Personalized Neural Embeddings for Collaborative Filtering with Text

Guangneng Hu
Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong, China
njuhgn@gmail.com

Abstract

Collaborative filtering (CF) is a core technique for recommender systems. Traditional CF approaches exploit user-item relations (e.g., clicks, likes, and views) only and hence they suffer from the data sparsity issue. Items are usually associated with unstructured text such as article abstracts and product reviews. We develop a Personalized Neural Embedding (PNE) framework to exploit both interactions and words seamlessly. We learn such embeddings of users, items, and words jointly, and predict user preferences on items based on these learned representations. PNE estimates the probability that a user will like an item by two terms—behavior factors and semantic factors. On two real-world datasets, PNE shows better performance than four state-of-the-art baselines in terms of three metrics. We also show that PNE learns meaningful word embeddings by visualization.

1 Introduction

Recommender systems are widely used in e-commerce platforms, such as to help consumers buy products at Amazon, watch videos on Youtube, and read articles on Google News. They are useful to alleviate the information overload and improve user satisfaction. Given history records of consumers such as the product transactions and movie watching, collaborative filtering (CF) is among the most effective approaches based on the simple intuition that if users rated items similarly in the past then they are likely to rate items similarly in the future (Sarwar et al., 2001).

History records include both implicit (e.g., purchase and clicks) and explicit (e.g., likes/dislikes and ratings) feedback which can be represented as a user-item interaction matrix. Typically, observed user-item interactions are incomplete with a large portion remaining not recorded. The goal of recommendation is to predict user preferences on these missing interactions. This setting requires to complete the partial observed rating matrix. Matrix Factorization (MF) techniques which can learn latent factors for users and items are the main cornerstone for CF (Mnih and Salakhutdinov, 2008; Koren, 2008; Koren et al., 2009). It is effective and flexible to integrate with additional data sources (Hu et al., 2015). Recently, neural networks like Multilayer Perceptron (MLP) are used to learn an interaction function from data with the power of learning highly nonlinear relationships between users and items (Dziugaite and Roy, 2015; Cheng et al., 2016; He et al., 2017; Hu et al., 2018b). MF and neural CF exploit user-item behavior interactions only and hence they both suffer from the data sparsity and cold-start issues.

Items are usually associated with unstructured text, like news articles and product reviews. These additional sources are essential for recommendation beyond user-item interactions since they contain independent and diverse information. Hence, they provide an opportunity to alleviate the data sparsity issue (Ganu et al., 2009; Hu et al., 2018a). For application domains like recommending research papers and news articles, the unstructured text associated with the item is its text content (Wang and Blei, 2011; Wang et al., 2015; Bansal et al., 2016). For some domains like recommending products, the unstructured text associated with the item is its user reviews which justify the rating behavior (McAuley and Leskovec, 2013; He and McAuley, 2016). These methods adopt topic modelling techniques and neural networks to exploit the item content leading to performance improvement.

A typical way of exploiting text content is to firstly extract a feature vector for each document by averaging word embeddings in the document, and then to learn a text factor corresponding to this feature vector (Hu and Dai, 2017). These embeddings are pre-trained from a large corpus such as Wikipedia. This approach separates the extraction of text feature from the learning of user-item interaction. These two processes cannot benefit from each other and errors in the previous step maybe propagate to the successive steps. Another way is to learn a topic vector using topic modelling (Wang and Blei, 2011; McAuley and Leskovec, 2013; Bao et al., 2014) by aligning behavior factors and topic factors with a link function such as softmax and offset.

Recently, neural networks are used to learn a representation from the text using autoencoders (Wang et al., 2015; Zhang et al., 2016), recurrent networks (Bansal et al., 2016), and convolutional networks (Zheng et al., 2017; Catherine and Cohen, 2017). These methods treat different words in the document as equal importance and do not match word semantics with the specific user. Instead, we achieve to learn a personalized word embedding with the guidance of user-item interactions. That is, the importance of words is learned to match user preferences. The attention mechanism can be used to learn these importance weights. Memory Networks (MemNet) have been used in recommendation to model item content (Hu et al., 2018c; Huang et al., 2017), capture user neighborhood (Ebesu et al., 2018), and learn latent relationships (Tay et al., 2018). We follow this thread to adapt a MemNet to match word semantics with user preferences.

