Memory-Based One-Class Collaborative Filtering

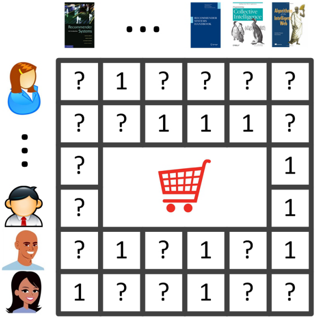
Weike Pan

College of Computer Science and Software Engineering
Shenzhen University

Outline

Recommendation with Implicit Feedback

- We may represent users' implicit feedback in a ***matrix*** form:



- If we can estimate the missing values (denoted as "?") in the matrix or rank the items directly, we can make recommendations for each user.

Notations

Table: Some notations.

n	user number
m	item number
$u \in \{1, 2, \ldots, n\}$	user ID
$i, j \in \{1, 2, \ldots, m\}$	item ID
$\hat{r}_{ui}$	predicted rating of user u on item i
$\mathcal{I}$	the whole item set
$\mathcal{U}$	the whole user set
$\mathcal{I}_u$	a set of items preferred by user u
$\mathcal{U}_k$	a set of users that prefers item k
$\mathcal{U}_j$	a set of users that prefers item j
$\mathcal{N}_j$	a set of top-K nearest neighbors of item j
$\mathcal{N}_u$	a set of top-K nearest neighbors of user u

Jaccard Index

- The similarity between item k and item j is as follows,

$$s_{kj} = \frac{|\mathcal{U}_k \cap \mathcal{U}_j|}{|\mathcal{U}_k \cup \mathcal{U}_j|} \tag{1}$$

- The default value is $s_{kj} = 0$

Cosine Similarity

- The similarity between item k and item j is as follows,

$$s_{kj} = \frac{|\mathcal{U}_k \cap \mathcal{U}_j|}{\sqrt{|\mathcal{U}_k|}\sqrt{|\mathcal{U}_j|}} \tag{2}$$

- The default value is $s_{kj} = 0$

Cosine Similarity - Penalty on Popular Item

- The similarity between item k and item j is as follows,

$$s_{kj} = \frac{|\mathcal{U}_k \cap \mathcal{U}_j|}{|\mathcal{U}_k|^{\alpha} |\mathcal{U}_j|^{1-\alpha}} \tag{3}$$

  where $0.5 \leq \alpha \leq 1$.

- Notes
  - When $\alpha = 0.5$, it reduces to the cosine similarity
  - A large value of α (i.e., $\alpha > 0.5$) will give a penalty on a popular item k
  - When $\alpha = 1$, it reduces to the confidence in association rule mining (will be discussed soon)

Cosine Similarity - Inverse User Frequency

- The similarity between item k and item j is as follows [Breese et al., 1998],

$$s_{kj} = \frac{\sum_{w \in \mathcal{U}_k \cap \mathcal{U}_j} \frac{1}{\log(1+|\mathcal{I}_w|)}}{\sqrt{|\mathcal{U}_k|}\sqrt{|\mathcal{U}_j|}} \tag{4}$$

- Notes
  - It is a well-known trick in information retrieval

Confidence in Association Rule Mining

- The confidence (or probability) that a user who preferred item k will prefer item j is as follows [Kim and Kim, 2003],

$$s_{kj} = \frac{|\mathcal{U}_k \cap \mathcal{U}_j|}{|\mathcal{U}_k|} \tag{5}$$

- Notes
  - Association rule mining can be considered as a special case of item-based OCCF
  - The support is defined as $s_{kj} = \frac{|\mathcal{U}_k \cap \mathcal{U}_j|}{|\mathcal{U}|}$

Normalization

- Once we have obtained the similarity matrix $[s_{kj}]_{m\times m}, k, j = 1, \ldots, m$, we can normalize the similarity as follows [Karypis, 2001],

$$\bar{s}_{kj} = \frac{s_{kj}}{\max_{j', j' \neq k} s_{kj'}} \tag{6}$$

- Notes
  - It is NOT a global normalization, i.e., $\frac{s_{kj}}{\max_{k', j'} s_{k'j'}}$
  - It makes the similarity "between item k and item j" and the similarity "between item k' and item j" more comparable

Neighborhood Selection

- Select the top-K most nearest items of item j w.r.t. the similarity measurement, e.g., $\mathcal{N}_j$.

Item-Based OCCF

- The prediction rule is as follows,

$$\hat{r}_{uj} = \sum_{k \in \mathcal{I}_u \cap \mathcal{N}_j} s_{kj} \tag{7}$$

  where s_{kj} is the similarity (or confidence in association rule mining) between item k and item j.

- Note that we may have $|\mathcal{I}_u \cap \mathcal{N}_j| < K$, but that is usually acceptable for OCCF

User-Based OCCF

- The prediction rule is as follows,

$$\hat{r}_{uj} = \sum_{w \in \mathcal{U}_j \cap \mathcal{N}_u} s_{wu} \tag{8}$$

  where s_{wu} is the similarity (or confidence in association rule mining) between user w and user u.

