From Preference into Decision Making: Modeling User Interactions in Recommender Systems

Qian Zhao*
Bloomberg L.P.
New York, United States
qzhao101@bloomberg.net

Martijn C. Willemsen
Eindhoven University of Technology
Eindhoven, The Netherlands
Jheronimus Academy of Data Science
's-Hertogenbosch, The Netherlands
M.C.Willemsen@tue.nl

Gediminas Adomavicius
F. Maxwell Harper
Joseph A. Konstan
University of Minnesota
Minneapolis, United States
gedas,max,konstan@umn.edu

ABSTRACT

User-system interaction in recommender systems involves three aspects: temporal browsing (viewing recommendation lists and/or searching/filtering), action (performing actions on recommended items, *e.g.*, clicking, consuming) and inaction (neglecting or skipping recommended items). Modern recommenders build machine learning models from recordings of such user interaction with the system, and in doing so they commonly make certain assumptions (*e.g.*, pairwise preference orders, independent or competitive probabilistic choices, etc.). In this paper, we set out to study the effects of these assumptions along three dimensions in eight different single models and three associated hybrid models on a user browsing data set collected from a real-world recommender system application. We further design a novel model based on recurrent neural networks and multi-task learning, inspired by Decision Field Theory, a model of human decision making. We report on precision, recall, and MAP, finding that this new model outperforms the others.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

decision making, recurrent neural networks, decision field theory

ACM Reference Format:
Qian Zhao, Martijn C. Willemsen, Gediminas Adomavicius, F. Maxwell Harper, and Joseph A. Konstan. 2019. From Preference into Decision Making: Modeling User Interactions in Recommender Systems. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3298689.3347065

1 INTRODUCTION

The process of user-system interaction in standard recommender systems involves three aspects: *user browsing* (item displays), *action* and *inaction*. For example, in Youtube or Netflix etc., users typically browse items page by page and make decisions of whether to act upon some of the items or not, *e.g.*, click to see details or directly consume.

When recognizing what's displayed, we can model user decisions as temporal processes of information gathering, paying attention, reasoning and decision making. The temporal aspect here is different from the classical user preference temporal dynamics modeled in prior work [9] where user interest is shifting across time between choices. We instead focus on the accumulative user attention on items temporally and the temporal dependency of a user's current behavior on the user's past interactions in the system, which may or may not involve user interest shift. The user-item interaction is also contextual because of the competition effects among items displayed together in a page. This user choice process has been modeled in prior work, *e.g.*, the Collaborative Competitive Filtering in [17].

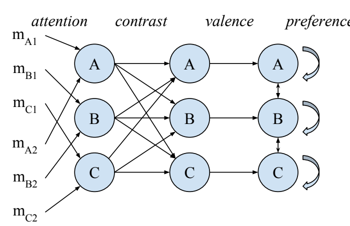

Figure 1: The connectionist network representation of DFT.

Formal accounts of behavioral decision making processes can be found in psychology literature, among which we find that the Decision Field Theory (DFT) [2] is particularly relevant here. This theory postulates a temporal deliberation process in the context where people are faced with multiple choices. Fig. 1 illustrates what happens in each unit of time (*i.e.*, the deliberation process is the temporal replication of Fig. 1). The first layer of the network models people's *attention* on the important aspects of the choices, which forms *contrast* in the second layer that models the competition among the choices and produces *valence* in the third layer, which recurrently accumulates to produce *preference*. The double-direction connections among the choices in the final layer models the effect of lateral inhibition (*i.e.*, a choice that happens to win out first might dominate later). DFT addresses both the choice context and the temporal dependency of choices. This directly matches the scenario in screen-based recommender systems where users browse items page by page and make decisions of action or not (note that in recently studied voice-driven recommender systems [7, 16], this process however may not apply).

*Work performed at the University of Minnesota.

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347065

As suggested by DFT, preference is actually the end result of micro-level decision making processes. However, many if not most prior modeling research in recommender systems have been on preference estimation only without going into modeling the underlying processes. For example, typical recommendation models assume certain parameters to represent a user which could be a single latent vector or a whole interaction history of the user represented by latent vectors and then fit the user feedback data observed in the system, *e.g.*, through the models of regression for explicit rating feedback as by [8], or point-wise binary logistic model after negative sampling for implicit action feedback as by [6], or learning-to-rank, particularly pairwise ranking as by [12], in which the relative preference order is fitted.

Besides the gap between the actual account of user decision making and simplified preference estimation models, we also notice that there is not much prior work systematically examining how modeling assumptions implicitly made by recommendation models affect recommendation accuracy or quality. In this paper, we set out to study the effects of these different modeling assumptions on a user browsing, action and inaction data set from a real-world recommender system application.

