ESM3061 Data Mining

Recommender System: Collaborative Filtering

Jong-Seok Lee

Dept. of Systems Management Engineering
Sungkyunkwan University

Recommender System

- Systems for recommending items (e.g. books, movies, CD's, web pages) to users based on examples of their preferences

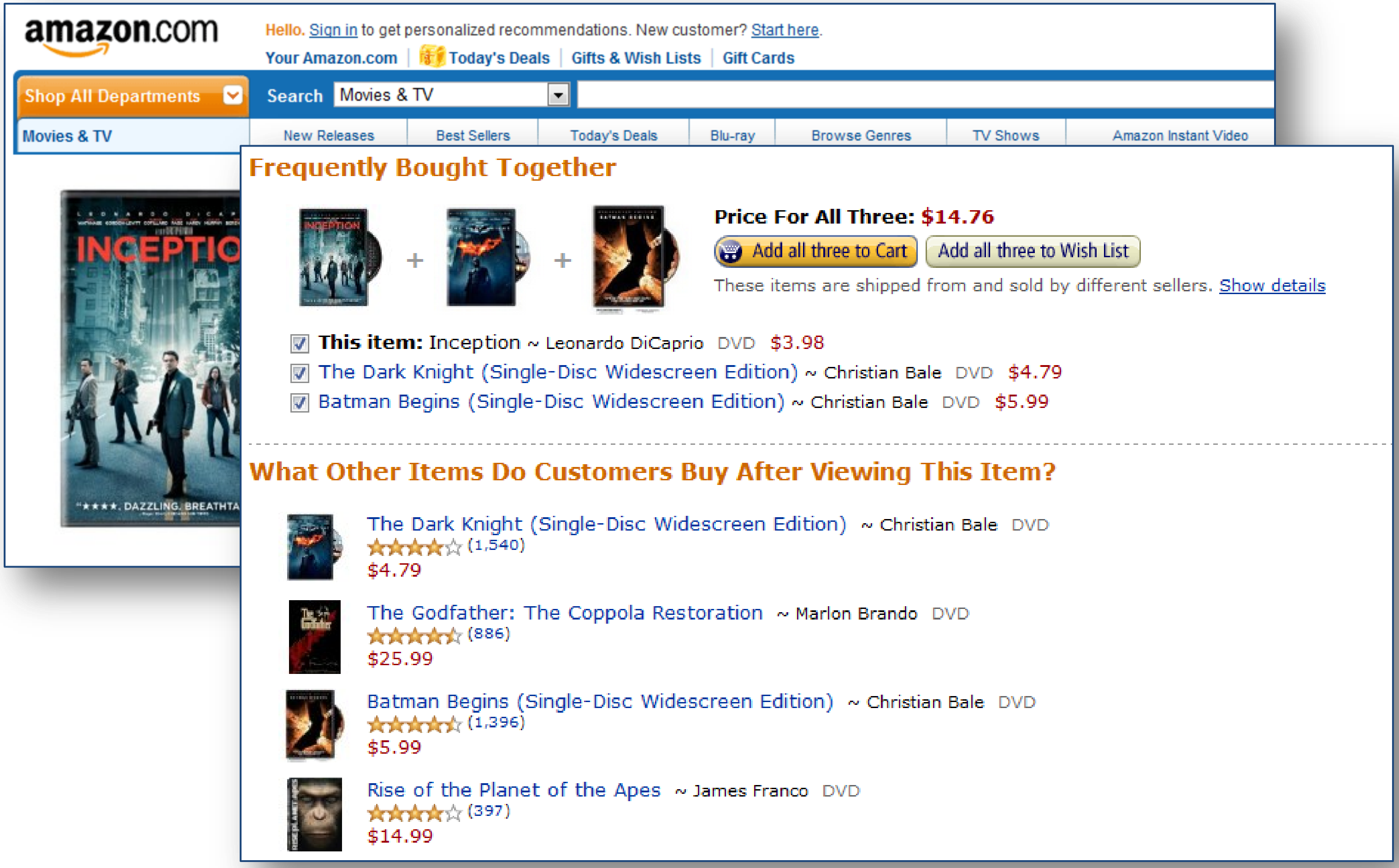
- Many online stores provide recommendations (e.g. Amazon.com)

