Big Data Analytics & Applications

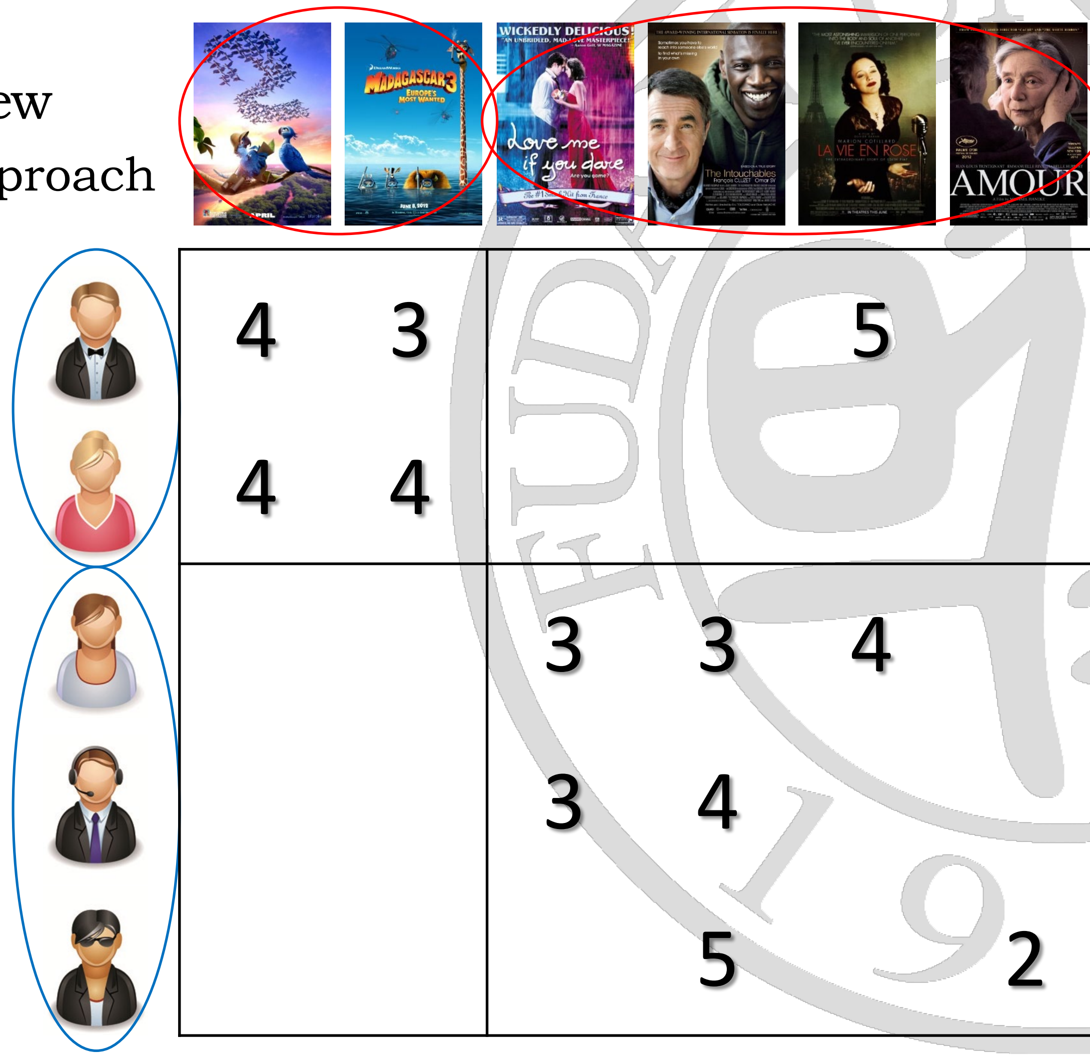
Bin Li

School of Computer Science

Fudan University

Block Models for Network Data

- Block-structure view
  - Co-clustering approach



4	3			5	
4	4				
		3	3	4	
		3	4		
			5		2

Block Models for Network Data

- Co-clustering

1	0
1	0
0	1
0	1
0	1

3.7	5
3.7	3.4

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

Assume two user and two item groups

Matrix tri-factorization: $\hat{\mathbf{X}} = \mathbf{PBQ}^{\mathrm{T}} \in R^{5\times6}$

