# EE491 - Online Recommendation in Temporal Interaction Networks

Shashank Gupta, 160643
IIT Kanpur
gshasha@iitk.ac.in

Suryateja B.V., 160729
IIT Kanpur
suryab@iitk.ac.in

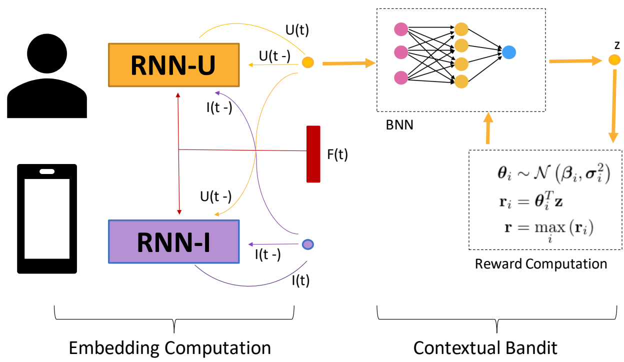


**Figure 1: Given sequential interactions $S_t = (u_t, i_t, f_t)$, our model learns to predict appropriate item to users at a future time. It consists of two parts - (i) Computation of user and item embedding trajectories following Kumar et. al's [20] work; and (ii) Using the user embedding as context vector in a contextual multi-arm bandit setting following Riquelme et. al's [28] work**

## ABSTRACT

**Advisors:** Prof. Ketan Rajawat (EE), Prof. Piyush Rai (CSE)

Modelling sequential interactions between two entities is crucial in various domains such as - social networks, e-commerce, education, and disease networks. For instance, wireless network selection problem could be seen as a set of sequential interactions between various users and available networks/items (WiFi, LTE, UMTS, LAN). While conventional recommender systems offer a simple solution to these problems, they do not model the temporal aspects that real-world instances demand. To this end, we focus on sequential recommender systems and tackle the problem by incorporating dynamic node embeddings and contextual multi-arm bandits together. Bandits, by definition, take up an online learning procedure suitable for modelling sequential interactions. Instead of using conventional feature vectors as context, the model we propose uses time-varying node embeddings. These embedding trajectories are learned by a combination of two RNNs and an attention layer, following [20]'s work. We evaluate our model on the MOOC dataset [24], and show promising results.


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

## KEYWORDS

networks, contextual bandits, semantic embeddings, temporal recommendation, Thompson sampling

**ACM Reference Format:**
Shashank Gupta, 160643 and Suryateja B.V., 160729. 2019. EE491 - Online Recommendation in Temporal Interaction Networks. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 RELATED WORK

While there is a fairly large volume of work associated to conventional recommender systems, interest in sequential recommender systems has grown only over the past decade. We cover various approaches to sequential prediction in the following paragraphs. For an extensive survey into this area, please refer to [42] or [41].
**Factorization models**: To incorporate temporal aspects, Dynamic Poisson Factorization method [3] models the user matrix to have a static part which is normally distibuted and a dyanmic part which has a Markovian state-space dependence capturing evolving interests. [4] introduces a dynamic Hierarchical Dirichlet Prior on the latent feature vectors to enable sharing of information between time-adjacent network snapshots. [12] extends DPF via an RNN formulation to account for long-term dependencies, and [8] integrates metadata such as social relations. However, these models do not cater to long-term dependencies and the time-splits aren't fine-grained. For example, three clicks per month could mean three clicks on a particular day, or 1 click on 3 different days.
**Temporal Point Processes**: TPPs [10, 17] model the time interval between events as random variables, and avoid the need to pick a

time window to aggregate events. They allow temporal events to be modeled in a more fine grained fashion, and have a remarkably rich theoretical support. For example, CoEvolve [10] models link prediction and information diffusion jointly, yielding better results. However, the intensity function is handcrafted, which requires domain knowledge. Some deep learning [6] and reinforcement learning techniques [22, 38] are proposed to overcome handcrafting, but these methods are extremely time-taking, making TPPs infeasible.
**Deep Learning**: To model higher order long-term dependencies, recent advances have centered on deep learning [50]. DL approaches are especially useful when there is an abundance of metadata such as texts, images, videos, and social relations of users.

