Incorporating Dwell Time in Session-Based Recommendations with Recurrent Neural Networks

Research In Progress[†]

Veronika Bogina
Haifa University
Israel
sveron@gmail.com

Tsvi Kuflik
Haifa University
Israel
tsvikak@is.haifa.ac.il

ABSTRACT

Recurrent Neural Networks (RNN) is a frequently used technique for sequence data predictions. Recently, it gains popularity in the Recommender Systems domain, especially for session-based recommendations where naturally, each session is defined as a sequence of clicks, and timestamped data per click is available. In our research, in its early stages, we explore the value of incorporating dwell time into existing RNN framework for session-based recommendations by boosting items above the predefined dwell time threshold. We show improvement in recall@20 and MRR@20 by evaluating the proposed approach on e-commerce RecSys'15 challenge dataset.

CCS CONCEPTS

Information retrieval -> Retrieval tasks and goals -> Recommender systems

KEYWORDS

Recommender systems, temporal aspects, dwell time, deep learning, recurrent neural networks

ACM Reference format:

V.Bogina, T.Kuflik 2017. Incorporating Dwell Time in Session-Based Recommendations with Recurrent Neural Networks. SIG Proceedings Paper in Word Format. In Proceedings of RecTemp Workshop co-located with ACM RecSys'2017, Como, Italy

1 MOTIVATION

Nowadays, users are flooded with a wide variety of items to purchase/listen/read in the Sea of Possibilities over the Internet. In this scenario, relevant and useful recommendations can be a life saver for a customer by reducing the number of alternatives. Hence, predicting next items a customer will be interested in (to click on, listen to, read through etc.) is an ongoing attractive and interesting research task, as it is important to the service provider and the customer alike [5]. To succeed in predicting the "next item to be clicked", some of the researchers considered past sessions of the user [5][8], item repetition in past interaction, favorite items, items co-occurrences, topics similarity [7] and more.

However, as the previous studies [2][10] show, dwell time plays a significant role in predictions based on implicit users' feedback which is defined as a sequence of clicks in news, music and e-commerce domains. Hitherto, with the development of RNN, sequences of events were taken into consideration, still without incorporating the dwell time - the time that user spent examining this specific item [4][5][6][9].

Therefore, we decided to explore the effect of incorporating dwell time into the input data of the RNN based on the existing framework, and evaluate it in a dataset taken from the e-commerce domain.

Next, we will describe first our method, then we present an initial evaluation and, finally, discuss our findings and future research directions.

2 METHOD

The general idea of the proposed method is that the longer user examines an item (stays on its web page), the more interested s/he is in that item. Obviously, we are not talking about outliers, where there is a possibility that the user just left the application or simply kept the web page open while moving away from the computer.
Therefore, this approach can be used in the next click recommendations techniques.
Let an e-commerce session be a sequence of clicks $\{x_1,\ldots,x_n\}$ and for each click there is a dwell time dt_i. We propose to use Boosting to boost items that have dwell time greater than a predefined threshold t. This way we multiply the number of such items in the session and define the number of occurrences of the item in the session as $(dt_i/t + 1)$.
Hidasi et al.[5] proposed to use the Gated Recurrent Unit (GRU) based RNN for session-based recommendations that was preferable to LSTM in its performance [3]. One of the challenges in session based recommendations modelling with RNN is that sessions differ in their length. Therefore, the authors proposed to represent each mini-batch as set of elements from parallel sessions – refer to **Fig1** and then predict *next* elements in the parallel sessions by predicting next mini-batch. On the left, we see the original click sequences while on the right – input to the model and an output from the model. All sessions are ordered by session id and time. After that the first elements of the few sessions are used in the first mini batch, the second elements in the second min-batch and so forth. Only elements with next element available are used in the input. Once session ends, another one is proceeded (as in the *Tetris* game). The

output forms the set of the next items for each item in the input batch. Therefore, the last element in the session is not part of the input (no item follows it), as well as the first element of the session is not part of the output (no item precedes it). We will refer to their model in the paper as *GRU4Rec*.

In our study, we propose to enrich Hidasi et al. [5] model (*GRU4Rec*) with the representation of dwell time as additional elements by incorporating item boosting.

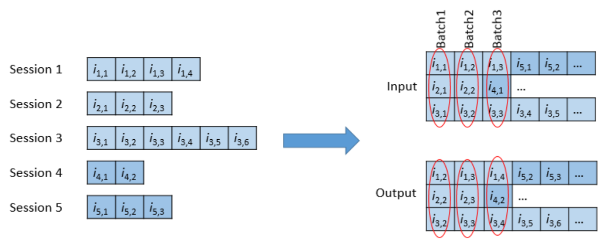

Figure 1. Session-parallel mini batch creation (following Hidasi et.al paper).