User membership matrix: $\mathbf{F} \in [0,1]^{5\times2}$

Item membership matrix: $\mathbf{G} \in [0,1]^{6\times2}$

Group-level rating matrix: $\mathbf{B} \in R^{2\times2}$

Block Models for Network Data

- Matrix Reconstruction
  - Predict missing ratings in the preference matrix

4	3			5	
4	4				
3.7		3	3	4	
		3	4	3.4	
			5		2

~

1	0
1	0
0	1
0	1
0	1

3.7	5
3.7	3.4

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

Block Models for Network Data

- Clustering users and items separately
  - Most straightforward way for co-clustering
  - Clustering one side using the other side as features
  - Any clustering algorithm can be applied (e.g., *K*-Means)

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

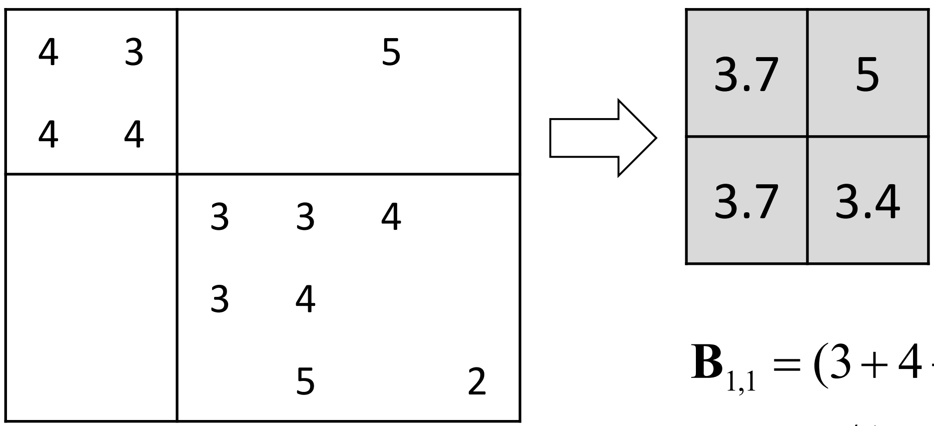
1	0
0	1
0	1
1	0
0	1

4		5			3
	3	4		3	
	3			4	
4					4
			2	5	

4		5			3
	3	4		3	
	3			4	
4					4
			2	5	

Block Models for Network Data

- Group-level rating matrix
  - Each entry is the average rating of a user-item joint group



$$\mathbf{B}_{1,1} = (3+4+4+4)/4 = 3.7$$

$$\mathbf{B}_{1,2} = 5/1 = 5$$

$$\mathbf{B}_{2,1} = (2+3\times 4+4\times 5+5\times 2)/12 = 3.7$$

$$\mathbf{B}_{2,2} = (2+3+3+3+4+4+5)/7 = 3.4$$

Block Models for Network Data

- Flexible Mixture Model[1] (FMM)
  - From hard-membership to soft-membership
  - Each user/item has a distribution over K user/L item groups

$$\hat{\mathbf{X}} = \mathbf{PBQ}^{\mathrm{T}} \text{ where } \mathbf{B}_{k,l} = \sum_r rp(r \mid k,l)$$

$$\text{User } u\text{'s membership in user group } k : \mathbf{P}_{u,k} = p(k \mid u)$$

$$p(k \mid u) \propto p(u \mid k)\, p(k)$$

$$\text{Item } m\text{'s membership in item group } l : \mathbf{Q}_{m,l} = p(l \mid m)$$

$$p(l \mid m) \propto p(m \mid l)\, p(l)$$

[1] Si & Jin: Flexible mixture model for collaborative filtering, ICML 2003

Block Models for Network Data

$\mathrm{E-Step:}$

$$p(k,l\,|\,x_{u,m}) = \frac{p(x_{u,m}\,|\,k,l)\,p(u\,|\,k)\,p(k)\,p(m\,|\,l)\,p(l)}{\sum_{k,l} p(x_{u,m}\,|\,k,l)\,p(u\,|\,k)\,p(k)\,p(m\,|\,l)\,p(l)}$$

