

# Research on Content-Aware Collaborative Filtering

## Content-aware Bayesian Personalized Ranking

Liucheng Xu  
2012080173

College of Computer Science and Software Engineering  
Shenzhen University

April 20, 2016

- 1 BPR
- 2 Adaptive Sampling Strategy
- 3 Content-Aware and Adaptive BPR
- 4 Experiment

- 1 BPR
- 2 Adaptive Sampling Strategy
- 3 Content-Aware and Adaptive BPR
- 4 Experiment

- 1 BPR
- 2 Adaptive Sampling Strategy
- 3 Content-Aware and Adaptive BPR
- 4 Experiment

- 1 BPR
- 2 Adaptive Sampling Strategy
- 3 Content-Aware and Adaptive BPR
- 4 Experiment

Table: Some notations

$s$	user number
$t$	item number
$k$	latent dimension (category number)
$U, Y^u \in \mathbb{R}^{s \times k}$	user latent matrix
$V, Y^v \in \mathbb{R}^{t \times k}$	item latent matrix
$X \in \mathbb{R}^{t \times k}$	ranking scores under categories
$x_c \in X$	ranking score vector under category $c$
$y_{*,c}^v \in Y^v$	$c$ -th column of $Y^v$
$L \in \mathbb{R}^{t \times k}$	ranking lists under categories
$\rho \in \mathbb{R}^k$	counters of category popularity
$e \in \{u, v\}$	entity
$A^e$	content feature of entities
$W^e$	mapping matrix
$Y^e$	entity latent matrix

# Pairwise Preference Assumption

			$v_i$	$v_j$			
$u$	?	?	1	?	?	1	?

user  $u$  prefers item  $v_i$  over  $v_j$

- define the pairwise preference of user  $u$  as:

$$p(i \succ_u j) := f(x_{uij}), \quad (1)$$

where

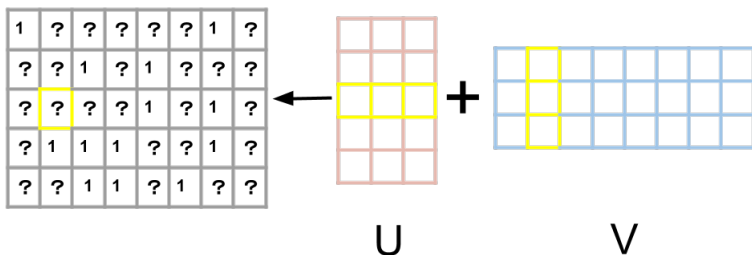
$$f(x) = 1 / (1 + \exp(-x)),$$

$$x_{uij} := \hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}.$$

# Prediction Rule

- The predicted rating  $\hat{r}_{ui}$  of user  $u$  on item  $i$  :

$$\hat{r}_{ui} = U_u \cdot V_i^T + b_i \quad (2)$$





# Likelihood of Pairwise Preference

- The random variable  $x$  with Bernoulli distribution :

$$\text{Ber}(x|p) = p^x (1-p)^{1-x} \quad \text{for } x \in \{0, 1\}, p \in [0, 1] \quad (3)$$

- The Bernoulli distribution of binary random variable  $x((u, i) \succ (u, j))$  is defined as follows :

$$\begin{aligned} LPP_u &= \prod_{i, j \in \mathcal{I}} p(\hat{r}_{ui} > \hat{r}_{uj})^{x((u, i) \succ (u, j))} [1 - p(\hat{r}_{ui} > \hat{r}_{uj})]^{1-x((u, i) \succ (u, j))} \\ &= \prod_{(u, i) \succ (u, j)} p(\hat{r}_{ui} > \hat{r}_{uj}) \prod_{(u, i) \preceq (u, j)} [1 - p(\hat{r}_{ui} > \hat{r}_{uj})] \end{aligned} \quad (4)$$

where  $(u, i) \succ (u, j)$  means that user  $u$  prefers item  $i$  to item  $j$ .

