A Variational Recurrent Neural Network for Session-Based Recommendations using Bayesian Personalized Ranking

Panayiotis Christodoulou *paa.christodoulou@edu.cut.ac.cy*
Cyprus University of Technology
Limassol, Cyprus

Sotirios P. Chatzis *sotirios.chatzis@cut.ac.cy*
Cyprus University of Technology
Limassol, Cyprus

Andreas S. Andreou *andreas.andreou@cut.ac.cy*
Cyprus University of Technology
Limassol, Cyprus

Abstract

This work introduces VRNN-BPR, a novel deep learning model, which is utilized in session-based Recommender systems tackling the data sparsity problem. The proposed model combines a Recurrent Neural Network with an amortized variational inference setup (AVI) and a Bayesian Personalized Ranking in order to produce predictions on sequence-based data and generate recommendations. The model is assessed using a large real-world dataset and the results demonstrate its superiority over current state-of-the-art techniques.

Keywords: Neural Networks, Latent Variable Models, Deep Learning Recommender Systems.

1. Introduction

Recommender Systems (RS) are currently being used as smart mechanisms in e-commerce websites aiming to enhance user experience and provide personalized recommendations [1, 4]. Recent studies on RS have been focused on a series of matrix factorization and neighborhood models that work well in cases where a clear user profile may be built; this leads to overcoming fundamental problems met in RS, like the cold-start and data sparsity [2]. Nowadays, researchers need to examine the use of RS in a wider spectrum of applications aiming to produce accurate recommendations for application domains that have never been addressed in the past or for cases described by complex empirical samples, as for example session-based data. Session-based recommendations is a recent challenge in the area of RS introduced in the RecSys Challenge of 2015 [10]. In this context a RS provides recommendations taking into consideration only the actions of users in the current browsing session [16]. This type of recommendations processes the historical data of users captured during an active session and relying only on a narrow piece of information describing the behavior of a specific user it predicts her/his next move and recommends an item.

The success of deep neural networks when dealing with image or speech recognition [3] was the initial starting point for incorporating such models in RS so as to deal with unstructured data and produce accurate session-based recommendations. In addition, the use of Recurrent Neural Networks (RNNs) for modeling variable-length sequence data also gained attention in the past few years and was utilized to tackle a session-based recommendation problem with remarkable results [10]. RNNs introduce a hidden state in the units in order to compose a network; this is in fact the main difference between RNNs and other feed-forward deep models. In a session-based recommendation problem the RS first considers the first item that a user selects when accessing a webpage as the initial input of the RNN and then each other sequential click produces a recommendation (output) that relies upon the previous clicks. The main challenges in this case are the set of items to choose from, which can be millions in most situations, and the training time and scalability issues that arise due to the fact that the click-stream datasets are enormous. In order to deal with the aforementioned challenges the systems use a ranking loss function to train the RNN and are mostly focused on recommending only a set of the top-items a user may be interested in.

This paper presents a framework that is called Variational Recurrent Neural Network for Session-Based Recommendations using Bayesian Personalized Ranking (VRNN-BPR) with the aim to increase the benefits of a RNN session-based RS and also deal with data sparsity. The proposed methodology is inspired by recently proposed RS systems which use Bayesian inference techniques [17]. In a RS environment Bayesian inference considers the system variables with some prior distribution on them; this helps the recommendation engine to deal with uncertainty over the sparse data and produce improved predictive results. Furthermore, the amortized variational inference paradigm (AVI) is used [18] to enable scalability with real-world datasets, which is demonstrated on a publicly available dataset. Finally, the performance of the VRNN-BPR model is compared against state-of-the-art techniques and the results suggest that the proposed model performs better without suffering from scalability issues.

The remainder of the paper is structured as follows: The next section provides an overview of related work, while section 3 presents the proposed model. Section 4 evaluates the model against various state-of-the-art methods and presents the outcomes. Finally, in the concluding section the contributions of the paper are summarized and future steps are briefly discussed.