*Recurrent Recommender Networks* [16, 43, 44, 48] capture temporal granularity well, and use deep models to generate text reviews which helps in better prediction. Session-based RNN models [14, 30].such as GRU4Rec, also incorporate context well and provide good performance in session-based settings. [15, 33] improve upon GRU4Rec by implementing parallel RNN architecture, data augmentation, model pre-training to account for temporal shifts in data distribution, and distillation using student-teacher setting to learn from small datasets. However, RNNs make strong model assumptions, as in, they assume relations to exist even among noisy data, and fail to capture extremely long-term dependencies since the gradients die out.

*Convolutional Networks* [35, 45, 46] treat the user-item matrix as an image, and apply stacked 1-D dilated CNNs to learn local features. These methods essentially focus on bringing tricks in visual domain, such as skip connections, into recommendation domain yielding impressive results. Due to the small size of filters, long-term dependencies are not captured well.

*Attention* based models [29, 49, 52] are extremely good at capturing long-term dependencies, and improve over RNNs by being very selective to noise. The transformer architecture in BERT4Rec [32] model enables bidirectional flow of information, and the formulation as a cloze task - that is, randomly masking certain items in the chain, helps in improving over predictions of GRU4Rec. Attention models are also attractive due to their interpretablity [51]. Inspired from NLP, [36] learns a latent knowledge graph over interaction networks, and a relation vector between user and item using neural attention. Infact, the model captures temporal aspects despite not training for it.

Other deep learning techniques include - TransRec [13], which models the next item a user interacts as a translation of the current item; Hierarchical Gating Network [25], which breaks the feature vectors into sub-vectors that encode different information; Mult-VAE [18], which incorporates user review text as a heterogenous prior - this helps in regularizing the variational posterior with a user-dependent prior instead of a Gaussian prior. Latest research focuses on multi-task learning by adding auxiliary tasks, and tries to incorporate both short-term and long-term dependencies using a Mixture of Experts setting [41, 42]. The M3 Model [34] proposes an MoE gating network on top of three sub-models - neural network (tiny-range), RNN / CNN (short-range), attention (long-range).

An alarming concern about deep neural recommender systems is that they are not reproducible [5], and that they don't outperform even simple non-neural baselines. We incorporate the suggestions proposed in [5] by explaining choice of hyperparameters, choice of evaluations and baselines.
**Graph-based techniques**: Here, next-item prediction is formulated as a link prediction task in a bipartite graph of users and items. Recent work has centered on learning robust node representations which change dynamically with time [37]. An appropriate distance-metric is used to find the closest item embedding to a given user embedding. Of particular interest to our work is the JODIE model [20], which predicts the future embedding trajectories in temporal interaction networks using two RNNs and an attention layer. We utilize this work to obtain embeddings in our model. Embedding models are fairly simple and efficient as compared to other methods. To ensure they are robust and disentangled, recent papers have tried to assign multiple embeddings to the same node [9, 23] - wherein each embedding for represents different contexts in which a node could exist.

*Knowledge graphs* offer rich relationships over node edges which help in ensuring explainability, increasing diversity, and tackling cold-start [40]. [39] builds an end-to-end system by applying Graph Neural Networks (GNN) on knowledge graphs. Representaions learned over KGs are then used to predict items to users following a metric learning approach. [31] uses GNNs on user's social networks to learn how social influence changes dynamically with each session.
**Multi-arm Bandits**: Bandit algorithms [2, 26–28], centered on explore-exploit trade off, are well suited to the dynamic environment that we are interested in due to their intrinsic online learning nature. [19] is one of the first papers to propose an efficient Thompson Sampling procedure for online Bayesian Matrix Factorization. [47] uses a time-varying reward mapping function, where the reward weight vectors are modeled using state space equations. Processing large feature vectors is a time-consuming process, so [1] uses bandits to learn a masking over feature vectors. In a similar vein of improving efficiency, many recent works on bandits focus on clustering of users and items. [7] proposes a multi-task learning setting to pool information of items together while doing reward regression. Other works [11, 21, 53] cluster users and items into classes and make the reward vector same for all members in a class - for instance, CoFiBa [21] co-clusters users and items dynamically, but the algorithm is quite involved and assumes that the content universe and known in advance. [11] improves upon CoFiBa by proposing a simple algorithm with context-dependent clustering, and avoiding the graph formulations of CoFiBa.