# Ojective Function

- Given a set of pairwise preference  $D_S$ , the goal of BPR is to maximize the likelihood of all pairwise preference:

$$\arg \max_{\Theta} \prod_{(u,i,j) \in D_S} p(i \succ_u j), \quad (5)$$

which is equivalent to minimize the negative log likelihood:

$$L_{feedback} = - \sum_{(u,i,j) \in D_S} \ln f(\hat{r}_{uij}) + \lambda \|\Theta\|^2, \quad (6)$$

where  $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$ ,  $\Theta$  denotes the set of all latent vectors and  $\lambda$  is a hyper-parameter

# Objective Function

- Specifically, Eq(6) is to minimize the following objective function :

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u} \sum_{j \in \mathcal{I} \setminus \mathcal{I}_u} \Phi_{uij} \quad (7)$$

where  $\Phi_{uij} = -\ln f(\hat{r}_{uij}) + \frac{\alpha_u}{2} \|U_{u\cdot}\|^2 + \frac{\alpha_v}{2} \|V_{i\cdot}\|^2 + \frac{\alpha_v}{2} \|V_{j\cdot}\|^2 + \frac{\beta_v}{2} \|b_i\|^2 + \frac{\beta_v}{2} \|b_j\|^2$ ,  $\Theta = \{U_{u\cdot}, V_{i\cdot}, b_i\}$  denotes the parameters to learn.

## SGD

- For a randomly sampled triple  $(u, i, j)$ , calculate the partial derivative for  $U_u$ :

$$\begin{aligned}
 \nabla U_u &= \frac{\partial \Phi_{uij}}{\partial U_u} = -\frac{\partial \ln f(\hat{r}_{uij})}{\partial f(\hat{r}_{uij})} \frac{\partial f(\hat{r}_{uij})}{\partial \hat{r}_{uij}} \frac{\partial \hat{r}_{uij}}{\partial U_u} + \alpha_u U_u. \\
 &= -\frac{1}{f(\hat{r}_{uij})} \frac{\partial f(\hat{r}_{uij})}{\partial \hat{r}_{uij}} \frac{\partial \hat{r}_{uij}}{\partial U_u} + \alpha_u U_u. \\
 &= -\frac{1}{f(\hat{r}_{uij})} f(\hat{r}_{uij}) f(-\hat{r}_{uij}) \frac{\partial f(\hat{r}_{ui} - \hat{r}_{uj})}{\partial U_u} + \alpha_u U_u. \\
 &= -f(-\hat{r}_{uij}) \frac{\partial f[(U_u \cdot V_i^T + b_i) - (U_u \cdot V_j^T + b_j)]}{\partial U_u} + \alpha_u U_u. \\
 &= -f(-\hat{r}_{uij}) (V_i - V_j) + \alpha_u U_u.
 \end{aligned} \tag{8}$$

## SGD

- For the rest of parameters, we have the partial derivatives:

$$\nabla V_{i.} = \frac{\partial \Phi_{uij}}{\partial V_{i.}} = -f(-\hat{r}_{uij}) U_{u.} + \alpha_v V_{i.} \quad (9)$$

$$\nabla V_{j.} = \frac{\partial \Phi_{uij}}{\partial V_{j.}} = -f(-\hat{r}_{uij}) (-U_{u.}) + \alpha_v V_{j.} \quad (10)$$

$$\nabla b_i = \frac{\partial \Phi_{uij}}{\partial b_i} = -f(-\hat{r}_{uij}) + \beta_v b_i \quad (11)$$

$$\nabla b_j = \frac{\partial \Phi_{uij}}{\partial b_j} = -f(-\hat{r}_{uij}) (-1) + \beta_v b_j \quad (12)$$

where  $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$ .

# Update Rules

- For a randomly sampled triple  $(u, i, j)$ , we have the update rules,

$$U_{u\cdot} = U_{u\cdot} - \gamma \nabla U_{u\cdot} \quad (13)$$

$$V_{i\cdot} = V_{i\cdot} - \gamma \nabla V_{i\cdot} \quad (14)$$

$$V_{j\cdot} = V_{i\cdot} - \gamma \nabla V_{j\cdot} \quad (15)$$

$$b_{i\cdot} = b_{i\cdot} - \gamma \nabla b_{i\cdot} \quad (16)$$

$$b_{j\cdot} = b_{j\cdot} - \gamma \nabla b_{j\cdot} \quad (17)$$

where  $\gamma$  is the learning rate.

