Factored Item Similarity Models ($FISM_{rmse}$)

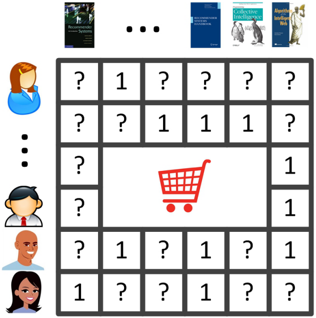
Weike Pan

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 (1/2)

Table: Some notations.

n	user number
m	item number
$u \in \{1, 2, \ldots, n\}$	user ID
$i, i' \in \{1, 2, \ldots, m\}$	item ID
r_{ui}	observed rating of user u on item i
$\mathcal{P}$	the whole set of observed (user, item) pairs
$\mathcal{A}, \lvert\mathcal{A}\rvert = \rho\lvert\mathcal{P}\rvert$	a sampled set of unobserved (user, item) pairs

Notations (2/2)

Table: Some notations.

$b_u \in \mathbb{R}$	user bias
$b_i \in \mathbb{R}$	item bias
$d \in \mathbb{R}$	number of latent dimensions
$V_{i\cdot}, W_{j\cdot} \in \mathbb{R}^{1\times d}$	item-specific latent feature vector
$\mathbf{V}, \mathbf{W} \in \mathbb{R}^{m\times d}$	item-specific latent feature matrix
$\hat{r}_{ui}$	predicted rating of user u on item i
T	iteration number in the algorithm

Factored Item Similarity Model (FISM)

FISM with different loss functions,

- $FISM_{rmse}$
- $FISM_{auc}$

Prediction Rule

The predicted rating of user u on item i,

$$\hat{r}_{ui} = b_u + b_i + \bar{U}_{u\cdot}^{-i} V_{i\cdot}^T \tag{1}$$

where,

$$\bar{U}_{u\cdot}^{-i} = \frac{1}{|\mathcal{I}_u \backslash \{i\}|^{\alpha}} \sum_{i' \in \mathcal{I}_u \backslash \{i\}} W_{i'\cdot}, \ 0 \leq \alpha \leq 1$$

Note that when $i \notin \mathcal{I}_u$, $\mathcal{I}_u = \mathcal{I}_u \backslash \{i\}$.

Objective Function

The objective function of $FISM_{rmse}$,

$$\min_{\Theta} \sum_{(u,i)\in\mathcal{P}\cup\mathcal{A}} f_{ui} \tag{2}$$

where $\Theta = \{V_{i\cdot}, W_{i'\cdot}, b_u, b_i\}, i, i' = 1, \ldots, m, u = 1, \ldots, n$, and

$$f_{ui} = \frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \frac{\alpha_v}{2}||V_{i\cdot}||_F^2 + \frac{\alpha_w}{2} \sum_{i'\in\mathcal{I}_u\backslash\{i\}} ||W_{i'\cdot}||_F^2 + \frac{\beta_u}{2}b_u^2 + \frac{\beta_v}{2}b_i^2$$

Notes:

- $\mathcal{P}$ is the whole set of observed (user, item) pairs
- $\mathcal{A}$ is a sampled set of unobserved (user, item) pairs
- According to the loss function, we can see that $FISM_{rmse}$ is a pointwise method

Gradients

For each $(u,i) \in \mathcal{P} \cup \mathcal{A}$, we have the gradients,

$$
\begin{aligned}
\nabla b_u &= \frac{\partial f_{ui}}{\partial b_u} = -e_{ui} + \beta_u b_u \\
\nabla b_i &= \frac{\partial f_{ui}}{\partial b_i} = -e_{ui} + \beta_v b_i \\
\nabla V_{i\cdot} &= \frac{\partial f_{ui}}{\partial V_{i\cdot}} = -e_{ui} \bar{U}_{u\cdot}^{-i} + \alpha_v V_{i\cdot} \\
\nabla W_{i'\cdot} &= \frac{\partial f_{ui}}{\partial W_{i'\cdot}} = -e_{ui} \frac{1}{|\mathcal{I}_u \backslash \{i\}|^{\alpha}} V_{i\cdot} + \alpha_w W_{i'\cdot}, i' \in \mathcal{I}_u \backslash \{i\}
\end{aligned}
$$

where $e_{ui} = r_{ui} - \hat{r}_{ui}$. Note that $r_{ui} = 1$ if $(u,i) \in \mathcal{P}$, and $r_{ui} = 0$ if $(u,i) \in \mathcal{A}$.