- Recommenders have been shown to substantially increase sales at online stores

- There is a very often used approach to recommending.
  – Collaborative Filtering

Amazon.com

- According to 2006 sales figures, 35% of Amazon's sales are done through recommendation system.



Personalized Recommender System

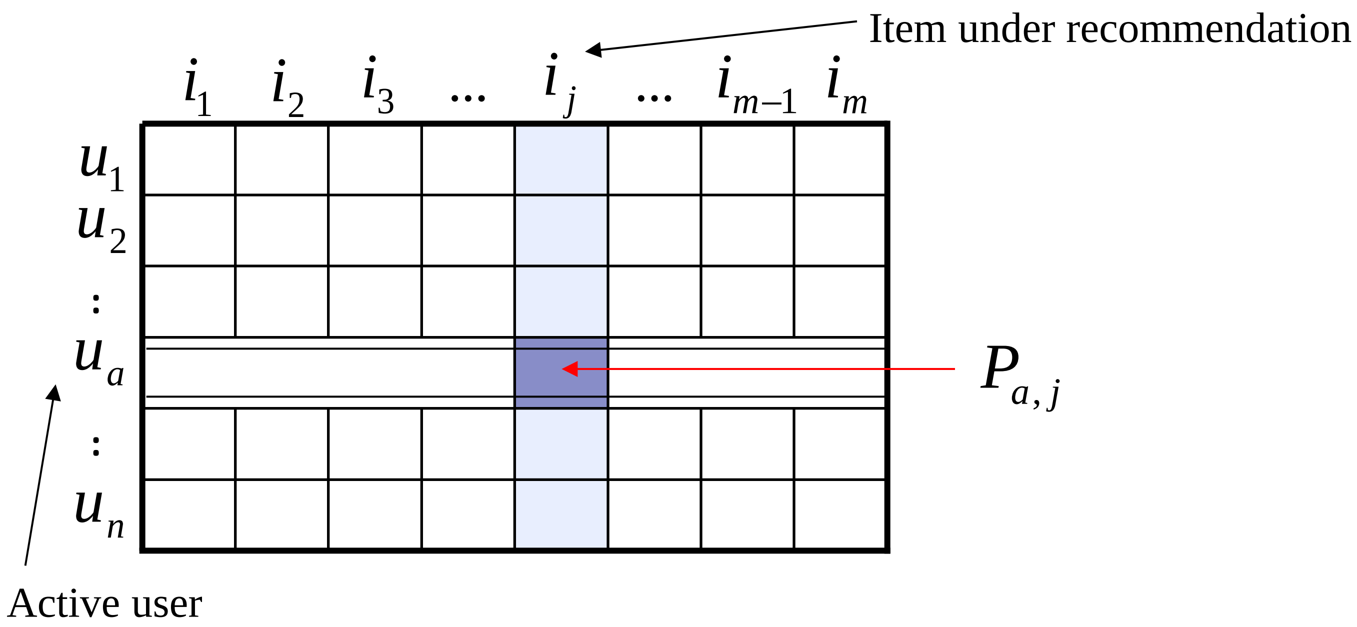
- Technique that uses the known preferences of a group of users to predict the unknown preferences of a new user
- A good way to predict preference is to analyze behavior of people who have similar interests. (Breese, 1998)
- From a business perspective, it is viewed as part of Customer Relationship Management (CRM).

