Modeling User Session and Intent with an Attention-based Encoder-Decoder Architecture

Pablo Loyola*
The University of Tokyo
Tokyo, Japan
pablo@weblab.t.u-tokyo.ac.jp

Chen Liu
Rakuten Institute of Technology
Tokyo, Japan
chen.liu@rakuten.com

Yu Hirate
Rakuten Institute of Technology
Tokyo, Japan
yu.hirate@rakuten.com

ABSTRACT

We propose an encoder-decoder neural architecture to model user session and intent using browsing and purchasing data from a large e-commerce company.

We begin by identifying the source-target transition pairs between items within each session. Then, the set of source items are passed through an encoder, whose learned representation is used by the decoder to estimate the sequence of target items. Therefore, as this process is performed pair-wise, we hypothesize that the model could capture the transition regularities in a more fine grained way. Additionally, our model incorporates an attention mechanism to explicitly learn the more expressive portions of the sequences in order to improve performance. Besides modeling the user sessions, we also extended the original architecture by means of attaching a second decoder that is jointly trained to predict the purchasing intent of user in each session. With this, we want to explore to what extent the model can capture inter session dependencies.

We performed an empirical study comparing against several baselines on a large real world dataset, showing that our approach is competitive in both item and intent prediction.

KEYWORDS

Recommender Systems, Recurrent Neural Networks, Attention Mechanisms

1 INTRODUCTION

Recommender systems have received a vast amount of interest in recent years, as a natural consequence of the increasing complexity and scale of web services and e-commerce platforms. These models have varied over time, but in general can be classified as *behavior-based* or *content-based*. The former takes historical data from browsing and purchasing activity, considering an explicit time component from which predictive models can be computed. On the other hand, content-based approaches generate a feature space from where items can be mapped and their similarity is derived by means of an ad-hoc metric. Moreover, approaches such as collaborative filtering [17] aimed to combine both trying to construct item relationships based on co-purchasing activity.

Besides choosing the type of method to generate the recommendations, there is also an inherent challenge on how the data is represented and how we feed it to the chosen model. In any statistical learning setting, it is known that the representation of the data has a considerable impact on performance, and that different representations can entangle different explanatory factors [3]. We consider this issue is specially relevant in e-commerce, as we are in the presence of a highly multimodal and usually noisy setting, where a single item can be represented from different perspectives, such as its purchasing activity, content, rating or reviews information, among others. Finding meaningful representations for users is also challenging, given the heterogeneity of sessions and difficulty on estimating motivations from indirect sources.

Recently, the success that representation learning approaches have achieved on fields such as computer vision and speech recognition has naturally motivated their application in a recommender setting. These methods aim at automatically learn distributed feature representations from the data as an alternative to handcrafted feature engineering [8]. Within this paradigm, recurrent neural networks (RNN) have emerged as powerful tools to model sequential data [6]. Recent approaches on recommender systems have incorporated these models, mainly for user session prediction [10]. While these models have provided relevant insights, we consider there is room for improvement, as session prediction is inherently a sequential problem. Moreover, from our exploratory analysis using standard recurrent architectures for session modeling, we found that they worked well when the number of items remains relatively small, But when we tested in a more realistic setting, increasing the dataset impacted negatively on the performance.

As an alternative to current models that use one network to model sessions, in this work we propose to explore a neural architecture based on the encoder decoder paradigm [21], which was initially designed for machine translation [4]. This model consists of two recurrent neural networks that work in a symbiotic way. We begin by identifying the source-target transition pairs between items within each session. Then, the set of source items are passed through an encoder, whose learned representation is used by the decoder to estimate the sequence of target items. Therefore, as this process is performed pair-wise, and both networks are jointly trained end-to-end, we hypothesize that this model could capture the transition regularities in a more comprehensive fine grained way than using a single network. At the core of our model we incorporated an attention mechanism to explicitly learn the more

*Work done while the author was an intern at Rakuten Institute of Technology

RecSys'17, August 27–31, 2017, Como, Italy
© 2017 ACM. ISBN 978-1-4503-4652-8/17/08…$15.00
DOI: http://dx.doi.org/0.1145/3109859.3109917

expressive portions of the sequences in order to improve performance.