$\mathrm{M-Step:}$

$$p(k) = \frac{\sum_l \sum_{w_{u,m}=1} p(k,l\,|\,x_{u,m})}{\sum_{(u,m)} w_{u,m}},\;\; p(l) = \frac{\sum_k \sum_{w_{u,m}=1} p(k,l\,|\,x_{u,m})}{\sum_{(u,m)} w_{u,m}}$$

$$p(u\,|\,k) = \frac{\sum_l \sum_{w_{v,m}=1 \cap v=u} p(k,l\,|\,x_{v,m})}{p(k)\sum_{(u,m)} w_{u,m}},\;\; p(m\,|\,l) = \frac{\sum_k \sum_{w_{u,m'}=1 \cap m'=m} p(k,l\,|\,x_{u,m'})}{p(l)\sum_{(u,m)} w_{u,m}}$$

$$p(r\,|\,k,l) = \frac{\sum_{w_{u,m}=1 \cap x_{u,m}=r} p(k,l\,|\,x_{u,m})}{\sum_{w_{u,m}=1} p(k,l\,|\,x_{u,m})}$$

Cold-Start Problem

- Cold-Start Problem in Collaborative Filtering

New item

4		5			3	
	3	4		3		
	3			4		
4					4	
			2	5		

New user

Cold-Start Problem

- A major limitation of CF
  - A reason that real-world RSs adopts hybrid strategies

- Solutions for user cold-start
  - Demography-based
  - Popularity-based (most popular items)
  - Social relationship based (friends' preference)
  - Implicit preference based (e.g., browsed items)

- Solutions for item cold-start
  - Content-based
  - Ratings borrowed from items of the same category

Temporal Changes

- A major challenge of CF
  - Real RSs usually take into account temporal factors

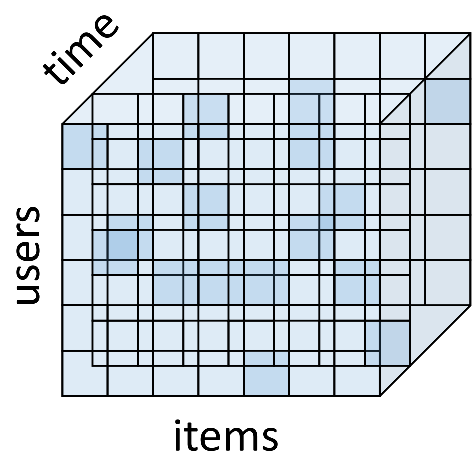
- Causes of temporal changes from users
  - Changing bias
  - Changing interest
  - Changing context

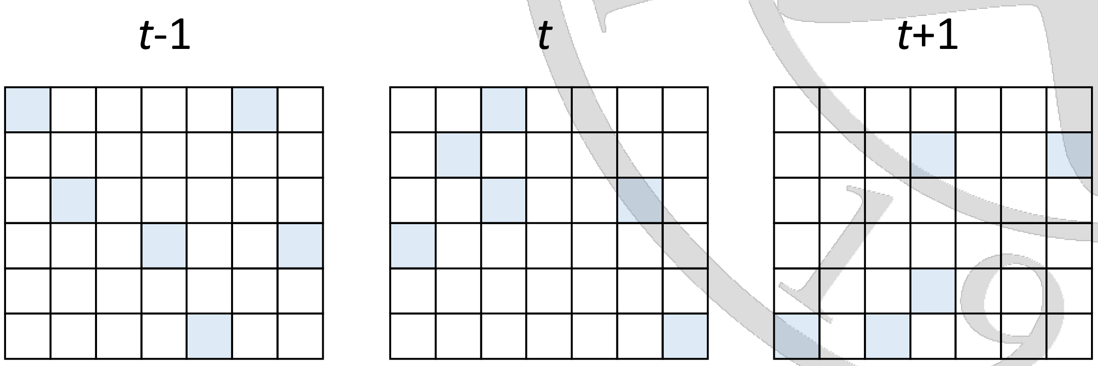
- Causes of temporal changes from items
  - Seasonal effects (Valentine's day, Mid-autumn day)
  - Trending (fashions, digital products)