## 2 OUR WORK

We present an online version of JODIE [20], by incorporating the rich, dynamic embeddings generated by the algorithm in a contextual bandit setting. In particular, we use Thompson sampling for decisions at each timestep, which utilizes a posterior distribution over the parameters of each arm for more informed rewards and updates.

### 2.1 Contextual Bandits with Thompson Sampling

In the contextual bandit setting, we have a context vector $x_t$ for each timestep $t$, and based on an algorithm $\mathcal{A}$ we choose a particular action $a_t$. This action yields a reward $r_t$, which is then used to update the internal state of out algorithm. In this particular case, our algorithm shall comprise a Bayesian linear regressor with non-linear transformation of inputs via neural networks, followed by Thompson sampling to choose the action. At the end of the process, the reward for the algorithm is given $r = \sum_{i=1}^{n} r_t$, and the cumulative regret is defined as $R_{\mathcal{A}} = r^* - r$. Our objective is to minimise this cumulative regret.

#### 2.1.1 Bayesian Linear Regression with Neural Features.

In this, we assume that the reward for datapoint j upon choosing action i $y_{ij}$ may be generated by $y_{ij} = \beta_i^T x_j + \epsilon_{ij}$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Thus, the parameters for an arm $i$ may be considered to be drawn from a distribution $\mathcal{N}(\mu_i, \Sigma_i)$. We now wish to obtain the joint posterior over $\beta$ and $\sigma^2$ over all arms, as it will form the basis for our Thompson sampling. For convenience, we shall drop the subscripts.

The posterior at time $t$ for action $i$, having seen inputs $X$ and the corresponding rewards $Y$ is $\pi_t(\beta, \sigma^2) = \pi_t(\beta|\sigma^2)\pi_t(\sigma^2)$, where $\sigma^2 \sim \mathrm{IG}(a_t, b_t)$ and $\beta|\sigma^2 \sim \mathcal{N}(\mu_t, \sigma^2\Sigma_t)$ (Inverse Gamma and Gaussian distributed, respectively). Given priors $IG(a_0, b_0)$ and $\mathcal{N}(\mu_0, \Sigma_0)$, the closed form updates for these are well known, and are presented below.

$$\Sigma_t = (X^T X + \Lambda_0)^{-1} \qquad \mu_t = \Sigma_t(\Lambda_0\mu_0 + X^T Y)$$
$$a_t = a_0 + t/2 \qquad b_t = b_0 + \frac{1}{2}\left(Y^T Y + \mu_0^T \Sigma_0 \mu_0 - \mu_t^T \Sigma_t^{-1} \mu_t\right)$$

The prior hyperparameters are $\mu_0 = 0, \Lambda = \lambda \mathbb{I}_d, a_0 = b_0 = \eta > 1$. It follows that for $\sigma_0^2 \sim \mathrm{IG}(\eta, \eta)$, $\Sigma_0 = \sigma_0^2/\lambda\mathbb{I}_d$. This is done for each arm independently. Note that this can easily be extended into the online case by considering the posterior of the previous time-step as the prior of the next time-step.

While this model provides accurate uncertainty estimates, it can only learn a limited class of linear models. One way to tackle this is to introduce complex non-linear features via a transformation of the input. We do this in two stages. In the first, we use JODIE (further explained in next section) to generate an embedding $u_t = f_{JODIE}(x_t)$. This serves as our new context for the bandit. We then pass $u_t$ through a simple feed-forward network to obtain a representation $z_t = g(u_t)$. Not only does this provide access to non-linear features, but it allows for the representation of abstract semantic concepts, like the dependence of the user's choice on the previous choices they made, via the first stage. While the posterior is updated after reading every new data-point, the embedding network is updated only after a certain number of points have been seen.