To further close the gap, we turn from modeling preference into modeling the decision making process of users inspired by the DFT above based on the techniques of RNN [14] and multi-task learning (MTL) [13]. The benefits of this human theory driven approach could be better recommendation accuracy, which we demonstrate in this work, and more importantly potential better user experience, which we discuss in the end.

2 RELATED WORK: *USER & PREDICTION* MODELING

In this section, we set up notations and synthesize a conceptual framework of *User & Prediction Modeling* from prior work. Similar disentanglement of recommendation modeling was done by Yang *et al.* for designing an extensible recommendation software tool: OpenRec [15]. A prominently different characteristic of recommendation modeling compared with general supervised machine learning is the step of learning user representations or user state (profile) modeling. Many recommendation techniques model the current user state and items with a low-dimensional vector space and predict the user's preferences on items by matching the two [9, 10].

Assume we have a user state model that combines user profile and context factors and lies in a low-dimensional space R_d. Denote $s = s(u) \in R_d$ as the current user's state vector in which u represents the user ID, profile and history in the system. We also sometimes use notation u to refer to a specific user ID depending on the context.

2.1 User Models: SVD and SVD++

Latent factor models have been designed to represent users and make predictions in recommender systems, *e.g.*, SVD by [10] and SVD++ by [9]. In SVD, the user representation is simply a d dimensional latent vector, denoted as U_u where $U \in R_{N \times d}$ and N is the number of users in the system, *i.e.*, $s(u) = U_u$. Note that we omit the user (and item) bias (or scalar) representations which model the global biases of users (and items) of the feedback distribution for the simplicity of presentation. In SVD++, the user representation also incorporates the whole set of interacted items by the user as shown in Equation 1 where $|I|$ is the set of acted-upon items and $Q \in R_{M \times d}$ represents the event of action by a user on an item.

$$s(u) = U_u + \frac{1}{\sqrt[2]{|I|}} \sum_{i \in I} Q_i \tag{1}$$

2.2 User Model: Recurrent Neutral Networks

Neural network based models can also be used to model user state in recommender systems. We examine a Recurrent-Neural Network (RNN) based model that has recently been demonstrated more effective than prior techniques and models a user as the user's temporal action sequence in the system [4, 14]. For a user sequence, each step is a vector representation of the item that the user acted upon (*e.g.*, liked, consumed), concatenating the embeddings of not only the item ID but also its side information, overall denoted as $CAT(\theta_{u_t})$ where θ denotes the embedding or transformation parameters of u_t. In the simplest case, each step can be Q_{a_t} where a_t is an item ID, similar to SVD++. One widely used step transition model for the sequence model is LSTM by [5]. Additional layers of transformation can be applied before modeling the transition with LSTM, *e.g.*, through a layer of Rectified Linear Units (ReLU) by [11]. In summary, a RNN model has the following user state model s in Equation 2 where t represents the time step.

$$s_t(u) = LSTM(s_{t-1}(u), ReLU(Q_{a_t})) \tag{2}$$

2.3 Prediction Models

Denote $o(v)$ as the item representation vector where v represents the item ID and the properties of an item including its side information. In the simplest case, $o(v) = W_a$ where $W \in R_{M \times d}$ represents the embeddings of item IDs. With the user state s and item representation o, making predictions on items involves modeling the match between the two $f(s, o)$. A typical choice is the inner product of s and o, *i.e.*, $f(s, o) = s^T o$. This prediction function is used to approximate or learn from user feedback r, which could be user actions (*e.g.* purchase, consumption etc.) on items, user browsing or item displays, or sampled items as pseudo negative observations.

Three possible ways of fitting the observations r can be found in prior literature: the logistic model, *e.g.*, by [6] (treating the observations as following independent Bernoulli distributions), the pairwise ranking model, *e.g.*, by [12] (modeling the relative preference order) and the softmax model, *e.g.*, by [17] (treating the feedback as an observation of an exclusive categorical distribution).

