer). Obtained relevance scores (p_i) are used to summarize entire memory in a single vector using weighted sum. The two vectors are combined using dot product resulting in vector o_i. The process can be repeated in several layers ($H \geq 1$). Finally, the combined representation is used as a summary of the memory and observed app events and fed to a final fully connected layer to retrieve potential items to be purchased.

Learning the DMN. Optimization objective of DMN model can be defined as a cross entropy loss. The goal of the loss function is to optimize embedding matrices A, B, C, and weight matrix W: $\hat{A}, \hat{B}, \hat{C}, \hat{W} = argminH$. For optimizing the DMN model we use an Adam optimizer which controls the learning rate.

4.2.2 Mobile user embeddings. Learning user specific behavior by the DMN is achieved using two steps. The first step is to model embeddings for users, this can be achieved by learning representation u_n of user n in sessions $x_i^{(n)} = \{u_n, \{a_1, a_2, a_3, \ldots\}, \{t_1, t_2, \ldots\}\}$. Second, memory context of observed user actions are only his previous activities with purchases. The model is thus forced to account for the user specificities, providing personalized item predictions.

4.2.3 Optimization at scale. The cardinality of mobile app events $\mathcal{A}$ and app store items $\mathcal{T}$ can be very large, which poses a problem for modeling, especially at scale. With such cardinality, running the final softmax layer in Memory Networks model over all items can be infeasible. In this study, however, we resort to the importance sampling technique called sampled softmax [4] to allow the model to scale for the large vocabulary of app store items found in our data. We opted for sampled softmax as we need speedup at training time only, and wish to have ability to retain logits for all items at inference time.

5 EXPERIMENTS

The app embeddings are learned using large-scale app event logs coupled with mobile app store receipts. Mobile app sessions were built as observed apps events an hour before app store purchase was made. The training set is generated from sessions for 80 days, totaling 1,517,830 sessions, while the testing set consist of the 15 days data and totals 37,476 sessions.

Baselines for the experiments are selected to either represent previously published studies or as models that are expected to fit well with the given setup and they include: Naïve distribution sampling of items to purchase, LDA [8], word2vec [2] and RNN [6].

As the evaluation metric we use *recall@K*, which measures success of guessing relevant (purchased) items. The *recall@K* is computed for each mobile session as a fraction of truly predicted purchased items within the K retrieved ones, and average recall is reported. For baseline model comparison, we use *recall*@5 measure, as most of the mobile screens allow up to 5 advertisements.

5.1 Predicting app store items purchase

The goal of this task is to predict the actual item that a user will purchase after a session of app events. The predictive problem is very difficult due to the large universe of targets ($\sim$ 140,000 unique items selected that were purchased more than 10 times in our dataset). If we would model target uniformly, the recall@5 would be $3.6e^{-5}$, demonstrating the difficulty of the task at hand.

When examining prediction of most likely purchased items (recall@5 in Table 1), we observed that the DMN model outperforms all baselines. The second-best model among the baselines is word2vec model with nearest neighbors search in the embedded space, followed by RNN approach, while LDA and Naïve model, failed to

Table 1: Recall@5 for DMN model over baselines on purchase items prediction task.

Model	Recall@5
Naïve	0.000151
LDA	0.000475
RNN	0.001089
word2vec	0.001576
DMN	*0.003315*
DMN-user	**0.028720**

Table 2: Hand curated 10 app events among 50 app events in the embedded space for query app events Snapchat: open and Snapchat:close as learned by the DMN model.

Snapchat: open	Snapchat: close
App: started_resumed_a_game	App: game_over
App: start_of_run	App: congrats_not_shown
App: ad_open_question	App: boost_purchased
App: start_session	App: mission_progress
App: 01_start_session	App: send_longchat_message
App: new_game_started	App: damage_dealt_summary
App: start_session	App: sbsdk_close_sidebar
App: start_session	App: response_error
App: start_session	App: choose_friend_viewed
App: start_session	App: video_proc_canceled
App: start_session	App: ce_player_save_id

deliver comparable accuracy to the other baselines. This demonstrates that the DMN model copes better with the noise in the data and class disbalance than the baselines.

Retrieving K most likely purchased app store items for the DMN model. We analyzed behavior of the top competing DMN model when retrieving different number (top K) of the most likely purchased items, where value of K is varied from 1 to 10. From Figure 3 we see (in blue) significant improvement after initial increases in K. Performance of the DMN model nearly doubles after we retrieve 2 most likely purchased items, and it increases up to 2.2 times for the top 5 items, after which it almost flattened out.

