HAM: Hybrid Associations Models for Sequential Recommendation (Supplementary Materials)

Bo Peng, Zhiyun Ren, Srinivasan Parthasarathy, *Member, IEEE,* and Xia Ning*, *Member, IEEE*

S1 Parameter Study of $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ on Comics

TABLE S1: Parameter Study of $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ on Comics in 80-20-CUT

parameter	d	n_h	n_l	n_p	p	Recall@5	Recall@10
d	200	7	2	5	3	0.1354	0.1914
	400	7	2	5	3	**0.1385**	**0.1945**
	600	7	2	5	3	0.1378	0.1929
n_h	400	6	2	5	3	0.1372	0.1930
	400	7	2	5	3	0.1385	0.1945
	400	8	2	5	3	**0.1389**	0.1945
	400	9	2	5	3	0.1387	**0.1948**
n_l	400	7	0	5	3	0.1223	0.1767
	400	7	1	5	3	†**0.1398**	†**0.1953**
	400	7	2	5	3	0.1385	0.1945
	400	7	3	5	3	0.1352	0.1902
n_p	400	7	2	4	3	0.1380	0.1930
	400	7	2	5	3	**0.1385**	**0.1945**
	400	7	2	6	3	0.1367	0.1944
p	400	7	2	5	1	0.1299	0.1874
	400	7	2	5	2	0.1331	0.1888
	400	7	2	5	3	**0.1385**	**0.1945**
	400	7	2	5	4	0.1311	0.1859

In this table, d, n_h/n_l, n_p and p are the dimension of embeddings, number of items in high-order/low-order associations, number of items to calculate recommendation errors during training and the order of item synergies. The best performance overall is †**bold** and the best performance in each row block is **bold**. The best results based on validation sets and the corresponding parameters tuned on the validation sets are underlined. The "parameter" column presents the parameter to be studied in each row block.

Table S1 presents the results of parameter study on the Comics dataset in 80-20-CUT. As shown in this table, similar to that on CDs and Children, on Comics, $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ still achieves the best performance with a large d (i.e., 400) and a small n_l (i.e., 2). Similar to that on Children, $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ achieves the best performance with relatively large n_h, n_p (n_h=8, n_p=5) and p (i.e, 3). On Comics, compared to $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ without item synergies (i.e., p=1), $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ with item synergies achieves significant improvement (e.g., p=3: 0.1385 vs p=1: 0.1299 on Recall@5, 6.6% improvement), demonstrating that item synergies could improve the recommendation performance on most benchmark datasets.

- *Bo Peng is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, 43210. E-mail: peng.707@buckeyemail.osu.edu*
- *Srinivasan Parthasarathy and Xia Ning are with the Department of Biomedical Informatics, and the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, 43210. E-mail: srini@cse.ohio-state.edu, ning.104@osu.edu*
- *Zhiyun Ren is with the Department of Biomedical Informatics, The Ohio State University, Columbus, OH, 43210. E-mail: ren.685@osu.edu*
- **Corresponding author*

Manuscript received April 19, 2005; revised August 26, 2015.

S2 Parameter Study of SASRec 3-LOS

TABLE S2: Parameter Study of SASRec on Comics in 3-LOS

parameter	d	n	h	Recall@5	Recall@10
d	200	600	1	0.2543	0.3209
	400	600	1	0.2611	0.3250
	600	600	1	0.2750	0.3353
	800	600	1	0.0374	0.0588
n	600	200	1	0.2714	0.3333
	600	400	1	0.0459	0.0737
	600	600	1	0.2750	0.3353
	600	800	1	OOM	OOM
h	600	600	1	0.2750	0.3353
	600	600	2	0.0357	0.0587
	600	600	4	OOM	OOM

In this table, d, n and h are the dimension of embeddings, maximum sequence length and the number of head of the multi-head attention. The best results on validation sets and the corresponding parameters are underlined. The "parameter" column presents the parameters to be studied in each row block. The "OOM" represents the out-of-memory issue.

In this section, we present more details of the parameter study on SASRec as discussed in Section 6.3 of the main manuscript. Table S2 presents the results of SASRec on the validation set of Comics in 3-LOS with respect to different parameters. In Table S2, the "OOM" represents the out-of-memory issue and the best results on the validation set and the corresponding parameters are underlined. Table S2 shows that the performance of SASRec changes dramatically by slightly changing the parameters. For example, when the dimension of embeddings d is changed from 600 to 800 and all the other parameters are fixed, the performance of SASRec on Recall@10 decreases dramatically from 0.3353 to 0.0588 (6-fold decrease). The similar trend could also be found in the other parameters (i.e., maximum sequence length n and the number of head in the multi-head attention h). For example, when h is changed from 1 to 2, the performance decreases from 0.3353 to 0.0587 on Recall@10

