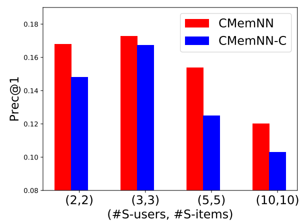
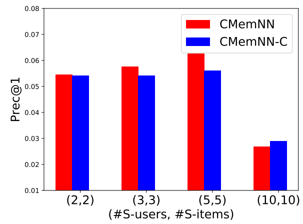
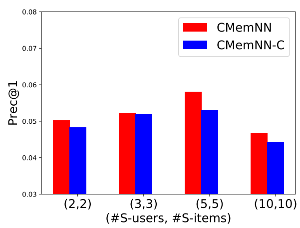
Figure 8: Performance analysis (Prec@1) on different numbers of similar users and items used in *CMemNN* and *CMemNN-C* .



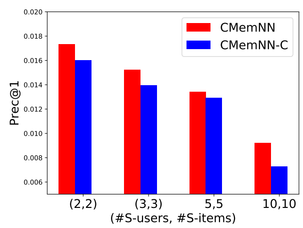
(a) MovieLens



(b) Pinterest



(c) BeerAdvocate



(d) LibraryThing

results using *Prec@1*. Both *CMemNN* and *CMemNN-C* are used with 1 *collaborative attention* layer and we report their best performance here. One can see on all datasets, similarity-based selection dominates uniform sample, which suggests that stronger similar evidences can lead better preference modeling via proposed frameworks. For *CMemNN-C*, we observe a sharp drop when using uniform sampling instead of similarity-based selection, which shows the cross *collaborative attention* allowing interactions between similar users and items in *CMemNN-C* is more sensitive to the selections of similar users and items. Based on these observations, we will only show the results based on similarity-based selection scheme for both *CMemNN* and *CMemNN-C* in the following discussions.

Similar Users and Items Impact Analysis. In Figure 8, we take experiments on different number of similar users and items fed into *CMemNN* and *CMemNN-C*. We increase the numbers from 2 to 10, and present Prec@1 on four datasets. We find that although incorporating similar users and items shows the contribution in improving preference modeling, adding a large number of them into modeling may still bring uncertainty and noise, especially when we lack of strong evidence to define similarity between users and items.

Deep Architecture Analysis. We also take experiments to show the impact of deep architectures on recommendation accuracy. Due to space limit, we only show results on two large and sparse datasets in Table 4 and 5. We find that enabling deep architecture can improve *CMemNN* and *CMemNN-C* performance at some scales. However, the improvement of *CMemNN-C* saturates when the number of layers increases, suggesting that over-interactions between similar users and items in deep *CMemNN-C* framework may bring noisy knowledge that harms the preference modeling.

Table 4: Deep architecture analysis on BeerAdvocate.

	#layer	Prec@1	Prec@3	Prec@5
CMemNN-C	1	0.0533	0.0482	0.0428
	3	0.0556	0.0480	0.0422
	5	0.0428	0.0351	0.0304
CMemNN	1	0.0546	0.0478	0.0427
	3	0.0546	0.0472	0.0446
	5	0.0580	0.0512	0.0472

Table 5: Deep architecture analysis on LibraryThing.

	#layer	Prec@1	Prec@3	Prec@5
CMemNN-C	1	0.0163	0.0105	0.097
	3	0.0153	0.0100	0.0086
	5	0.0130	0.0074	0.0067
CMemNN	1	0.0152	0.0087	0.0077
	3	0.0173	0.0102	0.0086
	5	0.0166	0.0106	0.0085

5 RELATED WORK

Recommender systems focus on analyzing patterns of interest in items to provide personalized suggestions to users. Several works have studied recommendation problems based on users' implicit feedback and the proposed algorithms can be mainly divided into two branches: pointwise methods [11, 21] and pairwise methods [24, 25]. Pointwise methods aim to fit a numeric value associated with each evaluated item. These methods view positive feedback as high preference scores and use several strategies to sample negative feedback as low preference scores. Then existing matrix factorization methods [26–28] can be used to fit the preference scores. Pairwise methods always consider implicit feedback as relative relationships indicating that users show higher preference on positive feedback than on negative feedback. In [24], Rendle et al. propose a bayesian personalized ranking (BPR) framework. Following this, various ideas have been proposed that incorporate different types of contextual information into the BPR framework. However, these methods are mostly based on representing users and items with low-dimensional latent factors and only consider linear interactions between user and item parameters. To release the linear limitation, deep learning, with its flexible architecture and capability of capturing non-linear property, has attracted many research interests and brings great success in recommender systems [8, 30, 31, 33, 39, 42]. Wang et al. [33] proposed a collaborative deep learning (CDL) method for top-k recommendation, which is based on a hierarchical Bayesian model combining stacked denoising autoencoder with probabilistic matrix factorization (PMF). Further, Li et al. [16] extended CDL to a collaborative variational autoencoder (CVAE), which replaced the deep neural component of CDL with variational autoencoder. Summarizing this line of research, they use auto-encoder frameworks to extract informative hidden representations from auxiliary information [39], such as text reviews and images. Another line of auto-encoder based deep recommendation methods is to use the reconstruction layer to directly recover users' missing feedback without modeling auxiliary knowledge [31, 46]. For example, AutoRec [30] takes users' or items' partial observed feedback vectors as input, and aims to recovery all