2. Related Work

Neighborhood and matrix factorization models are extensively used in RS proposed in the relevant literature. Russakovsky et al. [4] explore the item-based collaborative filtering techniques to address the key challenges of RS. Item-based approaches analyze the user-item matrix in order to discover correlations that exist between the different items and then use these relationships to calculate the recommendations for each user. A list of various methods for computing similarities between items and then using these similarities for obtaining recommendations are examined in literature [2,4,5], with evaluation results indicating that item-based approaches outperform user-based algorithms in terms of quality.

Koren et al. [5] present a set of latent factor models based on matrix factorization. Matrix factorization techniques characterize both users and items by vectors of factors inferred from item rating patterns; these techniques also combine good scalability with high accuracy and are used to model real-world scenarios. One of the strengths of these techniques is that they allow additional information so when explicit ratings are not available RS can infer user preferences using implicit means. Except from the different forms of feedback, matrix factorization models can integrate many vital aspects of data, such as temporal dynamics and confidence levels; this is the main reason that these approaches yield better results compared to classic neighborhood models.

The work resented in [6] argues that it is more suitable to view the problem of producing recommendations as a sequential optimization problem and that the use of Markov decision processes (MDP) can provide a better model ready to be used in RS. MDP take into account the long-term effects and the expected value of each recommendation. The proposed MDP model described in [6] performs better than the classic Markov Chain model when deployed on a real commercial website.

Deep learning techniques have been applied with high success in a wide area of applications like image and speech recognition [7]. A recent method called Collaborative Topic Regression was used to couple two components that can learn from various sources of information in order to deal with the problem that the latent representation learned from a regression model is not very useful when the additional information is very sparse. The work of Wang et al. [8] presents a hierarchical Bayesian model called Collaborative Deep Learning (CDL) for RS to address the data sparsity problem. The proposed approach performs deep representation learning between the content information and the collaborative filtering ratings matrix. Experiments on real-world datasets from various domains demonstrate that the use of CDL advances the state-of-the-art methods.

The study of Salakhutdinov et al. [9] claims that most of the existing collaborative filtering methods cannot deal with very large datasets and presents a class of two-layer undirected graphical models called Restricted Boltzmann Machines (RBM) that can be used to model tabular data. A set of efficient learning and inference approaches are presented for the RBM model and applied on the Netflix dataset. Evaluation results showed that the RBM approach outperformed the Singular Value Decomposition (SVD) models.

Zhang et al. [11] argue that click prediction is one of the main problems in sponsored search and that most works in the literature make use of machine learning approaches in order to predict ad click for each event. In a real-world system users' behavior on advertisements depends on how they behaved in the past. The authors introduce a novel framework based on a Recurrent Neural Network (RNN) network. The proposed RNN was evaluated using the click-through logs of a large scale commercial engine and the results indicated significant improvements on the click prediction accuracy compared to sequence-independent approaches. Moreover, Hidasi et al. [10] present a RNN network applied on a challenge that most real-world RS face, that is, how to deal with long session-based data that exist on large ecommerce websites, and hence, with data sparsity. In this kind of problems the frequently used matrix factorization approaches are not accurate enough so modeling the whole session can result in more accurate recommendations to users. The proposed methodology introduces several modifications to classic RNNs, such as the Gated Recurrent Unit (GRU) and a ranking loss function, but at the same time it considers practical aspects of the task. Evaluation is performed on two datasets, the first being that of the RecSys Challenge 2015 and the second one was collected from the OTT video service platform. Experimental results show that the suggested approach outperforms the best baseline method of item-KNN on such problems and that further improvements can be performed when adjusting the RNN parameters and changing the loss function. Finally, the paper by Tan et al. [16] further analyzes RNN-based models for session-based RS and proposes two techniques that lead to improvements on the models' performance. The present work was evaluated using the dataset introduced in the RecSys Challenge 2015 and the results were compared against the models presented in [10]; the proposed methodology outperformed the baseline algorithms by 12.8% and 14.8% on the Recall@20 and MRR@20 metrics.