Temporal CF

- Temporal CF Problem
  - Each timestamp has a rating matrix
  - Can be represented as a tensor

user	movie	date	rate
1	34	11-04-02	3
1	296	09-05-02	4
2	11	18-01-02	5
2	59	23-02-02	4
2	124	03-04-02	2

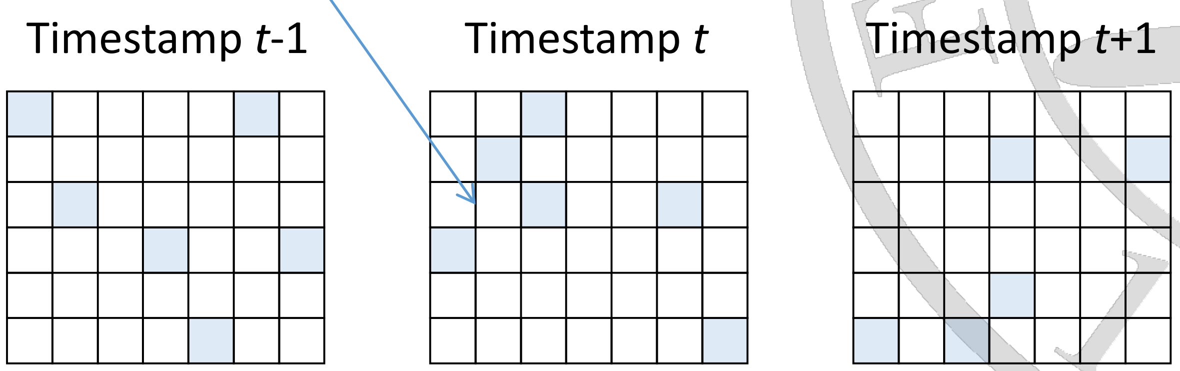




Temporal CF

- TimeSVD++[1]: Netflix Winner’s Method
  - An improvement of SVD++ for temporal CF
  - TimeSVD++ considers time-dependent factors: user rating bias $b_u(t)$, item rating bias $b_m(t)$, and user feature vector $\mathbf{f}_u(t)$

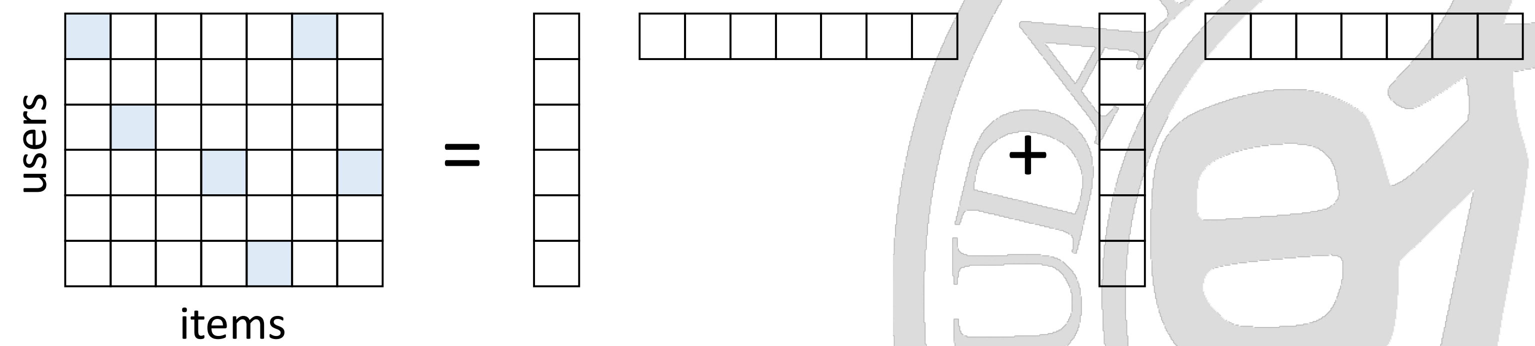
$$x_{u,m,t} = \mu + b_u(t) + b_m(t) + \mathbf{g}_m^{\mathrm{T}} \mathbf{f}_u(t)$$