Update Rules

For each $(u, i) \in \mathcal{P} \cup \mathcal{A}$, we have the update rules,

$$
\begin{aligned}
b_u &= b_u - \gamma \nabla b_u \\
b_i &= b_i - \gamma \nabla b_i \\
V_{i\cdot} &= V_{i\cdot} - \gamma \nabla V_{i\cdot} \\
W_{i'\cdot} &= W_{i'\cdot} - \gamma \nabla W_{i'\cdot}, i' \in \mathcal{I}_u \backslash \{i\}
\end{aligned}
$$

where $e_{ui} = r_{ui} - \hat{r}_{ui}$. Note that $r_{ui} = 1$ if $(u, i) \in \mathcal{P}$, and $r_{ui} = 0$ if $(u, i) \in \mathcal{A}$.

Algorithm

1: Initialize the model parameters Θ
2: **for** $t = 1, \ldots, T$ **do**
3: Randomly pick up a set $\mathcal{A}$ with $|\mathcal{A}| = \rho|\mathcal{P}|$
4: **for** each $(u, i) \in \mathcal{P} \cup \mathcal{A}$ in a random order **do**
5: Calculate $\bar{U}_{u\cdot}^{-i} = \frac{1}{|\mathcal{I}_u \backslash \{i\}|^{\alpha}} \sum_{i' \in \mathcal{I}_u \backslash \{i\}} W_{i'\cdot}$
6: Calculate $\hat{r}_{ui} = b_u + b_i + \bar{U}_{u\cdot}^{-i} V_{i\cdot}^T$
7: Calculate $e_{ui} = r_{ui} - \hat{r}_{ui}$
8: Update the $b_u, b_i, V_{i\cdot}$ and $W_{i'\cdot}, i' \in \mathcal{I}_u \backslash \{i\}$
9: **end for**
10: **end for**

Figure: The SGD algorithm for $FISM_{rmse}$.

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/

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 $Rec@k = \sum_{u \in \mathcal{U}^{te}} Rec_u@k / |\mathcal{U}^{te}|$.

Initialization of Model Parameters

We use the statistics of training data to initialize the model parameters,

$$
\begin{aligned}
b_u &= \sum_{i=1}^{m} y_{ui}/m - \mu \\
b_i &= \sum_{u=1}^{n} y_{ui}/n - \mu \\
V_{ik} &= (r - 0.5) \times 0.01, k = 1, \ldots, d \\
W_{i'k} &= (r - 0.5) \times 0.01, k = 1, \ldots, d
\end{aligned}
$$

where r $(0 \leq r < 1)$ is a random variable, and $\mu = \sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui}/n/m$.

Parameter Configurations

We fix $\alpha = 0.5$ and $\gamma = 0.01$, and search the best values of the following parameters,

- $\alpha_v = \alpha_w = \beta_u = \beta_v \in \{0.001, 0.01, 0.1\}$
- $T \in \{100, 500, 1000\}$

Finally, we use $\alpha = 0.5$, $\gamma = 0.01$, $\rho = 3$, $d = 20$, $\alpha_v = \alpha_w = \beta_u = \beta_v = 0.001$ and $T = 100$, which performs best in our experiments.
It takes about 75 seconds for training.

Results

Table: Prediction performance of PopRank and $FISM_{rmse}$ on MovieLens100K (u1.base.OCCF, u1.test.OCCF).

	PopRank	$FISM_{rmse}$
Pre@5	0.2338	**0.3846**
Rec@5	0.0571	**0.1270**

Conclusion

- Learning factored (item, item) similarities is helpful.

Homework

- Implement $FISM_{rmse}$ and conduct empirical studies on u2.base.OCCF, u2.test.OCCF of MovieLens100K with similar pre-processing
- Read the KDD 2013 paper [Kabbur et al., 2013], and study the algorithm of $FISM_{auc}$

Kabbur, S., Ning, X., and Karypis, G. (2013).
Fism: Factored item similarity models for top-n recommender systems.
In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 659–667, New York, NY, USA. ACM.