Average Filling

Weike Pan

College of Computer Science and Software Engineering
Shenzhen University

Outline

Data Modeling in Recommender Systems

- behaviors
- $\rightarrow$ **feedback**
- $\rightarrow$ **preferences**
- $\rightarrow$ profile
- $\rightarrow$ recommendation

- Our focus: **feedback** $\rightarrow$ **preferences**

Explicit Feedback

- A user's preference is explicitly expressed, e.g., numerical ratings or grade scores in $\{1, 2, 3, 4, 5\}$, thumbs up, etc.

★★★★★ 

Implicit Feedback

- A user's preference is implicitly expressed, e.g., examination (click, browsing, download, collection), transaction, etc.

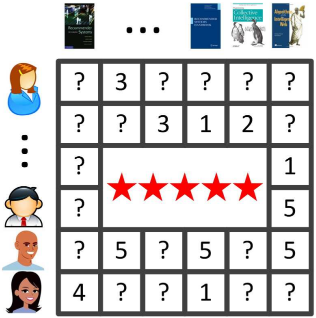
 

One-Class Feedback

- We only know whether the user *clicked or not*, *bought or not*, *liked or not*, etc.

Recommendation with Explicit Feedback (1/2)

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



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

Recommendation with Explicit Feedback (2/2)

- We may represent users' explicit feedback in a ***record*** form:
  (Grace, *Recommender Systems Handbook*, 4)
  (Grace, *Collective Intelligence*, 5)
  (John, *An Introduction to Recommender Systems* 4)
  (Jacky, *Collective Intelligence*, 5)
  (Rebecca, *Algorithms of the Intelligent Web*, 5)
  ...

Notations (1/2)

- user number: n
- item number: m
- user ID: $u \in \{1, 2, \ldots, n\}$
- item ID: $i \in \{1, 2, \ldots, m\}$
- grade score set (or rating range): $\mathbb{G}$, e.g., $\mathbb{G} = \{1, 2, 3, 4, 5\}$
- observed rating of user u on item i: $r_{ui} \in \mathbb{G}$
- predicted rating of user u on item i: $\hat{r}_{ui}$

Notations (2/2)

- Training data
  - indicator variable: $y_{ui} = \begin{cases} 1, & \text{if } (u, i, r_{ui}) \text{ is observed} \\ 0, & \text{if } (u, i, r_{ui}) \text{ is not observed} \end{cases}$
  - rating records: $\mathcal{R} = \{(u, i, r_{ui})\}$
  - rating matrix: $\mathbf{R} \in \{\mathbb{G} \cup ?\}^{n \times m}$
  - number of observed ratings: $p = \sum_{u,i} y_{ui} = |\mathcal{R}|$
  - density (or sometimes called sparsity): $\frac{p}{nm}$
- Test data
  - rating records: $\mathcal{R}^{te} = \{(u, i, r_{ui})\}$

Problem Definition

Recommendation with Explicit Feedback

- **Input** (training data): observed rating records $\mathcal{R} = \{(u, i, r_{ui})\}$
- **Output**: for each record (u, i, r_{ui}) in $\mathcal{R}^{te}$, estimate the rating $\hat{r}_{ui}$
- **Evaluation**: how accurate is the predicted rating $\hat{r}_{ui}$

Statistics (1/2)

Statistics of training data,

- global average rating: $\bar{r} = \sum_{u,i} y_{ui} r_{ui} / \sum_{u,i} y_{ui}$
- average rating of user u: $\bar{r}_u = \sum_i y_{ui} r_{ui} / \sum_i y_{ui}$
- average rating of item i: $\bar{r}_i = \sum_u y_{ui} r_{ui} / \sum_u y_{ui}$

Statistics (2/2)

- bias of user u: $b_u = \sum_i y_{ui}(r_{ui} - \bar{r}_i) / \sum_i y_{ui}$
  - Is user u nice (when $b_u > 0$) or critical (when $b_u < 0$)?
- bias of item i: $b_i = \sum_u y_{ui}(r_{ui} - \bar{r}_u) / \sum_u y_{ui}$
  - Is item i popular (when $b_i > 0$) or not (when $b_i < 0$)?