While characterizing the distributional regularities of the transition within sessions represents a relevant input for generating recommendations, we consider that it is necessary to incorporate the user intent, for instance, if the user will just browse or perform a purchase within the session. We consider this factor of high relevance, as it provides an explicit feedback and more comprehensive context. To explore the feasibility of that idea, we attached a second decoder to the proposed model, which is jointly trained to estimate the likelihood of a purchase event for a given session. With this, we are also interested in assessing the flexibility of the architecture in the sense of its ability to incorporate other data sources to complement user modeling.

We performed an empirical study on user activity from a large scale e-commerce platform. We compared the performance of our model against three baselines, one of them a recurrent architecture. The results show that our model is competitive, especially as we increase the item space, where baseline performance decreases. Additionally, results on intent prediction suggest the feasibility of the idea, opening a new opportunity for further research.

In summary, we proposed a recurrent architecture that is able to provide relevant item predictions at session based level and it is flexible enough to incorporate additional tasks, such as intent estimation.

2 RELATED WORK

Recent advances in representation learning mainly from vision and natural language processing have been adapted to an e-commerce setting, with the goal of improving the accuracy of product recommendations [8].

In terms of how to generate recommendations through representation learning, literature shows attempts in categories ranging from a content-based and collaborative filtering to temporal and hybrid models. The main theme covering these methods is the construction of a feature learner to obtain unified user-item representations, which are then used for recommendations.

One of the first approaches was proposed on the field of music recommendation, where van den Oord et al [23] developed a model inspired on previous work in feature learning from audio signal [9] to learn latent factors which are used for training a classifier. In [25], Wang et al. proposed *collaborative deep learning*, a Bayesian model that combine both feature learning from product content information and collaborative filtering from the user ratings.

In a more recent work, Covington et al [7] used a deep architecture to learn unified representations for users and videos to generate recommendations on Youtube. In this case, taking into consideration factors such as scale, freshness and noise, the authors propose a system conformed by two neural networks, the first one in charge of choosing candidates and the second one is used to rank them. Both networks receive embeddings of learned features which passes through a series of non-linearities.

The problem of learning combined representation from users and products has been also studied by Zheng et al in [26], where the authors focused on unifying item reviews to learn item features and the history of reviews written by users to learn user behaviour. The learned representations are merged in a share layer that match users and items to perform recommendations.

The second line of research consists of considering the time as the main component and modeling the sequences of items the user visits. From understanding that sequence, these models are able to provide a prediction about the next item. In the case of using recurrent neural networks, we can mention the work by Hidasi et al [10], where the authors proposed a recurrent architecture to generate session-based recommendations, specifically by training a gated recurrent unit (GRU) based network [4] through a session parallel mini-batch process to avoid the need to restrict session length. Additionally, they presented a loss function that considers the ranking of relevant items. The same authors proposed an extension in [11] in which they incorporate other dimensions of product information such as image features, by implementing a parallel recurrent architecture, on which the information is merged to perform a more comprehensive prediction. More recently, Tan et al [22] proposed an extension to [10] based on data augmentation and adding the ability to directly predict embedding vectors instead of item id.

In [19], Song et al proposed an architecture that combines both long-term and short-term temporal user preferences which are modeled through different stepwise long short term memory (LSTM) networks [12].

3 PROPOSED APPROACH

We assume the availability of a set of user activity, both in terms of browsing sessions and purchasing events. Our goal is to develop a model capable of computing session based recommendations, by predicting the next item based on the past activity.

We rely on a recurrent architecture to learn feature representations and encode the transition dependencies at a distributional level.