3. Proposed Approach

The main contribution of this work lies with the use of a machine learning model capable of extracting temporal dynamics from sparse user session data and then utilizing this information to produce accurate recommendations. The VRNN-BPR formulates the generation session-based recommendations as a sequence-based prediction problem. Let us denote with $\{x_i\}_{i=1}^{n}$ a user session where x_i is the i^{th} clicked item; then, we may formulate session-based recommendations as the problem of predicting the score vector $y_{i+1} = [y_{i+1,j}]_{j=1}^{m}$ of the available items, where $y_{i+1,j} \in R$ is the predicted score of the j^{th} item. We are interested in recommending more than one item, therefore, at each time point we select the *top-k* items to recommend back to the user. The core approach of this work is the utilization of a deep learning model for predicting accurately vector y_{i+1}.

3.1. Methodological Background

Our model is inspired by the state-of-the-art RNN-based method presented in [10] and relies on an RNN structure that utilizes GRU units. The network displays the current user action at each time step *i* and expects to predict a score vector for the next user selection. The recurrent units' activation vectors of the GRU-based network, ***h***, are updated at time *i* using the following formula:

$$h_i = (1 - z_i) \cdot h_{i-1} + z_i \cdot \hat{h}_i \tag{1}$$

where h_{i-1} is the activation vector of the recurrent unit at the previous time point and $\hat{h}_i$ is the candidate activation vector of the GRU units:

$$\hat{h}_i = \tanh(Wx_i + U(r_i \cdot h_{i-1})) \tag{2}$$

In eq. (1) above, z_i is the update gate output, which essentially controls when and to what extent an update to a hidden state of the recurrent units should be made. z_i is given as

$$z_i = \tau(W_z x_i + U_z h_{i-1}) \tag{3}$$

where τ is the logistic sigmoid function and r_i is the output of the reset gate of the GRU network (see eq. (4)), which is trained to decide when the internal memory of the GRU units must be reset.

$$r_i = \tau(W_r x_i + U_r h_{i-1}) \tag{4}$$

W, *U*, W_z, U_z, W_r and U_r in the aforementioned equations are trainable network parameters.

3.2. Model Formulation

The VRNN-BPR model extends the principles discussed in the previous section and introduces a novel approach that renders the GRU-based model manageable by a Bayesian inference. We consider the component recurrent unit activations as stochastic latent variables and impose a prior distribution over them:

$$p(h_i) = N(h_i \mid 0, I) \tag{5}$$

where $N(\xi \mid \mu, \Sigma)$ is a multivariate Gaussian density with mean μ, covariance matrix Σ and identity matrix *I*.

Also, the sought posteriors *q(h)* take the form of Gaussians with means and isotropic covariance matrices parameterized via GRU networks as follows:

$$q(h_i; \theta) = N(h_i \mid \mu_\theta(x_i), \sigma_\theta^2(x_i) I) \tag{6}$$

In the above formula the mean vectors $\mu_\theta(x_i)$ and the variance functions $\sigma_\theta^2(x_i)$ are outputs of the GRU network with parameters θ so we now have:

$$[\mu_\theta(x_i), \log \sigma_\theta^2(x_i)] = (i - z_i) \cdot [\mu_\theta(x_{i-1}), \log \sigma_\theta^2(x_{i-1})] + z_i \cdot \hat{h}_i \tag{7}$$

where

$$z_i = \tau(W_z x_i + U_z[\mu_\theta(x_{i-1}), \log \sigma_\theta^2(x_{i-1})]) \tag{8}$$

$$\hat{h}_i = \tanh(Wx_i + U(r_i \cdot [\mu_\theta(x_{i-1}), \log \sigma_\theta^2(x_{i-1})])) \tag{9}$$

$$r_i = \tau(W_r x_i + U_r[\mu_\theta(x_{i-1}), \log \sigma_\theta^2(x_{i-1})])) \tag{10}$$

and [ξ, ζ] presents the concatenation of vectors ξ and ζ. The values of the latent variables h_i can be computed by posterior samples from the inferred posterior density.