# The SGD algorithm for BPR

---

**Algorithm 1:** The SGD algorithm for BPR

---

```
1 initialize the model parameter  $\Theta$ ;  
2 for  $t_1 = 1, \dots, T$  do  
3   for  $t_2 = 1, \dots, |\mathcal{P}|$  do  
4     Randomly pick up a pair  $(u, v_i) \in \mathcal{P}$ ;  
5     Randomly pick up an item  $v_j$  from  $\mathcal{I} \setminus \mathcal{I}_u^+$ ;  
6     Calculate the gradients via Eq.(8-12);  
7     Update the model parameters via Eq.(13-17);  
8   end  
9 end
```

---

# Discussion about Randomly Sampling

- For a given training sample  $(u, i, j) \in D_s$ , the stochastic gradient of an arbitrary parameter  $\theta \in \Theta$  is:

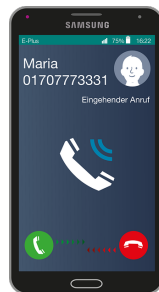
$$\frac{\partial L_{feedback}}{\partial \theta} = -f(-r_{uij}) \frac{\partial (r_{uij})}{\partial \theta} = (f(r_{uij}) - 1) \frac{\partial (r_{uij})}{\partial \theta} \quad (18)$$

- The massive training samples are inefficient to SGD.





# how to select a reasonable negative item $v_j$ ?



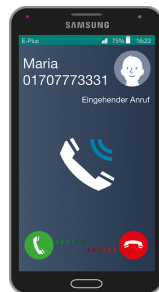
## First step

infer the event that user  $u_m$  selected item  $v_i$  happens on **which category** by the categorical distributions.

## Second step

select an item  $v_j$  with **a high probability to be browsed** by user  $u_m$  under the selected category.

# how to select a reasonable negative item $v_j$ ?



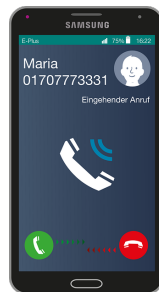
## First step

infer the event that user  $u_m$  selected item  $v_i$  happens on **which category** by the categorical distributions.

## Second step

select an item  $v_j$  with a **high probability to be browsed** by user  $u_m$  under the selected category.

# how to select a reasonable negative item $v_j$ ?



## First step

infer the event that user  $u_m$  selected item  $v_i$  happens on **which category** by the categorical distributions.

## Second step

select an item  $v_j$  with **a high probability to be browsed** by user  $u_m$  under the selected category.

# Categorical Distribution

- The probability that the entity  $e_i$  belongs to the category  $c \in C$ :

$$p(c|e_i) \propto \exp\left(\frac{y_{i,c}^e - \mu_c}{\sigma_c}\right) \quad (19)$$

where  $\mu_c = E(y_{*,c}^e)$  and  $\sigma_c = Var(y_{*,c}^e)$  denote the empirical mean and variance over all entity factors, respectively.

# Categorical Distribution

- It is assumed that the categorical distributions of users and items are independent.
- Then, the probability of observed user-item pair  $(u_m, v_i)$  associating with the category  $c$  could be derived to be a joint probability:

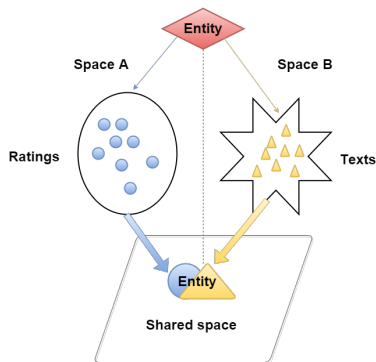
$$p(c|u_m, v_i) = p(c|u_m) p(c|v_i) \quad (20)$$

# Rank-Invariant of Item List

- We adopt Geometric distribution to draw the item  $v_j$  from the ranking list of the category  $c$  :

$$p(v_j|c) \propto \exp(-r(j)/\lambda), \lambda \in \mathbb{R}^+ \quad (21)$$

where  $r(j)$  denotes the ranking place of the item  $v_j$  ,  $\lambda$  is a hyper-parameter which tunes the probability density.