Let us assume a sequence of items representing an user session in the form $s = \{i_1, i_2, \ldots, i_N\}$. From that, we define a transition as the pair of two consecutive items, (i_{j-1}, i_j), where the first element is called *source* and the second, *target*. Then, the set of item transition pairs for s, P_s will be given by $P_s = \{(i_1, i_2), (i_2, i_3), \ldots, (i_{N-1}, i_N)\}$. We subsequently obtain the group of sources $S_s = \{i_1, \ldots, i_{N-1}\}$ and targets $T_s = \{i_2, \ldots, i_N\}$. Additionally, for each session, we can obtain a binary feature indicating if there is a purchasing event or not.

Therefore, we can formalize our problem as trying to learn a model M, such as $M : S_s \longrightarrow T_s$. As this can be seen as a sequential modeling problem, a recurrent model can be trained for this task in a way that we can find a set of parameters θ that maximizes the likelihood of a element in the target set t_i given an input coming from the sources set s and the currently history of sequence , t_1^{i-1}, $\prod P(t_i | t_1^{i-1}, s; \theta)$. We propose to parametrize model M with an encoder-decoder architecture, more commonly known as *sequence to sequence* [21], which was initially proposed for neural machine translation [5], but recently its use has been expanded successfully outside the field of natural language processing[24].

The generic architecture assumes two recurrent neural networks working together [15]. The first one, the encoder, receives the input sequence and generates a fixed length representation $c =$

$q(h_1, \ldots, h_T)$, where q is a non-linear function and h_i is the hidden state of the encoder at time i.

For the encoder, a bidirectional RNN [18] is used, which reads the sequence forward and backwards, generating two sets of hidden states. Therefore, the representation for a given internal state h_j is the concatenation of both vectors, $h_j = [\overrightarrow{h_j}; \overleftarrow{h_j}]$. In our setting, the encoder reads the sequence of elements coming from the *source* set. With this, we are able to obtain a fixed length representation of the whole sequence.

Then, the second network, the decoder, takes the output of the encoder and uses it as the initial state from which it starts to estimate the sequence of elements coming from the *target* sequence. In our setting, the decoder RNN is unidirectional. This network works by, at each time step i, computing its internal states as $d_i = f(d_{i-1}, c_i)$, where d_{i-1} represents the decoder state at time i_1. We also treat c_i as an *attention* mechanism [2], to encapsulate the portions of the *source* sequence that has more expressive power. In this case, this vector is the result of the weighted sum of all encoder internal states:

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j \tag{1}$$

where the weights $\alpha_{i,j}$ are learned by an additional feed forward neural network that is jointly trained with the rest of the components of the system:

$$\alpha_{i,j} = \frac{exp(e_{i,j})}{\sum_{k=1}^{T} exp(e_{i,k})} \tag{2}$$

$$e_{i,k} = g(s_{i-1}, h_k) \tag{3}$$

where g is parametrized with a feed forward neural network. This configuration is shown in the upper part of Figure 1, where A represents the attention mechanism the computes c_i.

Finally, we experimented with an explicit transfer of information between the two networks, which is shown the bottom part of Figure 1. We called this process *alignment* and consists of passing both y_{i-1}, the previous emitted label associated to the set of target items (represented by a purple line) and h_i, internal state computed by the encoder at time i, to the decoder. Note the that we are making an explicit alignment between internal states from both networks and therefore the expression for the decoder state will be $d_i = f(d_{i-1}, y_{i-1}, h_i, c_i)$.

3.1 Adding Session Intent

While being able to represent and estimate the distributional characteristics of item transitions among sessions is our primary goal, another key element that could provide additional and useful insights for user understanding is the actual action that the user performed during the session under study or on the future activity. In that sense, of special interest is to know if the user is likely to *purchase* or just *browse*. We define those two actions as part the *intent* of the user. In order to incorporate this additional element, we propose to modify the encoder-decoder architecture by adding a second decoder, which is also jointly trained with the rest of the system. This second decoder takes the output from the encoder and it is configured to output only one element, which is associated to the class *purchase* or *browse* and can be modeled as a binary variable.