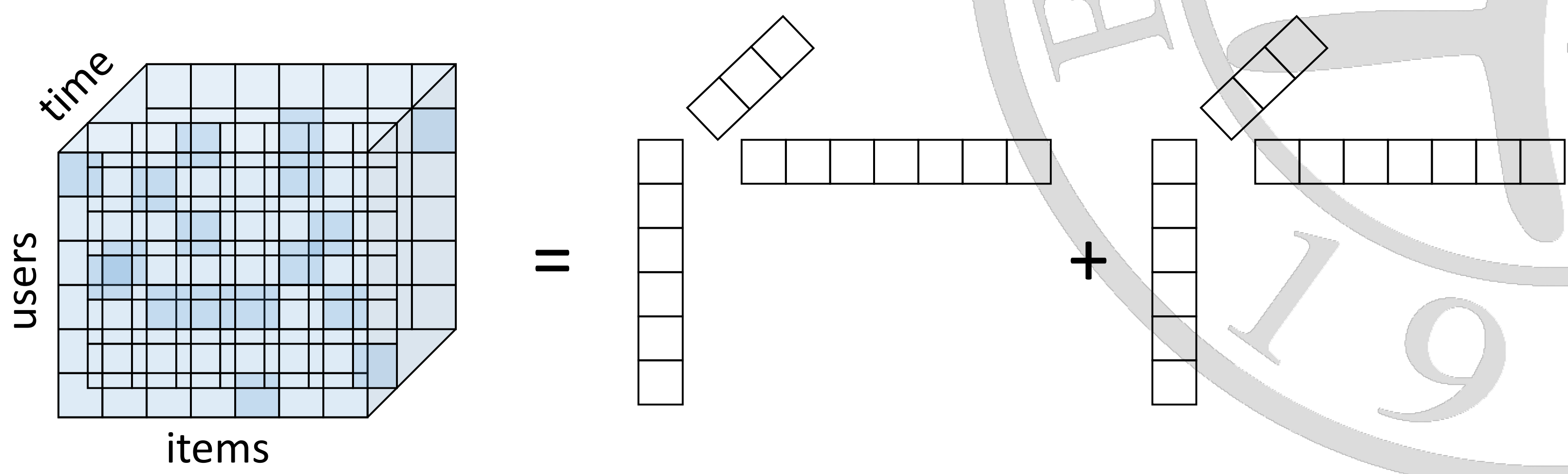
[1] Y. Koren: Collaborative Filtering with Temporal Dynamics, KDD 2009

Temporal CF

- Matrix Factorization



- Tensor Factorization



Noise Problem

- Spammer Detection (malicious users)
  - Promote certain items with misleading information
  - Usually formulate as a classification problem to detect malicious users

- Shilling Detection (malicious users)
  - A group of colluded users inserting untruthful profiles to promote or degrade certain items
  - Fake profiles are usually generated according to certain distributions
  - Usually formulate as a clustering or principal component analysis problem to detect colluded users

Noise Problem

- Natural Noise Detection (nonmalicious users)
  - Difficult to detect because no patterns
  - Difficult to define natural noisy users
  - Difficult to quantify the noise

- Solutions: Consistency of Preference
  - The larger the difference, the more likely a user is to be noisy
  - E.g., consistency between observed and predicted ratings
  - E.g., consistency between multiple ratings on same items

Implicit Feedback

- Implicit Feedback Data
  - Click-through records, purchased records, etc.
  - Easy and cheap to obtain
  - Large amount
  - Noisy