In this paper, we propose a novel neural framework to exploit relational interactions and text content seamlessly. The proposed Personalized Neural Embedding (PNE) model fuses semantic representations learnt from unstructured text with behavior representations learnt from user-item interactions jointly for effective estimation of user preferences on items. PNE estimates the preference probability by two kinds of factors. The *behavior factor* is to capture the personalized preference of a user to an item learned from behavior interactions. The *semantic factor* is to capture the high-level representation attentively extracted from the unstructured text by matching word semantics with user preferences.

To model the behavior factor, we adopt a neural CF approach, which learns the user-item nonlinear interaction relationships using a neural network (CFNet). To model the semantic factor, we adopt a memory network to match word semantics with the specific user via the attention mechanism inherent in the memory module (MemNet), determining which words are highly relevant to the user preferences. PNE integrates relational interactions with unstructured text by bridging neural CF and memory networks. PNE can also learn meaningful word embeddings.

2 Approach

We present PNE to jointly learn representations of users, items, and words. PNE seamlessly captures nonlinear user-item interaction relationships and matches word semantics with user preferences.

Denote the set of users by $\mathcal{U}$ and items by $\mathcal{I}$. We use a rating matrix $\boldsymbol{Y} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$ to describe user-item interactions where each entry $y_{ui} \in \{0, 1\}$ is 1 (observed entries) if user u has an interaction with item i and 0 (unobserved entries) otherwise. Usually the interaction matrix is very sparse since a user $u \in \mathcal{U}$ only consumed a very small subset of all items. For the task of item recommendation, each user is only interested in identifying $topK$ items (typically K is small e.g. tens or hundreds). Items are ranked by their predicted scores:

$$\hat{y}_{ui} = f(u, i|\Theta), \tag{1}$$

where f is an interaction function and Θ denotes model parameters.

2.1 Architecture

PNE consists of a CF network (CFNet) to learn a nonlinear interaction function and of a memory network (MemNet) to match word semantics with user preferences. The information flow in PNE goes from the input (u, i) to the output $\hat{y}_{ui}$ through the following five modules.

1. *Input:* $(u, i) \to (\vec{e}_u, \vec{e}_i)$ This module encodes user-item interaction indices. We adopt the one-hot encoding. It takes user u and item i, and maps them into one-hot encodings $\vec{e}_u \in \{0, 1\}^{|\mathcal{U}|}$ and $\vec{e}_i \in \{0, 1\}^{|\mathcal{I}|}$ where only the element corresponding to that user/item index is 1 and all others are 0.

2. *Embedding:* $(\vec{e}_u, \vec{e}_i) \to \boldsymbol{x}_{ui}$ This module firstly embeds one-hot encodings into continuous representations $\boldsymbol{x}_u = \boldsymbol{P}^T\vec{e}_u$ and $\boldsymbol{x}_i = \boldsymbol{Q}^T\vec{e}_i$ by embedding matrices $\boldsymbol{P} \in \mathbb{R}^{|\mathcal{U}|\times d}$ and $\boldsymbol{Q} \in \mathbb{R}^{|\mathcal{I}|\times d}$ respectively, where d is the latent dimension. It then concatenates them as $\boldsymbol{x}_{ui} = [\boldsymbol{x}_u, \boldsymbol{x}_i]$ to be the input of following CFNet and MemNet modules.

3. *CFNet:* $\boldsymbol{x}_{ui} \to \boldsymbol{z}_{ui}^{behavior}$ This module is a CF approach to exploit user-item interactions. It takes continuous representations from the embedding

Table 1: Datasets and statistics.

Dataset	#user	#item	#rating	#word	#density	avg. words
Amazon	8,514	28,262	56,050	1,845,387	0.023%	65.3
Cheetah	15,890	84,802	477,685	612,839	0.035%	7.2

module and then transforms to a final *behavior factor* representation:

$$\boldsymbol{z}_{ui}^{\text{behavior}} = \text{ReLU}(\boldsymbol{W}\boldsymbol{x}_{ui} + \boldsymbol{b}), \qquad (2)$$

where $\text{ReLU}(x) = \max(0, x)$ is an activation function, and $\boldsymbol{W}$ and $\boldsymbol{b}$ are connection weights and biases.