5.1.1 Analysis of the embedded space. Here, we aim to analyze the capability the DMN model's embeddings to disambiguate events across different applications. A convention in naming app events exits, however it is not always used, therefore the model must be capable of learning meaningful representations of such noisy event names across different apps. To analyze the app events vector space, we show the nearest neighbors for two notably frequent app events: Snapchat 'open' and 'close', using distributed embedding learned in matrix **A**. To anonymize apps due to privacy restrictions, mobile app event tokens are converted from their original form into generic 'App: event_name', where 'App' hides the actual name. Note that in Table 2, there are no two identical Apps, while many events are coded by the standard convention.

The 'open' event primarily annotates events of opening the app, which is usually immediately followed by primary app activities. Example of these are games, where the user starts playing them immediately upon opening the game app. Another, notably frequent event in neighbors of 'Snapchat_open' are 'start_session' events, which are common initializing keywords for many apps. These examples are present in the first column of Table 2. In the second column we analyze 'close' event. Events in the same context as the 'close' event have different labels, whether they annotate ending of gameplay for game apps, app activity failures or finishing interaction activities with other users of the same app, they can semantically be considered similar. These examples show that distributed representation of DMN model were able to semantically disambiguate app events and thus efficiently use them for purchase intent predictions.

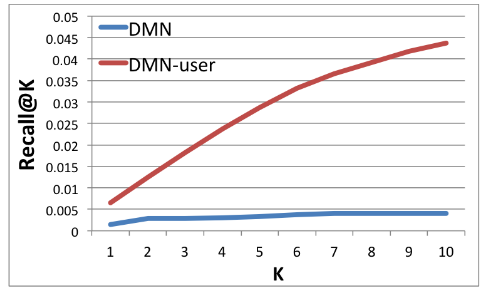

Figure 3: Recall@K for DMN vs. DMN-user model

5.2 Adding user embeddings to the context

We finally discuss results obtained on a user-specific session logs dataset compared to the simple time–ordered session logs dataset used in previous experiments. $504,383$ user specific sessions were used with a minimum of 3 purchases. Number of app events and app store items was down-sampled to $\sim 50,000$ most frequent of each, to compensate for the smaller memory size this setup can use.

Figure 3 shows improvements in *Recall@K* for the the user-DMN model (red) over plain DMN model (blue). We can see that we obtained 4.5 and $\sim$ 10 times better $recall@K$ for $K = 1$ $K = 10$, respectively, when accounting for user embedding and focusing memory on user's activities for item predictions. Compelling improvements, suggest that personalized mobile advertising is desirable to use, whenever possible, or when a user is observed in historical sessions. We also observe the improvement of 7 times in recall for $K = 10$ vs $K = 1$ for the DMN-user model, leaving advertisers higher potential for larger variability in purchase recommendation.

6 CONCLUSIONS AND FUTURE WORK

In this study we proposed a method based on neural language models and Deep Memory Networks for modeling purchases on mobile devices using app session events. To tackle scalability, we proposed to use sampled softmax in the final layer of the deep model, and additionally showed that extending the model to personalized recommendations is possible and superior to the non-personalized approach. The results of this study suggest that DMNs appear to be suitable models for tackling this problem as they outperform all baselines.

REFERENCES

[1] Narayan Bhamidipati, Ravi Kant, and Shaunak Mishra. 2017. A large scale prediction engine for app install clicks and conversions. In *CIKM 2017*. 167–175.
[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS 2013*.
[3] V. Radosavljevic, M. Grbovic, N. Djuric, N. Bhamidipati, D. Zhang, J. Wang, J. Dang, H. Huang, A. Nagarajan, and P. Chen. 2016. Smartphone app categorization for interest targeting in advertising marketplace. In *WWW2016*.
[4] Jean Sébastien, Cho KyungHyun, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *ACL 2015*.
[5] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *NIPS 2015*. 2440–2448.
[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS 2014*. 3104–3112.
[7] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative filtering meets mobile recommendation: A user-centered approach. In *AAAI 2010*, Vol. 10. 236–241.
[8] Hengshu Zhu, Enhong Chen, Hui Xiong, Kuifei Yu, Huanhuan Cao, and Jilei Tian. 2015. Mining mobile user preferences for personalized context-aware recommendation. *ACM Trans. Intell. Syst. Technol. (TIST)* 5, 4 (2015), 58.