Collaborative Filtering

- Maintain a database of many users' ratings of a variety of items.
- For a given user (called active user), find other similar users whose ratings strongly correlate with the current user.
- Recommend items rated highly by these similar users, but not rated by the current user.
- Almost all existing commercial recommenders use this approach

Data Structure for Collaborative Filtering

- Rating table



User-Based Collaborative Filtering

- r_{ij} : rating of user *i* on item *j*
- $\overline{r}_i$: mean rating of user *i*
- I_i : set of items on which user *i* has rated
- $w(a,i)$: similarity between user *i* and the active user *a*
- P_{aj} : predicted rating of the active user *a* for item *j*
- J : set of items on which user *i* and *a* has co-rated
- S : set of users whose *w(a,i)* can be computed

$$\boxed{P_{aj} = \overline{r}_a + \kappa_a \sum_{i \in S} w(a,i)(r_{ij} - \overline{r}_i)}$$

$$\overline{r}_i = \frac{1}{|I_i|} \sum_{j \in I_i} r_{ij} \qquad w(a,i) = \frac{\sum_{j \in J} (r_{aj} - \overline{r}_a)(r_{ij} - \overline{r}_i)}{\sqrt{\sum_{j \in J} (r_{aj} - \overline{r}_a)^2 \sum_{j \in J} (r_{ij} - \overline{r}_i)^2}} \qquad \kappa_a = \frac{1}{\sum_{i \in S} |w(a,i)|}$$

- The list of top-N items is recommended to the active user.

- A good way to find a certain user's interesting item is to find other users who have a similar taste.

Example

Bad — Good

1 — 5

		SF			Drama			Horror		
		Real Steel	Source Code	Rise of the Apes	Good Will Hunting	The Classic	Love Actually	Rite	Scream 4	Husk
SF Lovers	1	4	5	4		1	1	3	2	
	2	4	4	4				1	1	
	3	5	4		1	2		3	1	
Drama Lovers	4	1	2	1	4	3	5	2	2	2
	5	1	1		3	5	5			
	6		2		3	4	4	1	1	1
Horror Lovers	7	3	3	3	2	1	2	5	4	5
	8	1	2			3	1	4	4	
	9		1			1				5
Active User	10	5	3.87	3.91	1	1.56	1.36	2	1.71	1.73

Similarity Table (Pearson correlation coefficient)

	w(10,1)	w(10,2)	w(10,3)	w(10,4)	w(10,5)	w(10,6)	w(10,7)	w(10,8)	w(10,9)
New user 10	0.66	0.76	0.94	-0.89	-0.81	-0.12	0.05	-0.74	

Item-Based Collaborative Filtering

- r_{ui} : rating of user u on item i (5-star rating scheme is often used.)
- $\overline{r}_i$: mean rating of item i
- U : set of users that have co-rated on item i and j
- $sim(i, j)$: similarity between user i and the active user a
- P_{aj} : predicted rating of the active user a for item j
- U_i : set of users that have rated on item i