4. *MemNet:* $\boldsymbol{x}_{ui} \rightarrow \boldsymbol{z}_{ui}^{semantic}$ This module is to model the item content with the guidance of user-item interaction. The item content is modelled by memories. It takes representations from both the embedding module and the review text d_{ui} associated with the corresponding user-item (u, i) into a final *semantic factor* representation:

$$\boldsymbol{z}_{ui}^{\text{semantic}} = \sum\nolimits_{j:\, w_j \in d_{ui}} \text{Softmax}(a_j^{u,i})\boldsymbol{c}_j, \qquad (3)$$

where the external memory slot $\boldsymbol{c}_j$ is an embedding vector for word w_j by mapping it with an external memory matrix $\boldsymbol{C}$. The attentive weight $a_j^{u,i}$ encodes the relevance of user u to word w_j by content-based addressing:

$$a_j^{u,i} = \boldsymbol{x}_{ui}^T \boldsymbol{m}_j^{u,i}, \qquad (4)$$

where memory $\boldsymbol{m}_j^{u,i}$ is concatenated from internal memory slots $\{\boldsymbol{m}_j^u, \boldsymbol{m}_j^i\}$ which are mapped from word w_j by internal memory matrices $\boldsymbol{A}^u$ for user attention and $\boldsymbol{A}^i$ for item attention.

5. *Output:* $\boldsymbol{z}_{ui} \rightarrow \hat{y}_{ui}$ This module predicts the recommendation score $\hat{y}_{ui}$ for a given user-item pair based on the representation of both behavior factor and semantic factor from CFNet and MemNet respectively: $\boldsymbol{z}_{ui} = [\boldsymbol{z}_{ui}^{\text{behavior}}, \boldsymbol{z}_{ui}^{\text{semantic}}]$. The output is the probability that the input pair is a positive interaction. This is achieved by a logistic layer:

$$\hat{y}_{ui} = \frac{1}{1 + \exp(-\boldsymbol{h}^T \boldsymbol{z}_{ui})}, \qquad (5)$$

where $\boldsymbol{h}$ is model parameter.

2.2 Learning

We adopt the binary cross-entropy loss:

$$\mathcal{L}(\Theta) = -\sum_{(u,i)\in\mathcal{S}} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}), \qquad (6)$$

Table 2: Categorization of recommender approaches.

Baselines	Shallow method	Deep method
CF	BPR	MLP
CF w/ text	HFT, TBPR	LCMR, PNE (ours)

where $\mathcal{S} = \boldsymbol{Y}^+ \cup \boldsymbol{Y}^-$ is the union of observed interactions and randomly sampled negative examples. Model parameters $\Theta = \{\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{A}, \boldsymbol{h}\}$ where we use a single word embedding matrix $\boldsymbol{A}$ by sharing all memory matrices $\boldsymbol{A}^u$, $\boldsymbol{A}^i$, and $\boldsymbol{C}$ in order to reduce model complexity. The objective function can be optimized by stochastic gradient descent.

3 Experiment

In this section, we evaluate PNE on two datasets with five baselines in terms of three metrics.

3.1 Datasets

We evaluate on two real-world datasets. The public **Amazon** products (McAuley and Leskovec, 2013) and a company **Cheetah** Mobile news (Hu et al., 2018c; Liu et al., 2018) (see Table 1). We preprocess the data following the strategy in (Wang and Blei, 2011). The size of word vocabulary is 8,000.

3.2 Evaluation protocol

We adopt leave-one-out evaluation (Hu et al., 2018b) and use three ranking metrics: hit ratio (HR), normalized discounted cumulative gain (NDCG), and mean reciprocal rank (MRR).

We compare with five baselines (see Table 2).

- BPR (Rendle et al., 2009) is a latent factor model based on matrix factorization.
- HFT (McAuley and Leskovec, 2013) adopts topic distributions to learn latent factors from text reviews.
- TBPR (Hu and Dai, 2017) extends BPR by integrating text content via word embedding features. Word embeddings used in TBPR are pre-trained by GloVe (Pennington et al., 2014).

Table 3: Results ($\times 100$) on Amazon dataset. Best baseline marked with asterisk and best result in boldfaced.