Prediction Rule

Predicted rating of user u on item i,

- user average, $\hat{r}_{ui} = \bar{r}_u$
- item average, $\hat{r}_{ui} = \bar{r}_i$
- mean of user average and item average, $\hat{r}_{ui} = \bar{r}_u/2 + \bar{r}_i/2$
- user bias and item average, $\hat{r}_{ui} = b_u + \bar{r}_i$
- user average and item bias, $\hat{r}_{ui} = \bar{r}_u + b_i$
- global average, user bias and item bias, $\hat{r}_{ui} = \bar{r} + b_u + b_i$

Analysis

- From the perspective of **ranking**, $\hat{r}_{ui} = \bar{r}_u$ can not be used to rank candidate items
- From the perspective of **personalization**, $\hat{r}_{ui} = \bar{r}_i$ is not a personalization method
- Average Filling works well for **sparse data**, i.e., when $p/n/m$ is very small
- Average Filling is usually used as **a baseline method** when designing a new recommendation algorithm

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.

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

Evaluation Metrics (1/2)

- Mean Absolute Error (MAE)

$$MAE = \sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} |r_{ui} - \hat{r}_{ui}| / |\mathcal{R}^{te}|$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} (r_{ui} - \hat{r}_{ui})^2 / |\mathcal{R}^{te}|}$$

- Performance: the smaller the better.

Evaluation Metrics (2/2)

- MAE and RMSE on different segments of users w.r.t. users' activeness on training data
  - Sometimes, we use the user segmentation of test data for convenience if the distribution of training data and test data are similar
- MAE and RMSE on different categories of items w.r.t. items' popularity on training data

Implementation Details

- First scan of the training data: $\bar{r}$, $\bar{r}_u$, $\bar{r}_i$
  - the default value of $\bar{r}_u$ and $\bar{r}_i$ is $\bar{r}$ (instead of 0) when some user or item is not observed in the training data
- Second scan of the training data: b_u, b_i
  - the default value of b_u and b_i is 0 when some user or item is not observed in the training data

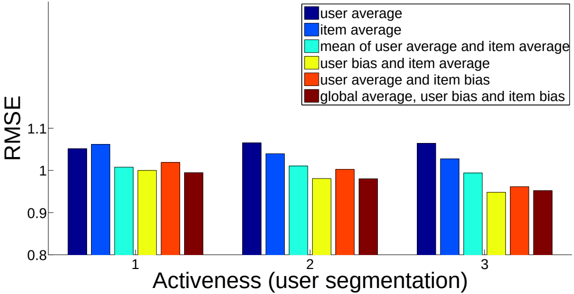
Results (1/3)

Table: Prediction performance of Average Filling (AF) on MovieLens100K (u1.base, u1.test).

Prediction rule	RMSE	MAE
$\hat{r}_{ui} = \bar{r}_u$	1.0630	0.8502
$\hat{r}_{ui} = \bar{r}_i$	1.0334	0.8276
$\hat{r}_{ui} = \bar{r}_u/2 + \bar{r}_i/2$	0.9985	0.8085
$\hat{r}_{ui} = b_u + \bar{r}_i$	**0.9602**	**0.7574**
$\hat{r}_{ui} = \bar{r}_u + b_i$	0.9758	0.7696
$\hat{r}_{ui} = \bar{r} + b_u + b_i$	0.9623	0.7613

Observation: $\hat{r}_{ui} = b_u + \bar{r}_i$ performs best.

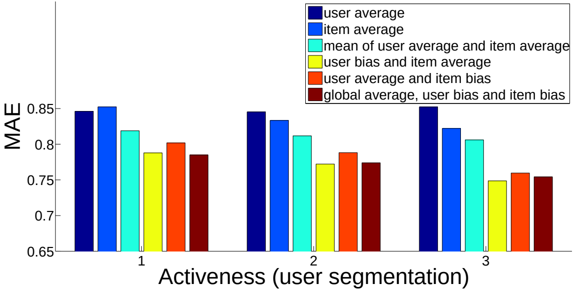
Results (2/3)



ID	Rating #	User #
1	$(0, 20]$	198
2	$(20, 50]$	118
3	$(50,$	143

Observation: consistent to the overall performance; more accurate for active users.

Results (3/3)



ID	Rating #	User #
1	$(0, 20]$	198
2	$(20, 50]$	118
3	$(50,$	143

Observation: consistent to the overall performance; more accurate for active users.

Conclusion

- Representation in matrix form and record form
- (Mathematical) problem definition
- Two stages of training and test
- Prediction rules of average filling

Homework

- Implement AF and conduct empirical studies on u2.base, u2.test of MovieLens100K
- Design a new rating prediction method to beat AF