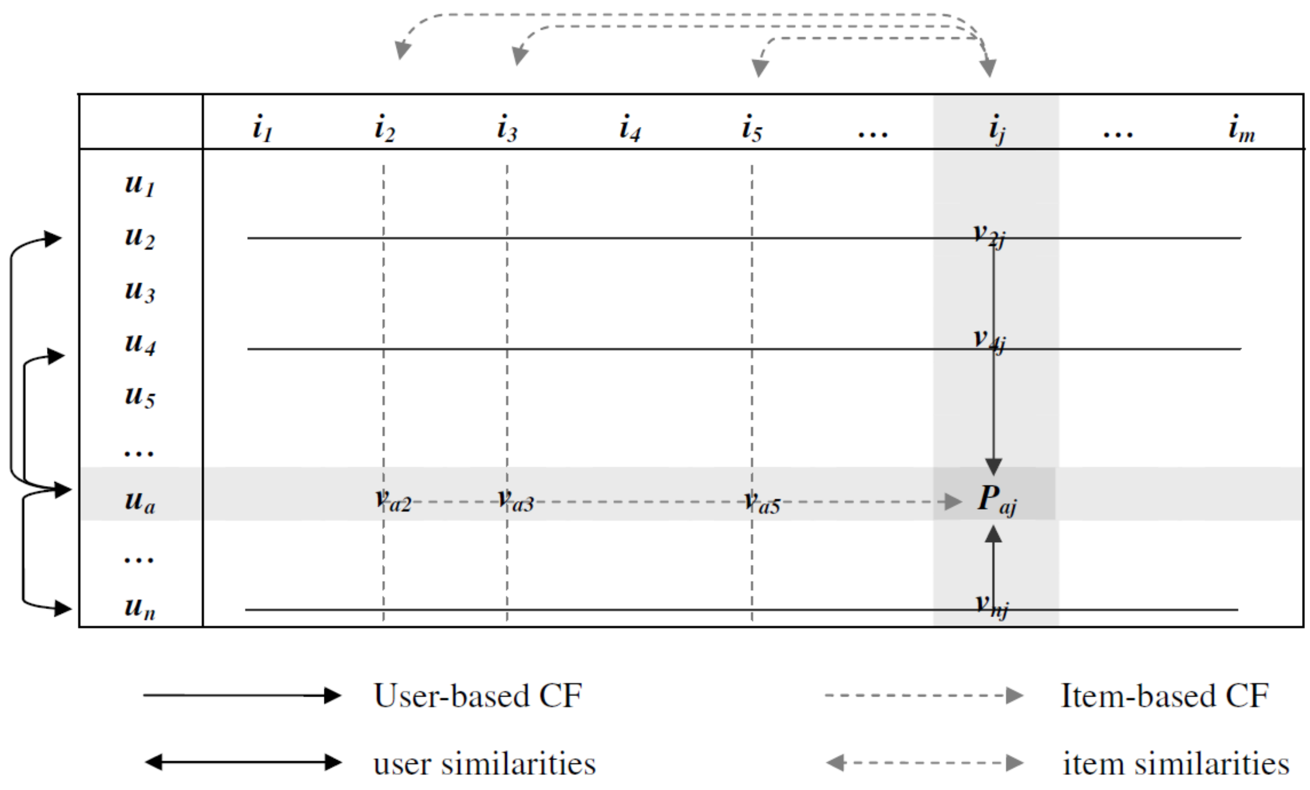
$$\boxed{P_{aj} = \frac{\sum_{i \in I_a} sim(i, j) r_{ai}}{\sum_{i \in I_a} \left| sim(i, j) \right|}} \qquad \overline{r}_i = \frac{1}{|U_i|} \sum_{u \in U_i} r_{ui} \qquad sim(i, j) = \frac{\sum_{u \in U} (r_{ui} - \overline{r}_i)(r_{uj} - \overline{r}_j)}{\sqrt{\sum_{u \in U} (r_{ui} - \overline{r}_i)^2 \sum_{u \in U} (r_{uj} - \overline{r}_j)^2}}$$

- The list of top-N items is recommended to the active user.

- The intuition behind this approach is that a user would be interested in purchasing items that are similar to the items the user liked earlier, and would tend to avoid items that are similar to the items the user didn't like.

Working Direction of Two CFs

- Lee and Olafsson (2009)



Some Issues

- Two challenges
  - Data Sparsity
    - Not enough ratings in database
  - Scalability
    - Computational complexity of *O(n)* where n is the number of users in database

- Two necessary conditions to make a prediction (P_{aj})
  - Minimum number of co-rated cells should be greater than or equal to 2.
  - Variance shouldn't be zero.