TopK	Metric	Method					
		BPR	HFT	TBPR	MLP	LCMR	PNE
5	HR	8.10	10.77	15.17	21.00*	20.24	**23.52**
	NDCG	5.83	8.15	12.08	14.86*	14.51	**16.46**
	MRR	5.09	7.29	11.04	12.83*	12.63	**14.13**
10	HR	12.04	13.60	17.77	28.36*	28.36*	**31.86**
	NDCG	7.10	9.07	12.91	16.97*	16.78	**19.15**
	MRR	5.61	7.67	11.38	13.71*	13.56	**15.24**
20	HR	18.21	27.82	22.68	38.20	39.51*	**42.21**
	NDCG	8.64	12.52	14.14	18.99	19.18*	**21.75**
	MRR	6.02	8.54	11.71	14.26*	14.20	**15.95**

Table 4: Results ($\times 100$) on Cheetah dataset. Best baseline marked with asterisk and best result in boldfaced.

TopK	Metric	Method					
		BPR	HFT	TBPR	MLP	LCMR	PNE
5	HR	43.80	49.66	49.48	53.80	54.76*	**56.48**
	NDCG	39.71	36.17	42.98*	41.21	41.89	**43.45**
	MRR	36.06	31.75	38.26*	37.02	37.62	**39.11**
10	HR	49.41	55.80	54.66	61.76	63.11*	**64.24**
	NDCG	41.82	40.93	44.99*	43.81	44.60	**45.98**
	MRR	36.94	33.65	39.13*	38.10	38.74	**40.16**
20	HR	53.98	65.47	61.23	67.93	69.27*	**69.52**
	NDCG	43.16	43.79	46.82*	45.29	46.19	**47.32**
	MRR	37.30	34.45	39.58*	38.51	39.18	**40.53**

- MLP (He et al., 2017) is a neural CF approach. Note that, CFNet of PNE is an MLP with only one hidden layer.

- LCMR (Hu et al., 2018c) is a deep model for CF with unstructured text. Note that, MemNet of PNE is the same with the local MemNet of LCMR with only one-hop hidden layer.

Our method is implemented by TensorFlow (Abadi et al., 2016). Parameters are randomly initialized from Gaussian with optimizer Adam (Kingma and Ba, 2015). Learning rate is 0.001, batch size is 128, the ratio of negative sampling is 1.

3.3 Results

Results on two datasets are shown in Table 3 and Table 4, respectively. We have some observations. First, PNE outperforms the neural CF method MLP on two datasets in terms of three ranking metrics. On Amazon dataset, PNE obtains a large improvement in performance gain with relative 12.3% HR@10, 7.7% NDCG@10, and 6.2% MRR@5. On Cheetah Mobile dataset, PNE obtains a large improvement in performance gain with relative 5.0% HR@5, and 4.2% NDCG@5, and 3.9% MRR@5. Since the CFNet component of PNE is a neural CF method (with only one hidden layer), results show

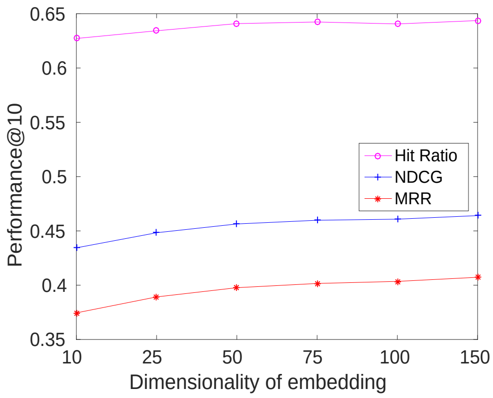

Figure 1: Dimension of embedding.

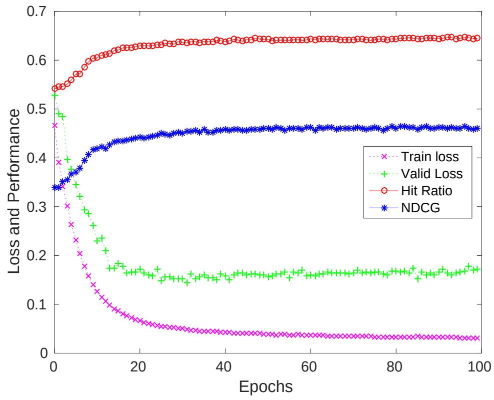

Figure 2: Loss and performance@10.

the benefit of exploiting unstructured text to alleviate the data sparsity issue faced by CF methods (BPR and MLP).