The *logistic* model involves a sigmoid transformation $g(f) = 1/(1 + exp(-f))$. It models independently for each user-item pair with a label $r \in 1, 0$ as shown in Equation 3.

$$\widehat{p(r|u, v)} = g^r (1 - g)^{1-r} \tag{3}$$

The *pairwise ranking* model collectively models a pair of items that a user has a relative preference order, *e.g.*, a user prefers an item a over an item b is modeled as Equation 4 which also involves transformation g.

$$p(r_a > \widehat{r_b | u, v_a}, v_b) = g(f(s, o(v_a) - o(v_b))) \tag{4}$$

The *softmax* model collectively models a positive item and a set of negative items (which could be sampled) as shown in Equation 5, where i are k are indexing in the $\alpha + 1$ observations, which include one positive and other negative ones, *i.e.*, α is the negative sample size and a here is an one-hot encoded representation of the positive and negative item IDs.

$$\widehat{p(a|u,v)} = \widehat{p_a}(a_i = 1|u,v) = \frac{exp(f(s,o_i))}{\sum_{k=1}^{\alpha+1} exp(f(s,o_k))} \quad (5)$$

3 THE EXAMINED MODELS

Table 1 (Models S1-8) lists eight models we examine in this work following the *User&Prediction* modeling framework. Note that it is not a factorial design but instead designed to gradually increase the model complexity.

To further explain some of them,

- *CTR*: This is fitting the action probability (*e.g.*, Click-Through Rate) using standard matrix factorization model SVD.
- *CCF*: This is the Collaborative Competitive Filtering model [17]. It has two versions D-CCF or S-CCF depending on where the negative items come from. Note that for D-CCF, we build a softmax for each page view of items because this best fits the Decision Field Theory and the recommender system applications where users typically browse items page by page and make decisions of whether to act upon the displayed items. For page views without any interactions, D-CCF as proposed by [17] introduces a user-specific parameter θ_u to model the threshold each user prefers to reach for action as shown in Equation 6. This situation does not apply for S-CCF which uses a normal softmax loss as in Equation 5 where negative items are randomly sampled.
- *BPR*: This is the pairwise ranking model where the assumption is that user prefers positive items over negative items which could be displayed but without action (D-BPR) or sampled through negative sampling (S-BPR).

$$\widehat{p(a|u,v)} = \widehat{p_a}(a_i = 1|u,v) = \frac{exp(f(s,o_i))}{\theta_u + \sum_{k=1}^{\alpha+1} exp(f(s,o_k))} \quad (6)$$

3.1 *Retrieving & Ranking Based Hybrid*

We further examine a two-stage recommendation modeling framework: *retrieving* and *ranking* to explore the possibilities of hybrid models. In this framework, the retrieving model initially retrieves a set of candidate items and the ranking model further ranks the candidates to make final recommendation. This modeling framework can be used to combine models for user browsing and action feedback data, which might be effective because what's displayed or recommended is usually higher-quality items. Models built on item displays and their actions might perform well on higher-quality items but might not generalize well on all items and vice versa. In this work, we tested three ranking models built on displayed positive action and negative inaction items combined with one retrieving model built on positive action and sampled negative items as listed in Table 1 (Models H1-3). Note that the first model (before & sign) is for retrieving and the second model (after & sign) is for ranking.

3.2 Page Level RNN (PL-RNN)

Applying the Decision Field Theory (DFT) into recommender systems takes simplification and adaptation. First, the attention weights and the connection weights in Fig. 1 are unknown and to be estimated from data. Second, the exact match of the deliberation in DFT is one page view in recommender systems which however lacks observational data to estimate the network weights unless there is eye-tracking data. Therefore, we turn to model the whole user page view sequence as the deliberation process so that Fig. 1 corresponds to one page view in the user page view sequence. Further, we represent an item as low-dimensional embedding vector, each dimension of which corresponds to a latent property that users pay attention to. For the contrast layer, we use a ReLU layer and for the recurrent valence layer, we use a RNN layer. This model can be directly matched with Eq. 2 except that the input comes not from one acted-upon item but from a page of displayed items with or without actions. Their embeddings are concatenated as the input of the network. The model is fully illustrated by Fig. 2. We name this model as *Page-Level RNN (PL-RNN)*. For the last output layer, we fit not only the observed acted-upon items but also all the displayed items in the next page view to estimate the network weights, *i.e.*, the technique of multi-task learning (MTL) [13].

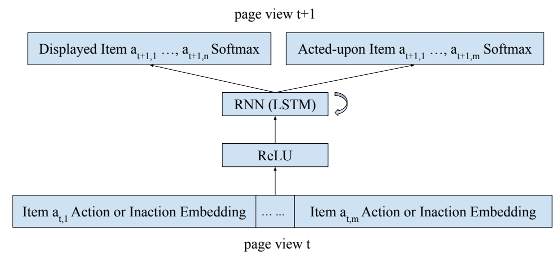

Figure 2: The architecture of the Page-Level RNN model. Each page view has n items and m of them are acted upon by the user. The input embedding differentiates acted-upon items or inaction items, *i.e.*, the vocabulary of the embeddings is $2\times$ the item vocabulary.