Let us continue with the output layer of the proposed model. Item ranking [10,12,13] can be pointwise, pairwise and listwise. In the proposed model we use the matrix factorization method Bayesian Personalized Ranking (BPR) [13] that uses a pairwise ranking loss. This technique compares the score of the positive item with several sampled items and then utilizes their average as the loss. The loss is defined at a given point in a session as follows:

$$L_s = -\frac{1}{N_s} \cdot \sum_{j=1}^{N_s} \log(\sigma(\hat{r}_{s,i} - \hat{r}_{s,j})) \tag{11}$$

where *i* is the next item in the session, *j* are the negative samples, N_s is the sample size and $\hat{r}_{s,k}$ is the score on item *k* at a given point.

3.3. Training Algorithm

The variational inference of the proposed model consists of the maximization of a lower-bound to the log-marginal likelihood [21]. Based on the model formularization, the ELBO expression of the VRNN-BPR is:

$$\log p(D) \geq \sum_i \{-KL[q(h_i;\theta) \,||\, p(h_i)] - E[L_s]\} \tag{12}$$

where $KL[q \,||\, p]$ is the KL divergence between the distribution *q* and the distribution *p*. More specifically,

$$KL[q(h_i;\theta) \,||\, p(h_i)] = -\frac{1}{2}\sum_{d=1}^{D}[\mu_\theta(x_i)^2]_d + \frac{D}{2}[1 + \log \sigma_\theta(x_i)^2 - \sigma_\theta(x_i)^2] \tag{13}$$

Unfortunately, the posterior expectation $E[L_S]$ cannot be computed analytically. This is due to the non-conjugate formulation of the proposed model, which stems from its nonlinear assumptions. As a result, training the entailed parameter sets θ is not possible. AVI resolves these issues by means of a smart re-parameterization of the Monte Carlo [22] samples of a Gaussian posterior density [18]. The MC samples resulted from the posterior density can be now expressed as differentiable functions of the parameters sets θ and some random noise variance ε. As a result, the problematic posterior expectation $E[L_S]$ is not reduced to a random noise variable. To this end [18] suggests the use of Adagrad as the stochastic gradient algorithm [14], so we also follow this advice and select Adagrad as the stochastic optimizer for training the proposed model.

4. Experimental Evaluation

The proposed model was developed and trained in Theano [15] on an Intel Xeon 2.5GHz Quad-Core server with 64GB RAM and an NVIDIA Tesla K40 GPU accelerator. The model was evaluated using the dataset presented in the RecSys Challenge 2015, which contains the click-streams of an ecommerce website. We use only the click events of the training dataset and then split that dataset into training and test sets utilizing the same procedure as in [10]. The final datasets used in this work are formed as follows: The training dataset contains 7,966,257 sessions of 31,637,239 clicks on 37,483 items and the test dataset comprises 15,324 sessions of 71,222 click actions on the same amount of items.

4.1. VRNN-BPR Model Configuration

As already described in the previous section, the VRNN-BPR utilizes a Bayesian Personalized Ranking (BPR) loss function [13] and Adagrad [14] as the appropriate learning rate adaptation strategy. The proposed methodology also uses session-parallel mini-batch training and applies dropout of one value at each time step in order to reduce over fitting [19]. The model is trained using a predefined number of epochs in order to minimize losses and at the same time avoid randomizing the order of sessions in each epoch. The hidden state is set to zero when a session is completed. The proposed technique uses an Adaptive Normalization that transforms the time series into a data sequence from which properties are obtained and used in the normalization process [20]. Finally, during training the Nesterov momentum [23] is applied and the uniform distribution is used.

According to [10], calculating a score for every item in each step could make the RNN model scale. Taking this finding into consideration, it is essential to sample the output and calculate the score only for a small subset of items. In addition to the output, we must calculate scores for some negative samples and adjust the weights so that the output is highly ranked; therefore, items are sampled based on their popularity. Our methodology uses the items from the other training examples of the mini-batch as negative examples. The benefits of the aforementioned technique are that we can decrease more the computational time by omitting sampling and that the code is not that complex, hence matrix operations become quicker. According to [10], this method is based on a popularity-based sampling since the likelihood of an item being in the other training samples of the mini-batch is proportional to its popularity. Finally, as described in [10] and [16], adding additional layers does not improve the performance of the RNN model, thus our approach uses only a single recurrent GRU layer.