Second, PNE outperforms the traditional hybrid methods HFT and TBPR on two datasets in terms of three ranking metrics. On Amazon dataset, PNE obtains a significantly large improvement in performance gain with relative 55.0% HR@5, 28.9% NDCG@5, and 20.4% MRR@5. On Cheetah Mobile dataset, PNE still obtains reasonably large improvements with relative 17.5% HR@10, 1.8% NDCG@10, and 1.9% MRR@10. Compared with traditional hybrid methods which integrate the text content using topic modelling or word embeddings, results show the benefit of integrating text information through memory networks (and exploiting the interaction data through neural CF).

Last, PNE outperforms neural hybrid method LCMR by a large margin on Amazon dataset with relative improvements of 16.2% HR@5, 9.6% NDCG, and 7.4% MRR@5. PNE obtains reasonable improvements on Cheetah Mobile dataset with relative improvements of 3.1% HR@5, 2.8% NDCG, and 2.7% MRR. The design of CFNet of PNE is

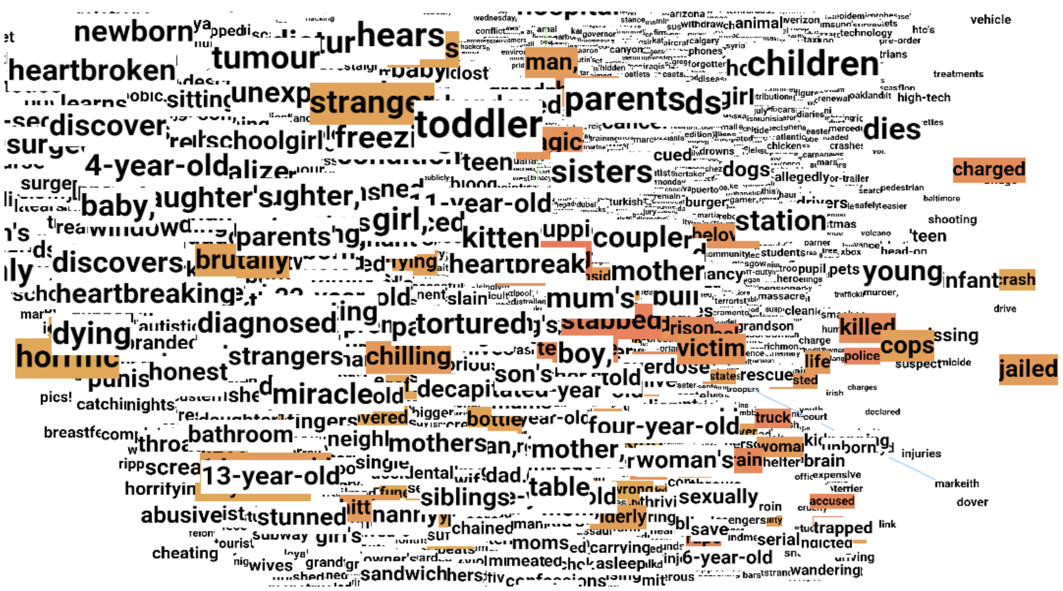

Figure 3: Visualization of word embeddings.

more reasonable than that of centralized memory module of LCMR which is equivalent to use a softmax activation between two hidden layers. The results show the effectiveness of fusing strategy in PNE to exploit unstructured text via MemNet and the interaction data via CFNet.

3.4 Analysis

We first evaluate the effects of the dimensionality of the embedding space. The x-axis in Figure 1 is the dimension of user/item and hence the dimensionality of input to CFNet is double since we adopt concatenation. It clearly indicates that the embedding should not be too small due to the possibility of information loss and the limits of expressiveness. The dimension 75 (and hence $d = 150$) is a good tradeoff between recommendation performance and computation burden.