Recommendation Using Binary Matrix

- Use of market basket data
  - Less sparse than ratings matrix
- A bit modified formula

$$\boxed{P_{aj} = \kappa_a \sum_{i=1}^{n} w(a,i) r_{ij}}$$

$$r_{ij} = \begin{cases} 0, & \text{no-choice} \\ 1, & \text{choice} \end{cases}$$

$$w(a,i) = \frac{\sum_{j=1}^{m} (r_{aj} - \bar{r}_a)(r_{ij} - \bar{r}_i)}{\sqrt{\sum_{j=1}^{m} (r_{aj} - \bar{r}_a)^2 \sum_{j=1}^{m} (r_{ij} - \bar{r}_i)^2}}$$

	i_1	i_2	i_3	…	i_j	…	i_m
u_1	1	1	0	…	1	…	0
u_2	0	1	0	…	1	…	1
:	:	:	:		:		:
u_a	1	1	1	…	$P_{a,j}$	…	1
:	:	:	:		:		:
u_n	0	0	1	…	1	…	0

  - Other similarity measures for binary variables are possible to compute $w(a,i)$.
    - Simple matching coefficient, Jaccard's coefficient, etc.

Example

	A	B	C	D	E	F	G	H	I	J	K	L
1	1	1	1	0	0	1	0	0	0	0	1	1
2	1	1	1	1	0	0	0	0	0	0	0	0
3	1	0	1	1	0	0	0	1	1	0	0	0
4	1	1	1	1	1	0	0	1	1	0	0	1
5	0	1	1	1	0	0	0	0	0	0	0	0
6	0	1	0	0	1	0	1	1	1	0	0	0
7	1	0	0	0	1	1	1	1	0	0	0	1
8	0	0	0	0	1	1	0	1	0	0	0	0
9	1	0	1	0	1	1	1	1	0	0	0	0
10	0	0	0	1	1	1	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	1	1	1	1
12	1	0	0	0	0	0	0	0	1	1	1	1
13	0	1	0	0	0	1	0	0	1	1	1	0
14	0	0	0	0	0	0	0	0	0	1	1	1
15	0	0	1	1	0	0	0	0	1	1	1	1
16	1	1	1	**0.22**	**-0.39**	**-0.21**	**-0.32**	**-0.21**	**0.04**	1	**0.14**	**0.04**

References

- Breese, J. S., D. Heckerman, and C. Kadie., Empirical analysis of predictive algorithms for collaborative filtering, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, Madison, WI, July 1998.
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., Item-based collaborative filtering recommendation algorithms, *Proceedings of the 10th international world wide web conference*, Hong Kong, May 2001.
- Lee, J.-S. and Olafsson, S., Two-way cooperative prediction for collaborative filtering recommendations, *Expert Systems with Applications*, 36, 5353-5361, 2009.
- Lee, J.-S., Jun, C.-H., Lee, J., and Kim, S., Classification-based collaborative filtering using market basket data, *Expert Systems with Applications*, 29, 700-704, 2005.

SUNG KYUN KWAN UNIVERSITY

Classification-Based Collaborative Filtering Using Market Basket Data

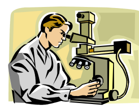
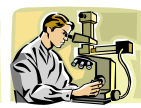
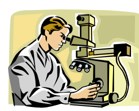
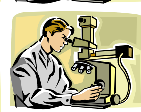
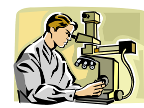
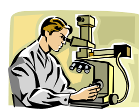
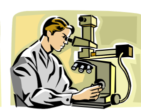
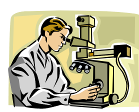
Jong-Seok Lee

Dept. of Systems Management Engineering
Sungkyunkwan University

Two Challenges in CF

- Data Sparsity
  - Not enough ratings in database
- Scalability
  - Computational complexity of $O(n)$ where n is the number of users in database

Research Scope

		Data Set	
		Rating Data	Market Basket Data
Approach	User-based	   	
	Model-based	  	**Proposed Method**