4.2. Performance Metrics

The accuracy of the recommendations was assessed using the same evaluation metrics as those presented in [10]. The main evaluation metric is called Recall@20 and represents all the cases that include the desired item between the *top-20* recommended items in all test cases; this metric does not take into account the rank of an item. The other metric used is the MRR@20 (Mean Reciprocal Rank), which describes the average of reciprocal ranks of the desired items and it is set to zero if the rank is above 20. It also takes into account the order of the item which is crucial in cases where the rank of recommendations matters.

4.3. VRNN-BPR performance

In this section the proposed VRNN-BPR model is compared against the best (baseline) algorithms presented in [10] and [16]. The best performing model in [10] that achieved the highest accuracy on the Recall@20 metric uses a BPR loss function; we refer to it as “GRU w/ BPR Loss” in Table 1. In addition, the best performing model from the same study that outperformed all other methods on the MRR@20 metric utilizes a TOP1 loss function; we refer to it as “GRU w/ TOP1 Loss”. M2 and M4 in Table 1 refer to the best performing approaches presented in [16].

Table 1. Comparison of the VRNN-BPR model against various baseline algorithms.

Method	Hidden Units	Recall@20	MRR@20
GRU w/ BPR Loss	1000	0.6322	0.2467
GRU w/ TOP1 Loss	1000	0.6206	0.2693
M2	100	0.7129	0.3091
M4	1000	0.6676	0.2847
VRNN-BPR	**750**	**0.7971**	**0.7845**

As depicted in Table 1, the proposed model achieved the best accuracy on both metrics with the following settings: A BPR loss function with 750 hidden units, 50 mini-batches, a dropout value of 0.5, a learning rate of 0.1 and momentum equal to 0.3. As it can be observed from Figure 1, the VRNN-BPR model outperforms previously reported results by an average of 21% on the Recall@20 metric and 284% on the MRR@20 metric.

Table 2 outlines the training and evaluation time of the VRNN-BPR model against the baseline methods of GRU w/ BPR Loss and GRU w/ TOP1 Loss presented in [10]. We run the baseline approaches on the same computer as the VRNN-BPR model using the graphics card. The overall time and prediction time of each click event of the VRNN-BPR model is 13% less than the GRU w/ BPR Loss method and 3.8% less than the GRU w/ TOP1 Loss method. The results obtained show that the proposed methodology significantly improves the accuracy of a RNN model when dealing with session-based recommendations. At the same time, it also increases the performance of a RNN model by producing recommendations quicker and without suffering from scalability issues.

Table 2. Training and evaluation times of the VRNN-BPR model compared with the baseline algorithms of [10].

Method	Training time (s)	Evaluation time (s)	Overall time (s)	Prediction time of each click event (s)
VRNN-BPR	43009.75	29.67	43039.42	0.604
GRU w/ BPR Loss	48692.48	0.516	48692.99	0.683
GRU w/ TOP1 Loss	44716.60	0.492	44717.09	0.627