1		1			1
	1	1		1	
	1			1	
1					1
			1	1	

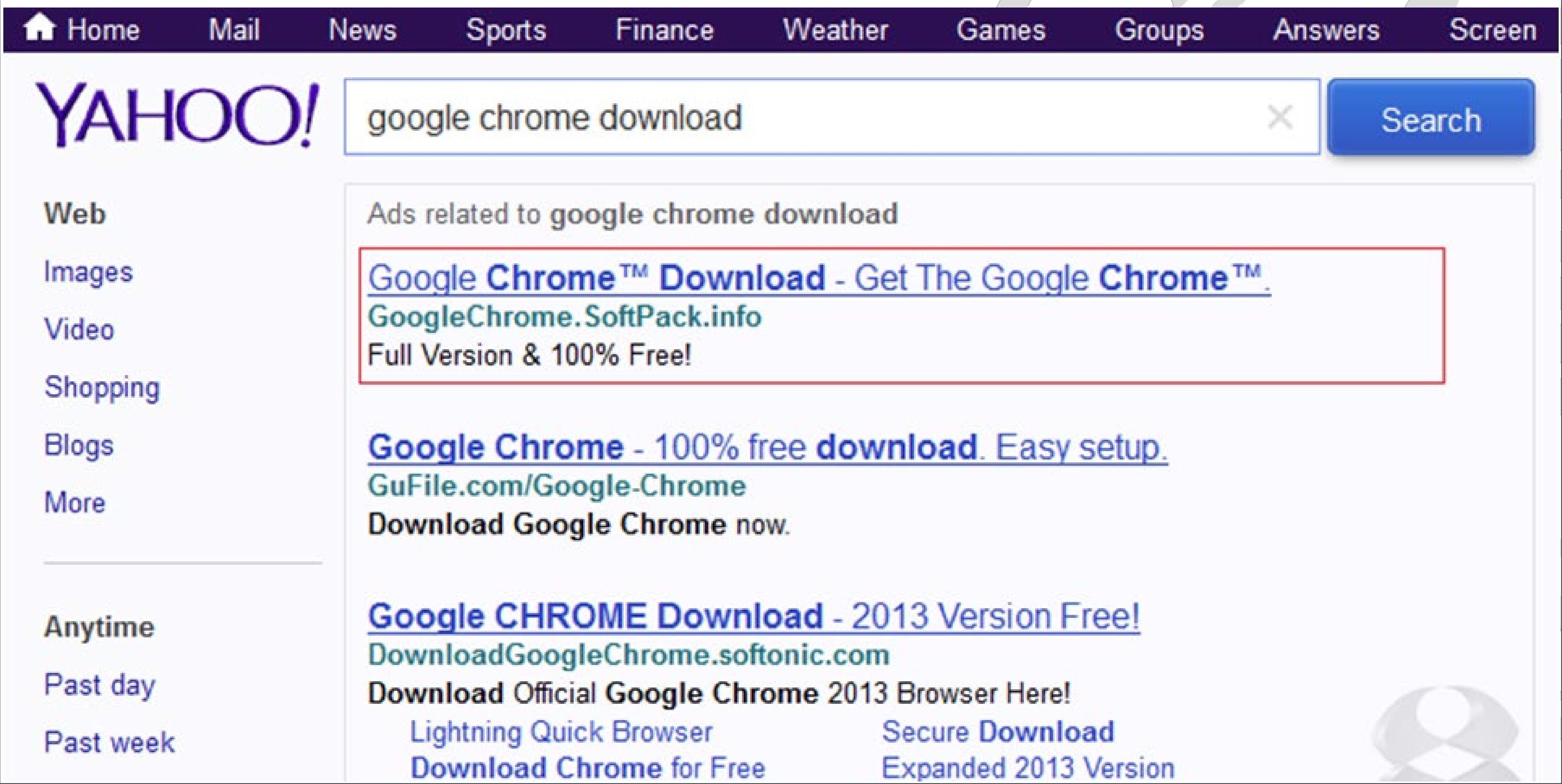
Implicit Feedback

- Characteristics of Implicit Feedbacks
  - Simple (usually binary data)
  - Abundant
  - Noisy
  - Sequential