- Note that we may have $|\mathcal{U}_j \cap \mathcal{N}_u| < K$, but that is usually acceptable for OCCF

Data Set

- We use the files u1.base and u1.test of MovieLens100K[1] as our training data and test data, respectively.
- user number: $n = 943$; item number: $m = 1682$.
- u1.base (training data): 80000 rating records, and the density (or sparsity) is $80000/943/1682 = 5.04\%$.
- u1.test (test data): 20000 rating records.
- Pre-processing (for simulation): we only keep the (user, item) pairs with ratings 4 or 5 in u1.base and u1.test as preferred (user, item) pairs, and remove all other records. Finally, we obtain u1.base.OCCF and u1.test.OCCF.

[1] http://grouplens.org/datasets/

Implementation Details

- How to store the implicit feedback if we use item-based method?
  - For similarity measurement s_{kj}: $\mathcal{U}_k, \mathcal{U}_j$
  - In prediction rule $\hat{r}_{uj}$: $\mathcal{I}_u, \mathcal{N}_j$
- How to store the implicit feedback if we use user-based method?
  - For similarity measurement s_{wu}: $\mathcal{I}_w, \mathcal{I}_u$
  - In prediction rule $\hat{r}_{uj}$: $\mathcal{U}_j, \mathcal{N}_u$

Evaluation Metrics

- *Pre@5*: The precision of user u is defined as,

$$Pre_u@k = \frac{1}{k}\sum_{\ell=1}^{k}\delta(i(\ell) \in \mathcal{I}_u^{te}),$$

where $\delta(x) = 1$ if x is true and $\delta(x) = 0$ otherwise. Then, we have $Pre@k = \sum_{u\in\mathcal{U}^{te}} Pre_u@k/|\mathcal{U}^{te}|$.

- *Rec@5*: The recall of user u is defined as,

$$Rec_u@k = \frac{1}{|\mathcal{I}_u^{te}|}\sum_{\ell=1}^{k}\delta(i(\ell) \in \mathcal{I}_u^{te}),$$

which means how many preferred items are recommended in the top-k list. Then, we have $Rec@k = \sum_{u\in\mathcal{U}^{te}} Rec_u@k/|\mathcal{U}^{te}|$.

Results

Table: Prediction performance of PopRank and memory-based OCCF (Jaccard Index, $K = 50$) on MovieLens100K (u1.base.OCCF, u1.test.OCCF).

	PopRank	Item-based	User-based	Hybrid
Pre@5	0.2338	0.3632	0.3768	**0.3978**
Rec@5	0.0571	0.1102	0.1207	**0.1314**

Observation: the hybrid method performs best, which is expected.

Conclusion

Three major components

- Similarity measurement and related techniques
- Neighborhood selection
- Prediction rule

Homework

- Reading: chapter 2 of the Chinese reference book by Liang Xiang (i.e., *Action in Recommender Systems*).
- More references:
  - user-based + item-based: [Wang et al., 2006], [Symeonidis et al., 2008]
  - item-based: [Deshpande and Karypis, 2004]
  - user-based: [Sarwar et al., 2000]
  - recent papers: [Sigurbjörnsson and van Zwol, 2008], [Aiolli, 2013]

Aiolli, F. (2013).
Efficient top-n recommendation for very large scale binary rated datasets.
In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 273–280, New York, NY, USA. ACM.

Breese, J. S., Heckerman, D., and Kadie, C. (1998).
Empirical analysis of predictive algorithms for collaborative filtering.
In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Deshpande, M. and Karypis, G. (2004).
Item-based top-n recommendation algorithms.
ACM Trans. Inf. Syst., 22(1):143–177.

Karypis, G. (2001).
Evaluation of item-based top-n recommendation algorithms.
In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 247–254, New York, NY, USA. ACM.

Kim, C. and Kim, J. (2003).
A recommendation algorithm using multi-level association rules.
In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, WI '03, pages 524–, Washington, DC, USA. IEEE Computer Society.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000).
Analysis of recommendation algorithms for e-commerce.
In *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, EC '00, pages 158–167, New York, NY, USA. ACM.

Sigurbjörnsson, B. and van Zwol, R. (2008).
Flickr tag recommendation based on collective knowledge.
In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 327–336, New York, NY, USA. ACM.

Symeonidis, P., Nanopoulos, A., Papadopoulos, A. N., and Manolopoulos, Y. (2008).

Nearest-biclusters collaborative filtering based on constant and coherent values.
Inf. Retr., 11(1):51–75.

Wang, J., de Vries, A. P., and Reinders, M. J. T. (2006).
Unifying user-based and item-based collaborative filtering approaches by similarity fusion.
In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 501–508, New York, NY, USA. ACM.