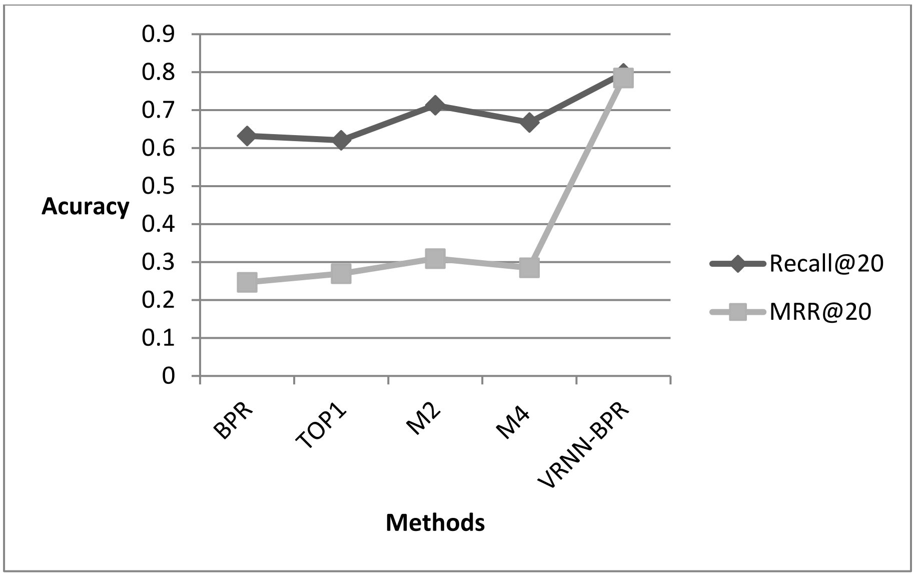

Fig. 1. VRNN-BPR accuracy against various baseline algorithms expressed with the Recall and MRR metrics on the top-20 recommended items.

4.4. Adjusting the size of hidden units

Table 3 presents how the accuracy of the VRNN-BPR model varies when we adjust the number of hidden units. The rest settings of the model remain the same as presented in section 4.3.

Table 3. Accuracy of the VRNN-BPR with different hidden units.

Hidden Units	Recall@20	MRR@20
100	0.5756	0.2127
500	0.7354	0.6793
750	**0.7971**	**0.7845**
900	0.7801	0.7563
1000	0.7760	0.7443
1250	0.7712	0.7318
1500	0.7629	0.7271
2000	0.7326	0.6737

As it can be seen from Figure 2, the accuracy of both metrics grows as the number of hidden units increases up to 750. When the value of hidden units goes beyond 750, there is a slight decrease in the accuracy of the system. We believe that this behavior is a direct manifestation of overfitting, which becomes stronger as the model grows excessively large.

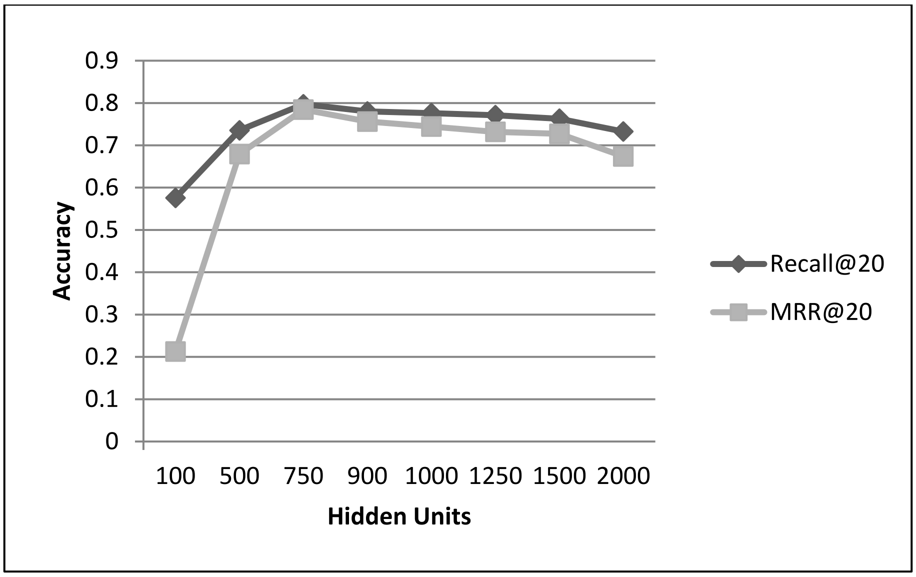

Fig. 2. Accuracy of the VRNN-BPR model for various hidden units expressed with the Recall and MRR metrics on the top-20 recommended items.