4 EXPERIMENTS AND RESULTS

To evaluate the list of models in Table 1, we conducted experiments on a data set of user browsing (item displays), (positive) action and inaction items collected from a real-world movie recommender system application MovieLens (https://movielens.org), which has around 60K movies in its database. The data set has 45M movie displays and 1.16M *(positive) actions* (which specifically refers to high ratings, *i.e.*, >= 4 out of 5 stars, clicking to see details or adding into a wishlist) from 22K real users between Jan. 12, 2017 and Jan. 14, 2018. The item displays came from two types of page views: the *front* page which by default displays 48 items and the *explore* page which by default displays 24 items (in a 3×8 grid). We use $m = 24$ for Fig. 2 and split the *front* page view into two page views, which we think is a reasonable assumption because users naturally browse from the top to the bottom of the page. This can be similarly done for other interfaces in different systems.

We employed a temporal training and testing procedure, treating the data from Jan. 12, 2017 until Oct. 31, 2017 as the *training* data and the rest as the *testing* data. The metrics we used are Precision@8-24, Recall@8-24, MAP@8-24 (Mean Average Precision) evaluating the top-N recommendation accuracy, *i.e.*, how

Table 1: Modeling factors and evaluation results of all the examined models in this work and the new PL-RNN model. S1-8 are single models. H1-3 are hybrid models (see Section 3.1).

User Model	Prediction Model	Negative Items	Model	Precision				Recall				MAP			
				@8	*@15*	*@20*	*@24*	*@8*	*@15*	*@20*	*@24*	***@8***	*@15*	*@20*	*@24*
SVD	logistic	displayed but inaction	**S1. CTR**	0.0017	0.0042	0.0041	0.0038	0.0004	0.0021	0.0027	0.0030	**0.0006**	0.0007	0.0006	0.0005
	softmax		**S2. D-CCF**	0.0014	0.0035	0.0044	0.0042	0.0003	0.0017	0.0029	0.0033	**0.0006**	0.0006	0.0006	0.0005
	pairwise		**S3. D-BPR**	0.139	0.121	0.112	0.107	0.0371	0.0605	0.0744	0.0855	**0.114**	0.098	0.092	0.090
		sampled	**S4. S-BPR**	0.140	0.129	0.116	0.109	0.0372	0.0646	0.0774	0.0873	**0.106**	0.096	0.089	0.086
	logistic		**S5. SVD**	0.147	0.127	0.118	0.113	0.0391	0.0635	0.0789	0.0900	**0.104**	0.087	0.081	0.078
	softmax		**S6. S-CCF**	0.132	0.123	0.110	0.103	0.0351	0.0615	0.0733	0.0821	**0.097**	0.088	0.082	0.080
SVD++			**S7. SVD++**	0.137	0.134	0.120	0.111	0.0365	0.0671	0.0802	0.0888	**0.106**	0.099	0.092	0.088
RNN			**S8. RNN**	0.157	0.151	0.133	0.130	0.0832	0.0943	0.1201	0.1433	**0.119**	0.115	0.104	0.101
RNN&SVD	softmax&logistic	both	**H1. RNN&CTR**	0.089	0.099	0.122	0.123	0.0327	0.0495	0.0961	0.1396	**0.040**	0.043	0.053	0.056
	softmax&softmax		**H2. RNN&D-CCF**	0.094	0.102	0.124	0.126	0.0324	0.0489	0.1049	0.1422	**0.037**	0.040	0.051	0.056
	softmax&pairwise		**H3. RNN&D-BPR**	0.168	0.154	0.141	0.130	0.0879	0.0990	0.1250	0.1449	**0.140**	0.130	0.118	0.111
PL-RNN	softmax	both	**PL-RNN**	0.170	0.147	0.147	0.140	0.0870	0.1058	0.1295	0.1556	**0.141**	0.136	0.122	0.118

accurate and complete the recommendation model can be in predicting the future positively acted-upon items. Note that top-N=8 to 24 here corresponds to the grid size in the interface described above. The embedding dimension in all models is $d = 32$. The number of sampled negative items is $\alpha = 2000$. We used AdaGrad by [3] to train all the models until convergence. The ranking model further re-ranks 24 retrieved candidate items by the retrieving model. All models are implemented in TensorFlow [1] and open sourced (https://github.com/grouplens/samantha).