- Based on the study of subspace learning, we can initialize the ranking lists according to content information of items.

## Select a popular category

- According to Eq(20), user-item pairs could be arranged into categories.
- We further count the number of **observed user-item pairs** under each category, and update the category popularity indicator  $\rho$ .



# Update the popular category

- In each iteration, we first sample a popular category  $c$  according to its popularity:

$$p(c|\rho) \propto \exp\left(\frac{\rho_c - \mu}{\sigma}\right) \quad (22)$$

where  $\mu$  and  $\sigma$  denote the empirical mean and variance over the variable  $\rho$ , respectively.

- If the change of ranking score vector under category  $c$  is over given threshold  $\delta$ , we update  $x_c$  by  $y_{*,c}^v$ .

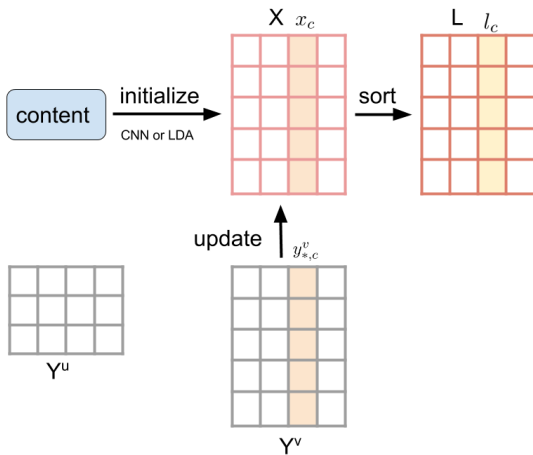


Figure: Adaptive sampling algorithm

# Adaptive sampling algorithm

---

## Algorithm 2: Content-aware and Adaptive sampling

---

- 1 Draw a popular category  $c$  from  $p(c|\rho)$ ;
  - 2 **if**  $\text{sim}(x_c, y_{*,c}^v) > \delta$  **then**
  - 3     Update  $x_c$  by  $y_{*,c}^v$ ;
  - 4     Reorder items under  $c$  and update  $l_c$ ;
  - 5 **end**
  - 6 Draw  $(u_m, v_i) \in \mathcal{P}$  uniformly;
  - 7 Draw a category  $c$  from  $p(c|u_m, v_i), (1 \leq c \leq k)$ ;
  - 8  $\rho_c \leftarrow \rho_c + 1$ ;
  - 9 Draw a rank  $r$  from  $p(r) \propto \exp(-r/\lambda), (1 \leq c \leq k)$ ;
  - 10  $v_j \leftarrow \begin{cases} \text{index}(c, r) & \text{if } \text{sgn}(y_{m,c}^u) = 1; \\ \text{index}(c, n - r - 1) & \text{else} \end{cases}$ ;
-

# Learning content-aware mappings

- We present the objective function to learn the content-aware mappings:

$$L_{content} = \|A^e W^e - Y^e\|_F^2 \quad (23)$$

where the matrix  $A^e = [a_1^e, a_2^e, a_3^e, \dots]$  denotes the content features of entities,  $W^e \in \mathbb{R}^{d^e \times k}$  denotes a mapping matrix, and  $k$  is the dimension of latent vectors.