5. Conclusions

This paper proposed a novel approach that couples deep learning techniques with recurrent units to handle the increased complexity introduced in Recommender Systems by session-based datasets and tackle the problem of data sparsity. The proposed model, called VRNN-BPR, increases the benefits of a classic RNN session-based RS by introducing a variational inference concept in Bayesian Personalized Ranking for session-based recommendations. The combination of Bayesian inference with RNN networks utilizing GRU layers provides the ability to the model to analyze temporal patterns that exist in session-based data and at the same time enables it to deal with uncertainty in sparse data when producing predictions and recommendations. Experimental evaluation of the methodology was performed using various setups of our model and was assessed on a real-world dataset that was first introduced in the RecSys 2015 challenge. The proposed approach was compared against various other baseline approaches with the results clearly indicating that it performs better than the current state-of-the-art methods. In addition, this superior performance does not jeopardize computational effectiveness during training or prediction time. Future work will concentrate on validating our approach using long session-based datasets and on stacking multiple GRU layers on the proposed system in order to enable analysis of temporal dynamics in multiple time-scales.

References

1. Konstan, J. A., & Riedl, J.: Recommender systems: from algorithms to user experience. User Modeling and User-Adapted Interaction, 22(1-2), 101-123 (2012)
2. Ning, X., Desrosiers, C., & Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In Recommender systems handbook pp. 37-76. Springer US (2015)
3. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252 (2015)

4. Sarwar, B., Karypis, G., Konstan, J. and Riedl, J.: Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, pp. 285-295 (2001)
5. Koren, Y., Bell, R. M. and Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer, 42(8), pp. 30-37, (2009)
6. Shani, Guy, Brafman, Ronen I, and Heckerman, David.: An mdp-based recommender system. In Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, pp. 453–460 (2002)
7. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C. and Bengio, Y.: A recurrent latent variable model for sequential data. In Advances in neural information processing systems, pp. 2980-2988 (2015)
8. Wang, H., Wang, N., and Yeung, D.Y.: Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pages 1235-1244 (2015)
9. Salakhutdinov, R., Mnih, A. and Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th international conference on Machine learning, pp. 791-798 (2007)
10. Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D.: Session-based recommendations with recurrent neural networks. CoRR, abs/1511.06939, (2015)
11. Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., Wang, B. and Liu, T.Y.: Sequential click prediction for sponsored search with recurrent neural networks. arXiv preprint arXiv:1404.5772, (2014)
12. Steck, H.: Gaussian ranking by matrix factorization. In Proceedings of the 9th ACM Conference on Recommender Systems, pp. 115-122 (2015)
13. Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, pp. 452-461 (2009)
14. Duchi, J., Hazan, E. and Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul), pp.2121-2159 (2011)
15. Team, T. T. D., Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., ... & Belopolsky, A.: Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688, (2016)
16. Tan, Y. K., Xu, X., & Liu, Y.: Improved recurrent neural networks for session-based recommendations. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 17-22 (2016)
17. Sotirios P. Chatzis: A Coupled Indian Buet Process Model for Collaborative Filtering. In Journal of Machine Learning Research: Workshop and Conference Proceedings, Vol. 25: ACML 2012, pp. 65–79 (2012).
18. Kingma, D., and Welling, M.: Auto-Encoding Variational Bayes. In Proc. ICLR’14, (2014)
19. Gal, Y., & Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In Advances in Neural Information Processing Systems, pp. 1019-1027 (2016)
20. Ogasawara, E., Martinez, L. C., De Oliveira, D., Zimbrão, G., Pappa, G. L., & Mattoso, M.: Adaptive normalization: A novel data normalization approach for non-stationary time series. In Neural Networks (IJCNN), The 2010 International Joint Conference, pp. 1-8 (2010)
21. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., and Saul, L.K.: An introduction to variational methods for graphical models. In Learning in Graphical Models, M.I. Jordan (Ed.). Kluwer, Dordrecht, pp. 105–162 (1998)
22. Salakhutdinov, R. and Mnih, A.: Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. In Proc. ICML’11, (2011)
23. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural Networks 12, 1, pp. 145–151 (1999).