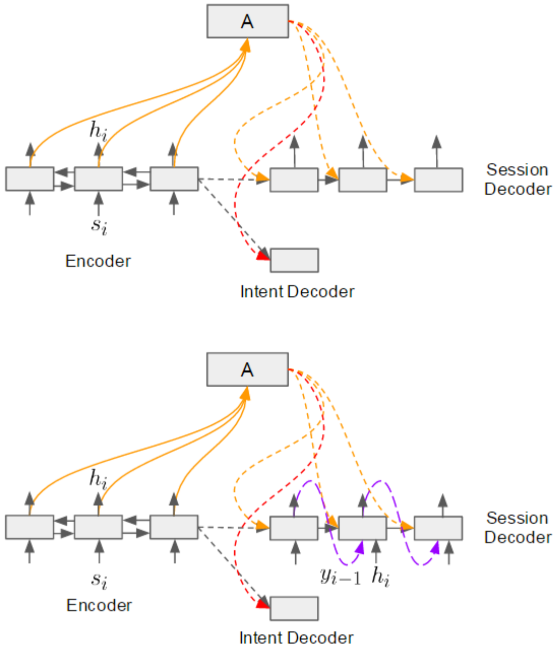

Figure 1: Proposed model for session and intent modeling with attention (top) and attention + alignments (bottom).

It must be noted that instead of RNN for this element, we could have just used another feed forward network, because strictly we are not trying to generate sequential output. But our exploratory analysis showed that in fact the use of a recurrent architecture performed better. We hypothesize this is because as the cost of both decoders are backpropagated jointly to the encoder at training time, costs from the same nature could generate a smoother learning signal. However a more comprehensive answer might require further analysis.

For the *intent* decoder, we also compute a context, which is passed along with the output from the encoder. The resulting architecture can be seen in Figure 1, where the red line represents the contribution from the attention mechanism . In terms of the intent we want to model, we propose to try two configurations. In the first place, we try to predict the intent in the current session, which means, just from the sequence transitions, estimate of any of the elements in the sequence will be purchased or not. For the second configuration, we obtained the intent from the next session that the user generated. With this, we wanted to explore if it is possible for the model to relate events that are temporally delayed, i.e., if there is a connection between the sequence of elements explored and the action in the immediate future.

4 EMPIRICAL STUDY

We designed and implemented and empirical study to test the validity and practical use of the proposed approach.

4.1 Data

We obtained a subset of the user activity from a large scale e-commerce system. Our total set consists of 9 million user sessions. Each session is conformed by the sequence of items browsed by the user and their respective time stamp, session id and user id. The set of unique items contained in our dataset reaches the approximately 400.000. We also collected the purchasing activity within the same period of time, allowing us to compute the intent of each session.

We divided the full dataset into training, and testing using a 80:20 proportion. From the training set, we took 10% as a validation set, which was used for hyper parameter selection.

4.2 Baselines

We compare the proposed approach against three baselines:

Item-KNN: This method provides an item recommendation based on its co-occurrence with other items along sessions and it is computed as a function of the number of co-occurrences of item pairs within sessions and the number of session that contains that item pair.

BPR-MF: Matrix factorization method that optimizes a pair wise ranking function through stochastic gradient descent [16]. We followed [10] in the sense of obtained an estimation of the feature representation for each item through averaging past occurrences.

GRU4Rec: Method proposed by Hidasi et al [10], which consists of a GRU-based architecture that iteratively read session items predicting the next one through a pair wise ranking function.

For each method we computed two metrics to measure performance. In the first place, Recall20, which provides the proportion of predictions which contain the ground truth among the top-20 items. Additionally we reported the mean reciprocal rank (MRR), which is a more strict metric as it explicitly considers the order of a set of candidates provided by a model. In this case we also consider a top 20 ranking.