In our proposed method, session *i* is represented by clicks, based on their dwell time. Let's assume that the predefined threshold for the dwell time is *t* seconds and in *session2* the dwell time of the first element is greater than *2t* seconds but less than 3*t*, then this session parallel mini batch is different from the previous one (as described in Fig1) by inclusion of 2 instances of the first element – see **Fig2.** We call it items boosting. Items dwell times are different. To differentiate items accordingly, we propose to increase the presence in the session of those, having dwell time greater than the pre-defined threshold. Indeed, the presence remains at the same location in the sequence.

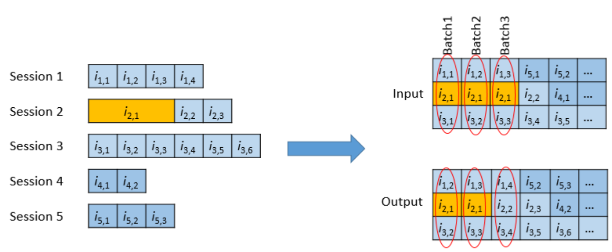

Figure 2. Dwell time based session-parallel mini batch creation using items boosting

Recently, Hidasi and Karatzoglou [6] (will refer to it later as *GRU4Rec with sampling*) improved the performance of the *GRU4Rec* by changing a sampling strategy: where for each example in the mini batch another example is used as a negative sampling; and presenting a novel family of ranking loss functions, based on individual pairwise losses. Their recall@20 and MRR@20 results outperformed *GRU4Rec*. We will also approach this method it in the next section of evaluation.

3 EVALUATION

Our proposed method was evaluated on RecSys'2015 challenge data set – Yoochoose [1]. Each e-commerce session is represented by clicks, including timestamp of the click. Clearly, dwell time for all clicks, except the last one in the session (as there is no dwell time for the last click in the data set), where dwell time is not available, can be extracted. The dwell time is calculated as a difference between timestamp of the current item and the next one. Following Hidasi et al. research [5] , the one-click sessions from the original training dataset were dropped due their nature – the lack of next click to predict.

Yoochoose click data was split to training and test sets the same way as it was done in [5]. To be able to compare our results with the results of GRU4Rec the same metrics were used – recall@20 and MRR@20.

The statistics on the dwell time (in seconds as the reader can see on axis X) is presented as a *boxplot* in **Fig3.** Items dwell time distribution is presented in **Fig4**.

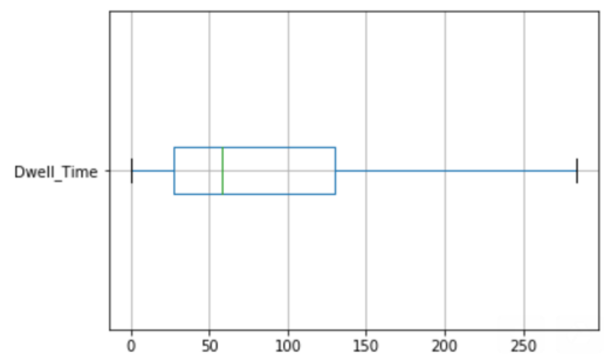

Figure 3. Boxplot with statistics on the data set's dwell time

The reader can see from the boxplot that the average dwell time is around 148 seconds, while the median is around 60 and standard deviation is 326.05. 25th percentile is 26 seconds, 50th - 58.5 seconds and 75th percentile - 130 seconds.

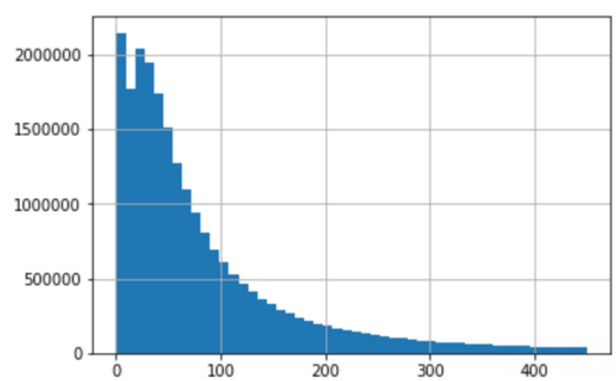

Figure 4. Dwell time distribution that is limited to items with dwell time less than 450 seconds.

We decided to test our approach initially with the threshold that is half of the average time – 75 seconds (from practical performance reasons), as well as between the 50th and 75th. Moreover, since we are curious about other threshold values, we have conducted an experiment with 100 seconds as a threshold and got worse results than with 75 seconds, as presented in Table 1. Looking at Table 1, the reader can see two baseline method's *GRU4Rec[5]* and *GRU4Rec with sampling[6]* results. Under each one of them we present results for the same method, but with the dwell time enrichment with the threshold set to 75 seconds and 100 seconds. According to the table, incorporating dwell time into *GRU4Rec with sampling*, using the threshold set to 75 provides the best results.