We next show optimization curves of performance and loss (averaged over all examples) against iterations on Cheetah Mobile dataset in Figure 2. The model learns quickly in the first 20 iterations and improves slowly until 50, while training losses continue to go down and valid losses stabilize. The average time per epoch of PNE takes 68.1s and as a reference it is 34.5s for MLP using one NVIDIA TITAN Xp GPU.

3.5 Visualization

We visualize the learned word embeddings $\boldsymbol{A}$. We show that we can learn meaningful semantics for word embeddings such that words are to cluster when they have relevant semantics. We give an example to show the neighbors of the word "drug" in the 3D space by projecting the high-dimensional word vectors using TensorFlow [1] as shown in Figure 3. The top nearest neighbors of *drug* are: *shot, shoots, gang, murder, killing, rape, stabbed, truck, school, police, teenage*. We can see they are highly semantic relevant. We may also infer that school teenagers have close relationships to the drug issue from the Cheetah News corpus. This should raise a concern for society and it shows the society impact of natural language processing (Hovy and Spruit, 2016). Try our trained word embeddings [2].

4 Conclusion

We showed that relational interactions can be effectively integrated with unstructured text under a neural embedding model. Our method attentively focuses relevant words to match user preferences with user and item attentions (semantic factor) and captures nonlinear relationships between users and items (behavior factor). Experiments show better performance than five baselines on two real-world datasets in terms of three ranking metrics. We learn meaningful word embeddings and rethink the society impact of language processing technology.

Acknowledgment The work is supported by HK CERG projects 16211214/16209715/16244616, NSFC 61673202, and HKPFS PF15-16701.

[1] https://projector.tensorflow.org/

[2] https://www.dropbox.com/sh/ef74fpagf6sd137/AACXF6EnEY6QBdmJcyIW4RE0a?dl=0

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 107–114. ACM.

Yang Bao, Hui Fang, and Jie Zhang. 2014. Topicmf: simultaneously exploiting ratings and reviews for recommendation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2–8. AAAI Press.

Rose Catherine and William Cohen. 2017. Transnets: Learning to transform for recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 288–296. ACM.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10. ACM.

Gintare Karolina Dziugaite and Daniel M Roy. 2015. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443*.

Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 195–204, New York, NY, USA. ACM.

Gayatree Ganu, Noemie Elhadad, and Amélie Marian. 2009. Beyond the stars: improving rating predictions using review text content. In *International Workshop on the Web and Databases (WebDB)*, volume 9, pages 1–6.

Ruining He and Julian McAuley. 2016. Vbpr: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 144–150. AAAI Press.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee.

Dirk Hovy and Shannon L Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 591–598.

Guang-Neng Hu and Xin-Yu Dai. 2017. Integrating reviews into personalized ranking for cold start recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 708–720. PAKDD.

Guang-Neng Hu, Xin-Yu Dai, Feng-Yu Qiu, Rui Xia, Tao Li, Shu-Jian Huang, and Jia-Jun Chen. 2018a. Collaborative filtering with topic and social latent factors incorporating implicit feedback. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):23.

Guang-Neng Hu, Xin-Yu Dai, Yunya Song, Shu-Jian Huang, and Jia-Jun Chen. 2015. A synthetic approach for recommendation: combining ratings, social relations, and reviews. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Guangneng Hu, Yu Zhang, and Qiang Yang. 2018b. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 667–676. ACM.

Guangneng Hu, Yu Zhang, and Qiang Yang. 2018c. L-cmr: Local and centralized memories for collaborative filtering with unstructured text. *arXiv preprint arXiv:1804.06201*.

Haoran Huang, Qi Zhang, and Xuanjing Huang. 2017. Mention recommendation for twitter with end-to-end memory network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1872–1878. AAAI Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

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*, pages 426–434. ACM.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer*, (8):30–37.

Bo Liu, Ying Wei, Yu Zhang, Zhixian Yan, and Qiang Yang. 2018. Transferable contextual bandit for cross-domain recommendation. In *AAAI Conference on Artificial Intelligence,*.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM.

Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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*, pages 452–461. AUAI Press.

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*, pages 285–295. ACM.

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 on World Wide Web*, pages 729–739. International World Wide Web Conferences Steering Committee.

Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM.

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*, pages 1235–1244. ACM.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362. ACM.

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*, pages 425–434. ACM.