- A Better Approach – Online Learning
  - Binary data is simpler for online learning
  - Performance can be reinforced using noisy but abundant data
  - Sequential arrived data is natural for online learning

Online Recommendation

- Example: Online Advertising



Online Recommendation

- Basic Idea of Online Recommendation
  - Initial recommending item set (usually popularity-based)
  - Recommend her favorite items based on users' feedback
  - Also try other items potentially extracting the user

- Exploitation and Exploration Problem
  - An online decision making problem
  - "Exploitation" of the items been frequently clicked
  - "Exploration" to get more information about the other items
  - A tradeoff between Exploitation and Exploration

Multi-Armed Bandits

- An ML problem originated from casino
  - Each slot machine has an unknown probability to win
  - The gambler faces a set of slot machines (*K-armed bandits*)
  - Play slot machines sequentially to achieve the largest possible reward

Multi-Armed Bandits

- Arms (Items)
  - K arms to select (i.e., K-armed bandits)
  - Each arm has an unknown (possibly changing) probability of reward
- Rewards (Click-throughs, Purchases)
  - At time t = 1, 2, ⋯, select one arm a_t to play and get a random reward r_t according to the unknown probability
- Trials
  - (a_1, r_1) (a_2, r_2) ⋯ (a_t, r_t) ⋯
- Goal
  - Maximize the accumulated rewards over time

MAB for Online Recommendation

- Bernoulli MAB

  - Prior of item m being clicked is $\mathrm{Beta}(\alpha, \beta)$
  - Online recommendations are Bernoulli trials : a_m (clicked), b_m (unclicked)
  - Posterior of being clicked is $\mathrm{Beta}(\alpha + a_m, \beta + b_m)$
  - Sample from $\mathrm{Beta}(\alpha + a_m, \beta + b_m)$

- Exploitation and Exploration
  - Items been clicked more are more likely to be recommended
  - Other items also have different probabilities to be tried

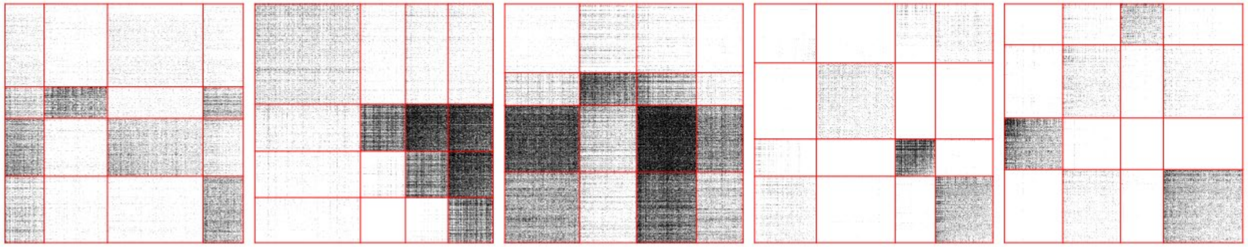
MAB for Online Recommendation

- Thompson sampling for the Bernoulli MAB

- for $t = 1, \ldots, T$ do
- for $m = 1, \ldots, M$ do
- draw θ_m from Beta$(\alpha + a_m, \beta + b_m)$
- end for
- select arm $\hat{m} = \arg\max_m \theta_m$ and observe click r_t
- if $r_t = 1$ then
- $a_k = a_k + 1$
- else
- $b_k = b_k + 1$
- end if
- end for

Project: Block Modeling for Network Data

- Dataset:
  - Public available datasets for collaborative filtering (e.g., https://movielens.org/) or social network analysis (e.g., https://snap.stanford.edu/data/)
- Method:
  - Use Infinite Relational Model for network data analysis: http://web.mit.edu/cocosci/Papers/Kemp-etal-AAAI06.pdf
- Experiments:
  - Visualize the block structures of the modeling results



Thanks

Email: libin@fudan.edu.cn