4.3 Experimental Settings

We implemented our approach on top of the Tensorflow library [1]. Hyperparameter optimization was conducted on the validation set where we explored variations on the type of cell used (LSTM or GRU) and the number of units. From that process we decided to use LSTM with 100 units. For training we set the batch size on 32 and a Dropout [20] of 0.5 as regularization. We experimented with batch normalization [14], but we did not observe relevant improvement, while it had a considerable overhead on the time consumption. We tested several optimization methods, and the best cost-effective performance was obtained with Adam [13].

In the case of the baselines, we adapted a publicly available implementation [10]. In the case of GRU4Rec, we studied variations related to the number of hidden units and the loss function. We used the same validation set to perform hyperparameter optimization.

5 RESULTS AND DISCUSSION

Item session prediction results are shown in Table 1, where we called our method EDRec. From them, we can see that the proposed approach on average outperforms the presented baselines. Specifically in terms of MRR, we observe that the model tends to order the resulting candidates in a more effective way than the other approaches. We hypothesize that the memory component of the recurrent models allows them to retain more effectively the order, and, in the particular case of the proposed encoder-decoder architecture, the use of additional matching mechanisms such as alignment and attention increase the chances of model long term dependencies.

Table 1: Recall and MRR for the different approaches presented and their variations.

Model	Recall@20	MRR@20
Item-KNN	0.327	0.139
BPR-MF	0.310	0.135
GRU4Rec	0.3481	0189
GRU4Rec (cross-entropy)	0.3506	0.207
EDRec	0.3775	0.214
EDRec w/ alignment	0.3905	0.249
EDRec w/ alignment and attention	0.3914	0.231

Regarding the intent estimation, our results show an accuracy of the 67 % when we try to predict the intent in the current session, but it drops to 61.2% when we try to estimate the intent of the next session for a given user. We think these results are caused due to the natural imbalance of the data, as the purchase intent is seen in approximately the 30% of the sessions in the dataset.

We also studied the change in the performance as we increase the amount of data, both in terms of number of feasible items and also number of sessions. In this case, we observed that, on our dataset, initially both GRU4Rec and our approach behave similarly, but when we start using approximately 70% of our dataset, GRU4Rec performance showed no increment, while our model performance kept increasing up to the point we used our entire dataset. We consider that this partial result that could enable a deeper analysis in the future related to the scalability of the models, and how to allow them to keep learning when the possible space of items to predict become extremely large.

Another element worth mentioning is the time associated to the execution of the studied models. In this case, we can see a clear overhead in the proposed approach, requiring up to two times more time to reach convergence. We consider this fact as a natural consequence of the more complex mechanics of the model. Therefore there is an explicit trade-off to be considered.

6 CONCLUSION AND FUTURE WORK

In this work, we proposed an encoder-decoder architecture for modeling user sessions in an e-commerce setting. Our initial study shows that our approach is able to reach state of the art results, while it is flexible enough to incorporate additional task, such as the way we propose to jointly predict the user intent. For future work, we consider necessary extending the scale of the empirical study in order to provide a better support and also explore if more data dimensions could be incorporated to enhance the analysis.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and others. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

[4] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 191–198.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT Press.

[9] Philippe Hamel and Douglas Eck. 2010. Learning Features from Music Audio with Deep Belief Networks.. In *ISMIR*, Vol. 10. Utrecht, The Netherlands, 339–344.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[11] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 241–248. DOI: http://dx.doi.org/10.1145/2959100.2959167

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[13] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] César Laurent, Gabriel Pereyra, Philémon Brakel, Ying Zhang, and Yoshua Bengio. 2016. Batch normalized recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2657–2661.

[15] Bing Liu and Ian Lane. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. *Interspeech 2016* (2016), 685–689.

[16] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[17] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[18] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[19] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 909–912.

[20] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[21] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

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

[23] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*. 2643–2651.

[24] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*. 2692–2700.

[25] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[26] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, New York, NY, USA, 425–434. DOI: http://dx.doi.org/10.1145/3018661.3018665