Table 1 shows the results. Since different metrics are generally consistent with each other in terms of the trend, for simplicity of presentation, we focus on comparing *MAP@8* (shown in bold in the table). First, we see that PL-RNN model performs the best. CTR and D-CCF do not seem to learn much about user preferences because their metric is orders of magnitude lower compared with others. D-BPR and S-BPR achieve similar level of accuracy while SVD and S-CCF perform comparatively. D-BPR or S-BPR generally performs better than SVD and S-CCF. SVD++ performs slightly better than SVD, but RNN demonstrates a much bigger improvement compared with SVD++. Further ranking using D-BPR on the retrieved candidates from RNN leads to a substantial performance boost. PL-RNN as one single model has similar level of performance improvement compared with RNN and further performs consistently better than the hybrid case RNN&D-BPR.

5 DISCUSSION AND CONCLUSION

To discuss the results, we highlight the key modeling factors studied in this work here, along which our results could be generalized:

- User State (or User Model): which could be *static* (*e.g.*, SVD), a *set* of interacted items (*e.g.*, SVD++), or *sequential* (*e.g.*, RNN)
- Preference Assumption (or Prediction Model): which could be *independent binary choices* (*e.g.*, logistic loss), *categorical competitive choices* (*e.g.*, softmax loss), or *relative preference* (*e.g.*, pairwise ranking)
- Item Feedback Space: which could be *(positive) action* items with *negative samples*, *displayed action* and *inaction items*, or the hybrid of the two (through *e.g.*, Retrieving&Ranking or PL-RNN)

We can make the following observations regarding each of these factors and how they interact with each other.

First, we observe that softmax and logistic preference (prediction or loss) models are very sensitive to the choice of negative items, *i.e.*, these models only learn well on negative items obtained through negative sampling, not on items that are displayed but do not have user action (comparing the model *SVD* vs. *CTR* and *S-CCF* vs. *D-CCF*). On the other hand, pairwise preference models can learn equally well on both types of negative items, which demonstrates the robustness of this method as a way of inferring user preferences (comparing the model *D-BPR* vs. *S-BPR*). What's more, pairwise preference models seem to have better recommendation accuracy than softmax or logistic models in terms of MAP scores (comparing the model *S-BPR* vs. *SVD* or *S-CCF*).

Modeling user actions as a sequence through RNN seems to learn a substantially better user representation since it achieves better recommendation accuracy compared with the user models of SVD or SVD++ (comparing the model *RNN* vs. *SVD* or *SVD++*). Further, we can gain substantially better recommendation accuracy by combining the RNN model built on positive and sampled negative items and the pairwise ranking model built on displayed and action or inaction items following the modeling framework of Retrieving & Ranking (comparing *RNN&D-BPR* vs. *RNN*).

We propose to model the user temporal interactions with recommender systems in the page view level to capture the temporal dependencies among the sequence of user page views and the competition effects of context items displayed together in the page view. The model is better aligned with the theory of human decision making from psychology literature, going beyond point-wise or pairwise user preferences between positive and negative items. We demonstrate that this page-level model has the best recommendation accuracy among all the models we compared. Especially, it gains a substantial accuracy improvement compared with the RNN that only models the sequence of positive items ignoring the observations of user browsing or item displays.

It is interesting future work to study how this page-level model affects user experience through a field experiment testing specific hypotheses on user perception. Since the state representation of this model encodes not only what's acted upon, but also what's been displayed to users, we conjecture that it might be able to mitigate situations where users are bored by the same repeated recommendations, or confusing situations where users couldn't find previously recommended items.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grant IIS-1319382. The first author was also supported by the Doctoral Dissertation Fellowship, 2016-17, by the Graduate School at the University of Minnesota.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

[2] Jerome R Busemeyer and James T Townsend. 1993. Decision field theory: A dynamic-cognitive approach to decision making in an uncertain environment. *Psychological review* 100, 3 (1993), 432.

[3] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.

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

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

[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on.* Ieee, 263–272.

[7] Jie Kang, Kyle Condiff, Shuo Chang, Joseph A Konstan, Loren Terveen, and F Maxwell Harper. 2017. Understanding how people use natural language to ask for recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 229–237.

[8] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[9] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[11] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[12] 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.

[13] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[14] 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.

[15] Longqi Yang, Eugene Bagdasaryan, Joshua Gruenstein, Cheng-Kang Hsieh, and Deborah Estrin. 2018. Openrec: A modular framework for extensible and adaptable recommendation algorithms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 664–672.

[16] Longqi Yang, Michael Sobolev, Christina Tsangouri, and Deborah Estrin. 2018. Understanding user interactions with podcast recommendations delivered via voice. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 190–194.

[17] Shuang-Hong Yang, Bo Long, Alexander J Smola, Hongyuan Zha, and Zhaohui Zheng. 2011. Collaborative competitive filtering: learning recommender using context of user choice. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 295–304.