TABLE S3: Best Parameters for $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ and Baseline Methods

	Dataset	$\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$					HGN			SASRec			Caser				
		d	n_h	n_l	n_p	p	d	L	T	d	n	h	d	L	T	nv	nh
80-20-CUT 80-3-CUT	CDs	400	5	2	3	2	200	5	2	400	600	1	200	5	4	2	16
	Books	400	9	2	7	2	400	4	4	400	600	1	200	6	4	2	8
	Children	400	6	1	4	3	200	2	4	400	200	1	100	4	4	2	16
	Comics	400	7	2	5	3	200	2	6	400	400	1	100	4	4	2	16
	ML-20M	400	9	3	2	3	100	5	3	400	400	4	100	6	2	4	8
	ML-1M	400	7	2	3	3	100	4	4	200	600	1	200	6	2	2	8
3-LOS	CDs	400	4	2	7	2	200	4	3	400	400	4	200	4	4	2	16
	Books	400	9	2	9	2	400	2	6	400	400	1	200	5	3	2	8
	Children	400	6	1	4	3	100	2	5	400	200	1	200	4	4	2	8
	Comics	400	7	1	5	3	200	2	5	600	600	1	200	4	4	2	8
	ML-20M	400	8	3	3	3	100	6	3	400	400	4	200	4	4	2	8
	ML-1M	400	8	2	2	3	100	3	4	200	600	2	200	5	2	2	16

In this table, in $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$, d, n_h/n_l, n_p and p are the embedding dimension, number of items in high-order/low-order associations, number of items to calculate recommendation errors during training and the order of item synergies, respectively. In HGN, d, L and T are the embedding dimension, length of the subsequences and the number of items used as targets in training. In SASRec, d, n and h are the embedding dimension, maximum sequence length and the number of head of the multi-head attention. In Caser, d, L, T, nv and nh are the embedding dimension, the length of the subsequences, the number of items used as targets in training the number of vertical filters and the number of horizontal filters, respectively. The $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$, HGN, SASRec, Caser columns present the best parameters on validation sets and thus are used in testing for $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$, HGN, SASRec and Caser, respectively.

(6-fold decrease). These results indicate that SASRec could be very sensitive to parameters. Table S2 also shows that SASRec requires a large amount of memory in training, as we got the out-of-memory issue when using large n (i.e., 800) and h (i.e., 4). This limits the real-application scenarios that SASRec could be used for. Our HAM methods do not suffer from such memory issues.

S3 Parameters for Reproducibility

In this Section, we report the parameters corresponding to the best Recall@10 results of $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ and all the other baseline methods for the sake of reproducibility. These parameters are identified through tuning on the validation sets. Recall that we use the same training and validation sets in 80-20-CUT and 80-3-CUT (Fig. 2 in the main manuscript). As a result, the best parameters based on the tuning on the validation sets are identical in 80-20-CUT and 80-3-CUT.

We implement HAM in python with pytorch 1.2.0 (https://pytorch.org). We used Adam optimizer with learning rate 1e-3 and regularization factor 1e-3 on all the datasets. The dimension of embeddings d, the number of items in high-order/low-order associations n_h/n_l, the number of items to calculate recommendation errors during training n_p, and the order of item synergies that are specific for each dataset are reported in the $\mathtt{HAM}^{\mathtt{s}}_{\mathtt{m}}$ column of Table S3. Our HAM implementation is publicly available at https://github.com/BoPeng112/HAM.

For HGN, we used the implementation provided by the authors in github[1]. We used the default Adam optimizer with learning rate 1e-3 and regularization factor 1e-3 on all the datasets. The dimension of embeddings, denoted as d, the length of the subsequences, denoted as L

For SASRec, we used the implementation provided by the authors in github[2] as well. We used the default Adam optimizer with learning rate 1e-3 and the exponential decay rate for the second-moment estimates beta2 0.98. The dimension of embeddings, denoted as d, the maximum sequence length, dented as n and the number of heads of the multi-head attention, denoted as h, of each dataset are reported in the SASRec column of Table S3.

For Caser, we also used the pytorch implementation suggested by the authors in github[3]. We used the default Adam optimizer with learning rate 1e-3 and regularization factor 1e-6. The dimension of embeddings, denoted as d, the length of the subsequences, denoted as L, the number of items used as targets in training, denoted as T, the number of vertical filters in CNNs, denoted as nv, and the number of horizontal filters in CNNs, denoted as nh, of each dataset are reported in the Caser column of Table S3

1. https://github.com/allenjack/HGN
2. https://github.com/kang205/SASRec
3. https://github.com/graytowne/caser_pytorch