#### 2.1.2 Thompson Sampling.

Once we obtain a $z_t$ from our $x_t$, we sample $\beta_{i,t}$'s from the posterior distribution obtained above. We then calculate estimated rewards for each action using $\beta_{i,t}^T z_t$. We now greedily choose the action $a_i$ that maximises the estimated reward. The action is played and the corresponding reward is obtained. We consider the reward for all the remaining actions to be zero. This gives us a reward vector $y_t$, which is then used to update the posterior distributions. Thus, Thompson sampling gives us a nice Bayesian framework to calculate rewards, and incorporates the benefits of the same, which is why we chose it for our experiments.

### 2.2 Semantic Embeddings using JODIE

JODIE aims to learn semantically useful embeddings of users and items for better predictions. In particular, it aims to learn the *trajectory* of the embeddings - that is, how they evolve over time. This can help model how user preferences change over time, and provides a much more powerful framework for modelling various dependencies for temporal recommendation. Embeddings from JODIE comprise of both a static part (represented by a one-hot encoding over the number of users) and a dynamic part. In the following discussion, we shall refer to the dynamic portion of the embedding.

In particular, upon receiving a user-item pair, JODIE picks out the corresponding user embedding so far, and the item embedding of the last item chosen by the user. It then applies a projection step, which uses the time elapsed since the last time this user was seen to predict the current state of the user embedding (thus trying to model how the user's preferences may have changed over the time they did not interact with the system). More formally, if the time difference is $\delta$, a time-context vector $w = W_p\delta$ is formed using a linear layer, and the projected embedding is taken as $u(t+\delta) = (1+w) * u(t)$, where $*$ denotes the element-wise product. The 1 is to ensure that if $\delta = 0$, $u(t+\delta) = u(t)$. This projected user embedding is concatenated with the previous item embedding. In our work, this is the embedding we take for the context for our bandit.

Now, the user-item embedding is used to generate an item embedding, which is supposed to represent the item in the user-item pair received. This should ideally align well with the actual item embedding so far. Next, the user and item embeddings are updated in a coupled fashion via recurrent neural networks. This helps encode both the dependence between the user and item, and the history of choices made. The network is trained as mentioned in the original paper. If we were to use the concept of t-batching (basically dividing the data into mini-batches where no user or item is repeated), even the updates would occur at the same times. This would allow us to separate the training of JODIE and the contextual bandit, by storing the user-item embedding sequence mentioned above. Alternatively, we may also train the network by analysing data points one-by-one, but then we must forgo the speed improvements available by t-batching.

## 3 EXPERIMENTS

The MOOC student dropout dataset [24] consists of actions, for example, viewing a video, submitting an answer, etc., done by students on a MOOC online course. We take the first 20,000 interactions consisting of 21 items (videos, answers, etc.) and 1435 users (students). The task is to predict the item a user picks at each interaction, that is, an online learning fashion. To check the effectiveness of dynamic embeddings, we compare its performance against the case of using only feature vectors as context; and to check the effectiveness of bandit setting, we compare its performance against a Support Vector Machine (SVM) baseline trained with Stochastic Gradient Descent (SGD) for one epoch (to mimic the online setting). We use a simple evaluation metric of Accuracy in our experiments. Simply subtracting this from 1 and multiplying by the number of data-points gives us the cumulative regret in the bandit case. Other metrics such as recall@10, Mean Reciprocal Rank (MRR) can also be used. Results are shown in Table 1

| Context vector | SVM with SGD | Neural Bandit |
|---|---|---|
| Feature vector | 0.214 | 0.230 |
| User embedding | **0.276** | **0.345** |

**Table 1: Experimental Results on MOOC Dataset (Accuracy)**

For the Bayesian linear regressors, we take $a_0 = b_0 = 6, \lambda_0 = 0.25$ as the prior parameters. The feed-forward network has a hidden layer of 50 units, is updated after every 50 interactions, and is trained for 4 epochs each time (ideally this should be much higher, but we had to reduce the computational time required). We implement a round-robin scheme where every action is chosen twice before moving to Thompson sampling. This initial forced exploration helps counteract the "cold-start" effect.