Expected Advantages

		Data Set	
		Voting Data Set	Market Basket Data Set
Approach	User-based	✓ **Sparsity Problem** ✓ **Scalability Problem**	✓ **Scalability Problem**
	Model-based	✓ **Sparsity Problem**	**Free from two main problems**

Classification-Based Collaborative Filtering

m

***jth* item**

1	0	0	1	0	1	1	0	0	0
0	0	1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1
0	0	0	1	1	1	0	1	0	0
0	0	0	1	1	0	1	0	1	1
0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	1	0
0	0	0	0	0	0	1	1	1	0
1	0	0	0	0	0	0	0	0	1
0	0	1	1	0	0	1	1	0	0
1	1	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	0	0
1	1	1	1	0	0	1	1	1	1
0	0	0	1	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1	1
0	0	0	0	1	1	1	1	1	0
0	0	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0	0

n

active user

Modeling

- ✓ Dependent variable : *j*th item
- ✓ Independent variables : the other items
- ✓ We build *m* prediciton models.

$$v_j = f_j(v_1, \cdots, v_{j-1}, v_{j+1}, \cdots, v_m) \qquad j = 1, \cdots, m$$

Recommendation

- ✓ Calculate the probabilities that the items will be chosen for the non-chosen(0) items.
- ✓ The *j*th model is used to calculate the probability that active user will choose the *j*th item.
- ✓ Recommend the N items which have the first Top-N probabilities.

Dimension Reduction

- Principal Component Analysis

		v_t		
		1	0	Total
v_s	1	a	b	a+b
	0	c	d	c+d
	Total	a+c	b+d	n

$$\mathbf{\Sigma}_j = \mathbf{P}\,\mathbf{\Lambda}\,\mathbf{P}'$$

$$\xi_1^j = w_{11}^j v_1 + \cdots + w_{1j-1}^j v_{j-1} + w_{1j+1}^j v_{j+1} + \cdots + w_{1m}^j v_m$$

$$\xi_2^j = w_{21}^j v_1 + \cdots + w_{2j-1}^j v_{j-1} + w_{2j+1}^j v_{j+1} + \cdots + w_{2m}^j v_m$$

$$\vdots$$

$$\xi_p^j = w_{p1}^j v_1 + \cdots + w_{pj-1}^j v_{j-1} + w_{pj+1}^j v_{j+1} + \cdots + w_{pm}^j v_m$$

New independent variables

$$v_j = f_j(\xi_1^j, \xi_2^j, \cdots, \xi_p^j)$$

$$j = 1, \cdots, m$$

$$Var(v_s) = \frac{1}{n}\sum_{i=1}^{n}(v_{si} - \bar{v}_s)^2 = \frac{a+b}{n}\cdot\frac{c+d}{n}$$ Diagonal element of $\mathbf{\Sigma}_j$

$$Cov(v_s, v_t) = \frac{1}{n}\sum_{i=1}^{n}(v_{si} - \bar{v}_s)(v_{ti} - \bar{v}_t) = \frac{a}{n} - \frac{a+b}{n}\cdot\frac{a+c}{n}$$ Off-diagonal element of $\mathbf{\Sigma}_j$

Classification Technique

- Binary Logistic Regression

$$P(v_j = 1) = p^j = \frac{\exp(\boldsymbol{\beta}^j \boldsymbol{\xi}^j)}{1 + \exp(\boldsymbol{\beta}^j \boldsymbol{\xi}^j)}, \quad j = 1, \cdots, m$$

✓ Maximum Likelihood Estimation

$$\underset{\boldsymbol{\beta}^j}{Max} \quad \log(L^j) = \sum_{i=1}^{n} v_{ji} \log\left(\frac{\exp(\boldsymbol{\beta}^j \boldsymbol{\xi}_i^j)}{1 + \exp(\boldsymbol{\beta}^j \boldsymbol{\xi}_i^j)} \right) + \sum_{i=1}^{n} (1 - v_{ji}) \log\left(\frac{1}{1 + \exp(\boldsymbol{\beta}^j \boldsymbol{\xi}_i^j)} \right)$$