# Parameter inference of CA-BPR

- The overall objective function of CA-BPR with latent vectors and content-aware mappings is expressed as:

$$\begin{aligned}
 \arg \min_{\Theta, W} L_{feedback} + L_{content} = & - \sum_{(m,i,j) \in D_s} \ln f(r_{mij}) + \lambda \|\theta\|^2 \\
 & + \|A^e W^e - Y^e\|_F^2 + \frac{1}{2} \sum_{e \in \{u,v\}} \lambda^e \|W^e\|_F^2
 \end{aligned} \tag{24}$$

- Given a latent factor matrix  $Y^e$ , we view  $Y^e$  as pseudo labels and treat  $L_{feedback}$  as a constant. Thus, the derivative of objective is

$$\frac{\partial L}{\partial W^e} = (A^e)^T (A^e W^e - Y^e) + \lambda^e W^e \quad (25)$$

Let  $\frac{\partial L}{\partial W^e} = 0$ , the updating rule for  $W^e$  can be derived as:

$$W^e = \left( (A^e)^T A^e + \lambda^e \mathbb{E} \right) A^e Y^e \quad (26)$$

where  $\mathbb{E} \in \mathbb{R}^{k \times k}$  is an identity matrix.

---

**Algorithm 3:** Learning paramters for BPR
 

---

**input :**

The observed user-item pair set  $S$ ;

The feature matrix of items  $F$ ;

The content features entities  $A := \{A^u, A^v\}$ ;

**output:**

$\Theta := \{Y^u, Y^v\}$ ;

$W := \{W^u, W^v\}$ ;

1 initialize the model parameter  $\Theta$  and  $W$  with uniform  $(-\sqrt{6}/k, \sqrt{6}/k)$ ;

2 standarized  $\Theta$ ;

3 Initialize the popularity of categories  $\rho$  randomly;

4 **repeat**

5     Draw a triple  $(m, i, j)$  with Algorithm 2;

6     **for** each latent vector  $\theta \in \Theta$  **do**

7          $\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta}$

8     **end**

9     **for** each  $W^e \in W$  **do**

10         Update  $W^e$  with the rule defined in Eq.26;

11     **end**

12 **until** *convergence*;

---

# Experiment

BPR-MF[Rendle et al., 2009] and CA-BPR[Guo et al., 2015]

Table: Characteristics of compared methods

Method	Content	Sampling
BPR-MF	no	uniform
CA-BPR	yes	non-uniform



# Experiment

**Table:** The performance of approaches by MAP and NDCG.

BPR-MF	k=10	k=20	k=30	k=40	k=50
MAP	0.0879	0.0877	0.1043	0.0888	0.1074
NDCG@3	0.3051	0.3545	0.3398	0.2491	0.3790
NDCG@5	0.3616	0.4296	0.3708	0.2984	0.4153
NDCG@10	0.4120	0.4632	0.4010	0.3163	0.4458
NDCG@20	0.4121	0.4575	0.4164	0.3415	0.4323

CA-BPR	k=10	k=20	k=30	k=40	k=50
MAP	0.1074	0.1072	0.1274	0.1016	0.1229
NDCG@3	0.3790	0.4336	0.4152	0.3044	0.4631
NDCG@5	0.4153	0.4752	0.4531	0.3646	0.5074
NDCG@10	0.4458	0.5101	0.4900	0.3865	0.5447
NDCG@20	0.4323	0.4946	0.5088	0.4173	0.5282

# Experiment

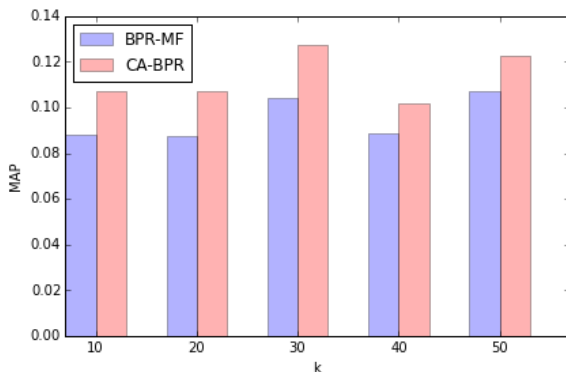


Figure: CA-BPR indeed performs better than BPR-MF.

Thank you !



Guo, W., Wu, S., Wang, L., and Tan, T. (2015).

Adaptive pairwise learning for personalized ranking with content and implicit feedback.

In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2015, Singapore, December 6-9, 2015 - Volume I*, pages 369–376.



Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009).

Bpr: Bayesian personalized ranking from implicit feedback.

In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press.