In the MOOC dataset, every user-item interaction also has a feature vector associated with it, which has information like duration of interaction and rating given by user. We this as the context for our baseline. We see that using JODIE embeddings gives us about 50% improvement over this baseline in both cases, which highlights that using these embeddings is worthwhile. The contextual bandit also does better than the SVM classifier in both cases, showing the effectiveness of the contextual bandit model in an online setting. These improvements are especially remarkable considering the low data and high model complexity of the neural bandit setting, and could probably be improved upon significantly.

While we do much better than random, we still need to extensively test the model with more data, more training epochs and different embedding styles. Note that the neural bandit model has high complexity, and hence it must be trained much more rigorously to give satisfactory results.

## 4 FUTURE WORK

There are multiple angles we could further extend this work. Some of these are:

- This scheme cannot handle datasets with thousands, let alone millions, of possible items to be recommended. For this, we could look at ways to reduce the number of parameters in the bandit model. This could be done by clustering the possible arms (i.e. grouping similar items together), or representing each arm as a linear combination of certain "base" arms. This would have the added benefit of information sharing between arms, which would lead to more holistic rewards and parameter updates.
- We could look at other methods to generate embeddings for the contextual bandit, to address the shortcomings of JODIE. For example, we could look to extend [34] to the online setting, where different types of networks are used to model different dependency ranges. On the other hand, one reason for poor performance could be the high complexity of the embedding model. Considering this, simpler models could also be tried out.
- We were limited by the amount of computational power available to us. It would be instructive to whether improvements can be obtained by increasing the amount of data, number of training epochs, etc., and using GPUs.
- Thompson sampling can be refined further by considering the posterior over all the parameters of the model, including the neural networks. For this, we would need to look at how to adapt these networks into their Bayesian versions, by assuming the weights are also derived from a probability distribution.

## REFERENCES


[1] Djallel Bouneffouf, Irina Rish, Guillermo A Cecchi, and Raphaël Féraud. 2017. Context attentive bandits: Contextual bandit with restricted context. *arXiv preprint arXiv:1705.03821* (2017).

[2] L Elisa Celis, Sayash Kapoor, Farnood Salehi, and Nisheeth Vishnoi. 2019. Controlling polarization in personalization: An algorithmic framework. In *Proceedings of the Conference on Fairness, Accountability, and Transparency.* ACM, 160–169.

[3] Laurent Charlin, Rajesh Ranganath, James McInerney, and David M Blei. 2015. Dynamic poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems.* ACM, 155–162.

[4] Sotirios Chatzis. 2014. Dynamic bayesian probabilistic matrix factorization. In *Twenty-Eighth AAAI Conference on Artificial Intelligence.*

[5] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. *arXiv preprint arXiv:1907.06902* (2019).

[6] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Deep coevolutionary network: Embedding user and item features for recommendation. *arXiv preprint arXiv:1609.03675* (2016).

[7] Aniket Anand Deshmukh, Urun Dogan, and Clay Scott. 2017. Multi-task learning for contextual bandits. In *Advances in neural information processing systems.* 4848–4856.

[8] Trong Dinh Thac Do and Longbing Cao. 2018. Gamma-poisson dynamic matrix factorization embedded with metadata influence. In *Advances in Neural Information Processing Systems.* 5824–5835.

[9] Alessandro Epasto and Bryan Perozzi. 2019. Is a Single Embedding Enough? Learning Node Representations that Capture Multiple Social Contexts. In *The World Wide Web Conference.* ACM, 394–404.

[10] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. 2015. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems.* 1954–1962.

[11] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. 2017. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70.* JMLR. org, 1253–1262.

[12] Chengyue Gong et al. 2017. Deep dynamic Poisson factorization model. In *Advances in Neural Information Processing Systems.* 1666–1674.

[13] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems.* ACM, 161–169.

[14] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. arXiv:cs.LG/1511.06939

[15] 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.* ACM, 241–248.

[16] Seyedabbas Hosseini, Ali Khodadadi, Keivan Alizadeh, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid RR Rabiee. 2018. Recurrent poisson factorization for temporal recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2018).

[17] How Jing and Alexander J Smola. 2017. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* ACM, 515–524.