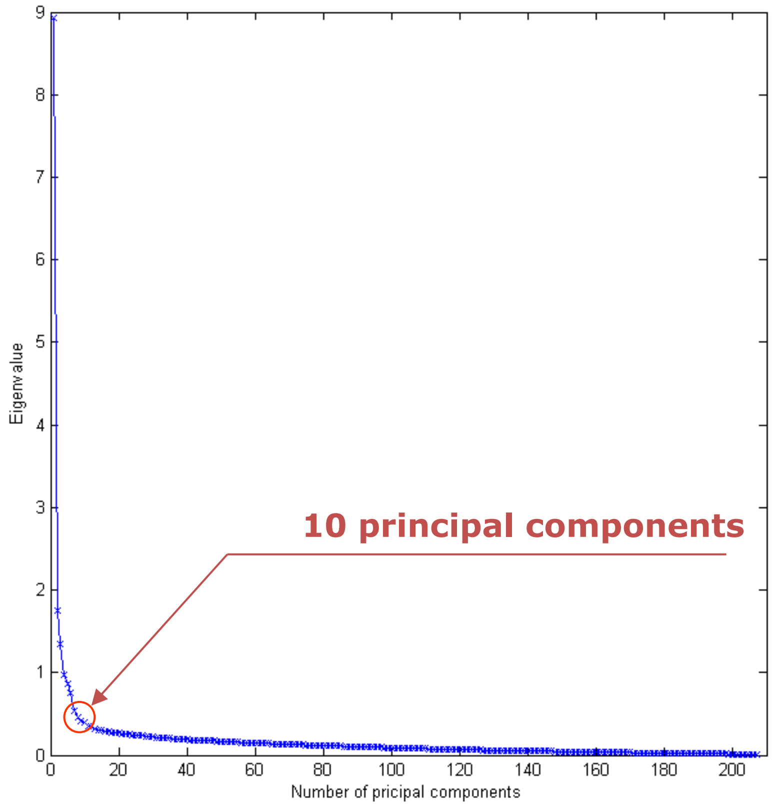
Experiments

	Independent variables	**Dependent variables**
Training set	**604 x 207** **Non-Zero = 33972** **27.17(%)**	**604 x 50** **Non-Zero = 6519** **21.59(%)**
Test set	**121 x 207** **Non-Zero = 9234** **36.87(%)**	**121 x 50** **Non-Zero = 2046** **33.82(%)**