Table 1. Comparison between different results

Method	recall@20	MRR@20
GRU4Rec [5]	*0.5853*	*0.2305*
GRU4Rec with dwell time threshold 75	0.7885	0.5834
GRU4Rec with dwell time threshold 100	0.7754	0.548
GRU4Rec with sampling [6]	*0.7117*	*0.308*
GRU4Rec with sampling and dwell time threshold 100	0.84	0.61
GRU4Rec with sampling and dwell time threshold 75	**0.853**	**0.636**

We will continue to experiment and test different thresholds effects on the next item recommendation task.
The loss function values progression over epochs in the extended version of the algorithm, based on Rec4GRU with sampling [6], is presented in **Fig5**. The goal is to minimize the loss function for the training data. Therefore, the smaller the value of the loss function, the better our predictions. Frequently used loss function is the cross-entropy loss. In Rec4GRU – TOP1 loss was used [5]. However, in Rec4GRU with sampling - Bayesian Personalized Ranking (BPR)-max provided the best results.

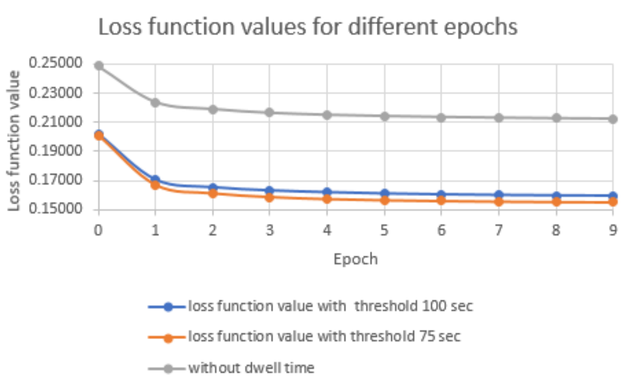

Figure 5. Loss function values for methods based on *Rec4GRU with sampling* with and without dwell time.

For each epoch in RNN the loss function value is outputted as well. On the first epoch0 the values are the worst and after that, slowly, it converges. When the threshold is set to 75 seconds, then the learning process is faster, since more data is available (relatively to *Rec4GRU with sampling*). As well given the fact that Hidasi et al. models were optimized.

4 DISCUSSION AND FUTURE WORK

We proposed and evaluated a method for incorporating dwell time in session-based recommendations with RNN for next item prediction. By multiplying items instances we increased the original training dataset that we had. As a result, we also improved recall@20 and MRR@20 by boosting significant to user items, based on their dwell time.
We showed that the newly suggested method enhances and outperforms the method suggested by [5] in a specific case study. This supports our claim that the recommendations of the next item are dynamic and depend also on the time user spends on the specific item. Moreover, they evolve over time.
In future work, we plan to explore the optimal threshold for the dwell time and check our findings on few more data sets, not only on RecSys'2015 challenge.

ACKNOWLEDGMENTS

The work is partially supported by the Israeli Innovation Authority, the Ministry of Economy and Industry, MAGNET "Infomedia" project.

REFERENCES

[1] Ben-Shimon, D., Tsikinovsky, A., Friedmann, M., Shapira, B., Rokach, L., & Hoerle, J. (2015). Recsys challenge 2015 and the yoochoose dataset. In Proceedings of the 9th ACM Conference on Recommender Systems (pp. 357-358). ACM.
[2] Bogina, V., Kuflik, T., & Mokryn, O. (2016). Learning Item Temporal Dynamics for Predicting Buying Sessions. In Proceedings of the 21st International Conference on Intelligent User Interfaces 251-255
[3] Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
[4] Greenstein-Messica, A., Rokach, L., & Friedman, M. (2017, March). Session-Based Recommendations Using Item Embedding. In Proceedings of the 22nd International Conference on Intelligent User Interfaces (pp. 629-633). ACM.
[5] Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939.
[6] Hidasi, B., & Karatzoglou, A. (2017). Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. arXiv preprint arXiv:1706.03847.
[7] Jannach, D., Kamehkhosh, I., & Lerche, L. (2017, April). Leveraging multi-dimensional user models for personalized next-track music recommendation. In Proceedings of the Symposium on Applied Computing (pp. 1635-1642). ACM.
[8] Park, S. E., Lee, S., & Lee, S. G. (2011). Session-based collaborative filtering for predicting the next song. In Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on(pp. 353-358). IEEE.
[9] Tan, Y. K., Xu, X., & Liu, Y. (2016, September). Improved recurrent neural networks for session-based recommendations. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (pp. 17-22). ACM.
[10] Yi, X., Hong, L., Zhong, E., Liu, N. N., & Rajan, S. (2014). Beyond clicks: dwell time for personalization. In Proceedings of the 8th ACM Conference on Recommender systems, 113-120.