[18] Giannis Karamanolakis, Kevin Raji Cherian, Ananth Ravi Narayan, Jie Yuan, Da Tang, and Tony Jebara. 2018. Item recommendation with variational autoencoders and heterogeneous priors. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems.* ACM, 10–14.

[19] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. 2015. Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In *Advances in neural information processing systems.* 1297–1305.

[20] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* ACM, 1269–1278.

[21] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on*

*Research and Development in Information Retrieval*. ACM, 539–548.
[22] Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. 2018. Learning temporal point processes via reinforcement learning. In *Advances in Neural Information Processing Systems*. 10781–10791.
[23] Ninghao Liu, Qiaoyu Tan, Yuening Li, Hongxia Yang, Jingren Zhou, and Xia Hu. 2019. Is a Single Vector Enough? Exploring Node Polysemy for Network Embedding. *arXiv preprint arXiv:1905.10668* (2019).
[24] Tharindu Rekha Liyanagunawardena, Andrew Alexandar Adams, and Shirley Ann Williams. 2013. MOOCs: A systematic study of the published literature 2008-2012. *The International Review of Research in Open and Distributed Learning* 14, 3 (2013), 202–227.
[25] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical Gating Networks for Sequential Recommendation. *arXiv preprint arXiv:1906.09217* (2019).
[26] Jérémie Mary, Romaric Gaudel, and Philippe Preux. 2015. Bandits and recommender systems. In *International Workshop on Machine Learning, Optimization and Big Data*. Springer, 325–336.
[27] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 31–39.
[28] Carlos Riquelme, George Tucker, and Jasper Snoek. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127* (2018).
[29] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 297–305.
[30] Elena Smirnova and Flavian Vasile. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*. ACM, 2–9.
[31] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 555–563.
[32] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. *arXiv preprint arXiv:1904.06690* (2019).
[33] 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.
[34] Jiaxi Tang, Francois Belletti, Sagar Jain, Minmin Chen, Alex Beutel, Can Xu, and Ed H Chi. 2019. Towards neural mixture recommender for long range dependent user sequences. In *The World Wide Web Conference*. ACM, 1782–1793.
[35] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. 565–573. http://doi.acm.org/10.1145/3159652.3159656
[36] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 729–739.
[37] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2018. Representation learning over dynamic graphs. *arXiv preprint arXiv:1803.04051* (2018).
[38] Utkarsh Upadhyay, Abir De, and Manuel Gomez Rodriguez. 2018. Deep reinforcement learning of marked temporal point processes. In *Advances in Neural Information Processing Systems*. 3168–3178.
[39] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. ACM, 968–977. https://doi.org/10.1145/3292500.3330836
[40] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *The World Wide Web Conference*. ACM, 3307–3313.
[41] Shoujin Wang, Longbing Cao, and Yan Wang. 2019. A survey on session-based recommender systems. *arXiv preprint arXiv:1902.04864* (2019).
[42] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 6332–6338. https://doi.org/10.24963/ijcai.2019/883
[43] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, and Alexander J Smola. 2016. Joint training of ratings and reviews with recurrent recommender networks. (2016).
[44] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. ACM, 495–503.
[45] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenburg, and Jure Leskovec. 2019. Hierarchical Temporal Convolutional Networks for Dynamic Recommender Systems. In *The World Wide Web Conference*. ACM, 2236–2246.
[46] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 582–590.
[47] Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. 2016. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2025–2034.
[48] Jia-Dong Zhang and Chi-Yin Chow. 2018. SEMA: Deeply learning semantic meanings and temporal dynamics for recommendations. *IEEE Access* 6 (2018), 54106–54116.
[49] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun. 2018. Dynamic Intention-Aware Recommendation with Self-Attention. *arXiv preprint arXiv:1808.06414* (2018).
[50] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 5.
[51] Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192* (2018).
[52] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2018. ATRank: An attention-based user behavior modeling framework for recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
[53] Li Zhou and Emma Brunskill. 2016. Latent contextual bandits and their application to personalized recommendations for new users. *arXiv preprint arXiv:1604.06743* (2016).