$$\text{Precision} = \frac{\text{hitting number}}{\text{Top - N}}$$

- **We build 50 models to make recommendations.**

Experiments



$\xi_1, \xi_2, \ldots, \xi_{10}$ v_j $j = 1, \cdots, 50$

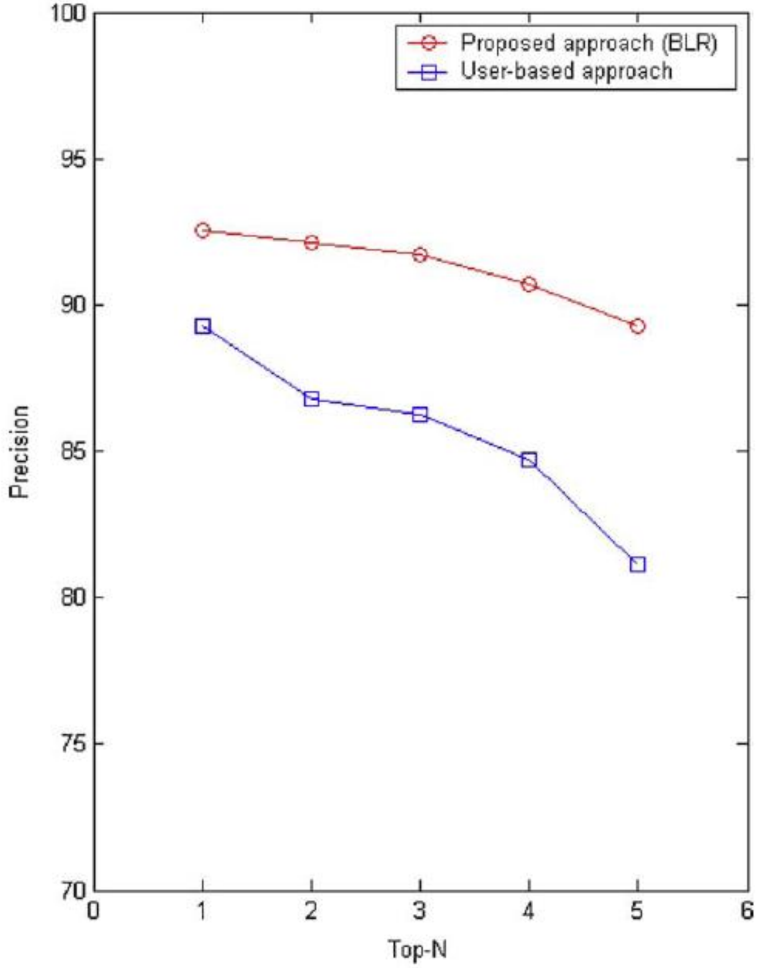
Model Building

Binary Logistic Regression

There are one 207×10 weight matrix and 50 coefficient vector($\boldsymbol{\beta}^j$)s of the models.

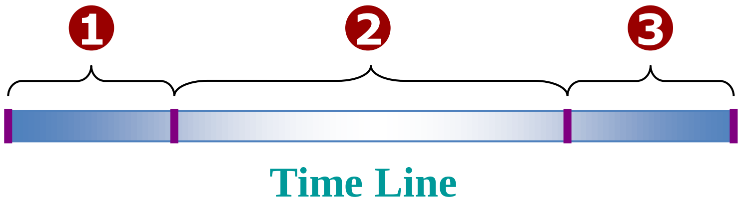
Experiments

- Precision comparison



Experiments

- Prediction time comparison



❶ **Zero-item search time**
❷ **Prediction time**
❸ **Recommendation time**

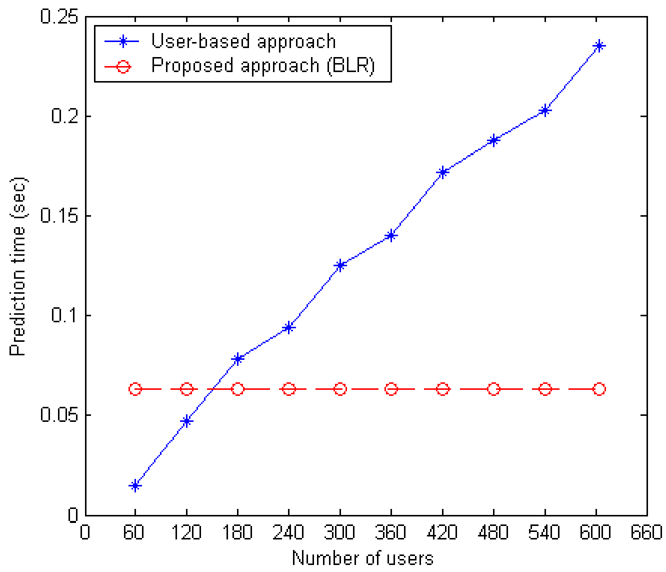
✓ Comparing the prediction time is valuable.

✓ No need to consider modeling (learning) time of the model-based approach.

✓ The figure describes the prediction time of the 50 items for one active user

✓ **Prediction time (BLR) : 0.061 (sec)**

Prediction time comparison with user-based approach



✓ **Classification-based approach is free from scalability problem.**