香港中文大學
The Chinese University of Hong Kong

Learning to Improve Recommender Systems

Guang Ling

Jan 23rd, 2015

Outline

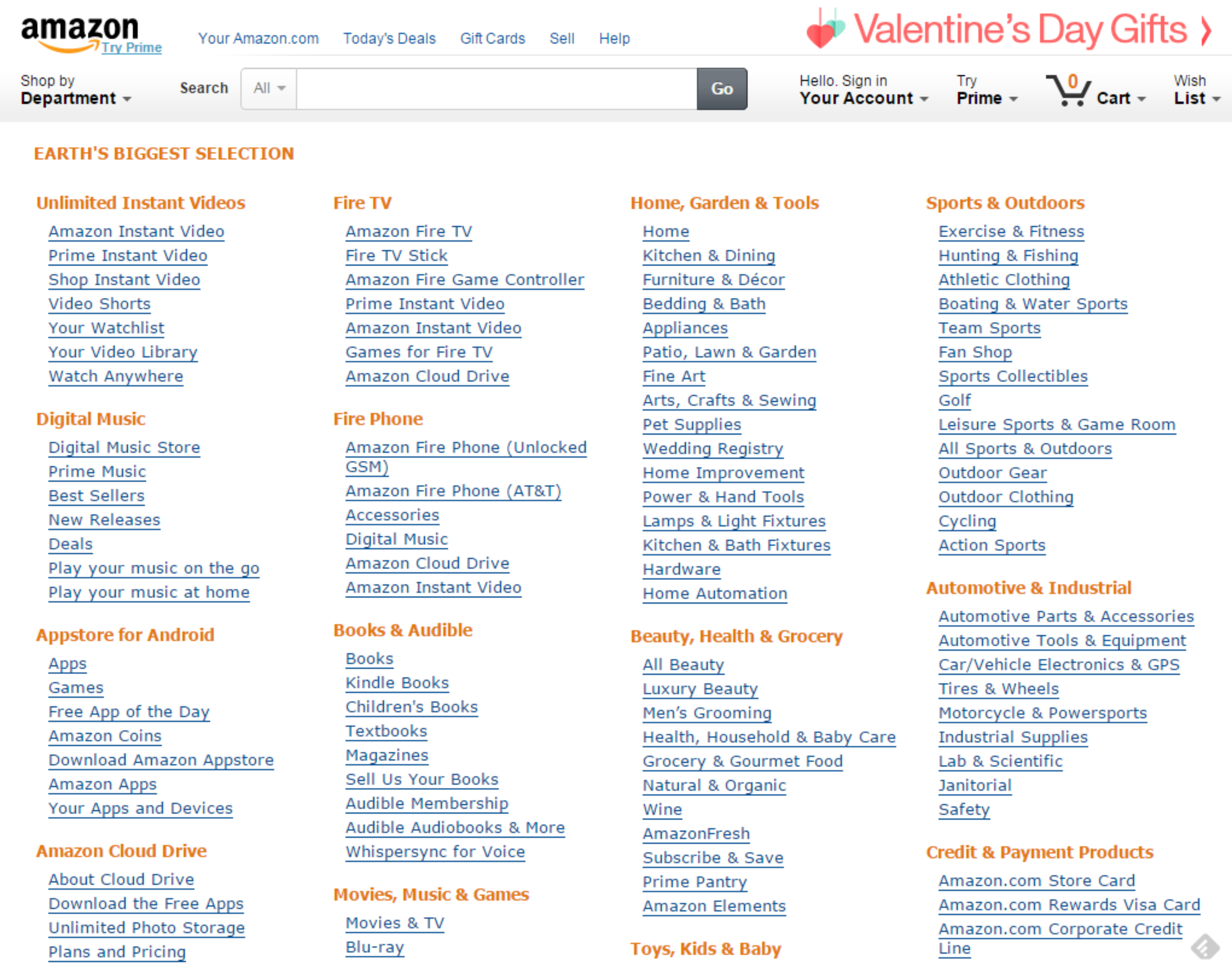
- Introduction and Background Review
- Online Collaborative Filtering
- Response Aware Collaborative Filtering
- User Reputation Estimation
- Combine Ratings with Reviews
- Conclusion

Outline

- Introduction and Background Review
- Online Collaborative Filtering
- Response Aware Collaborative Filtering
- User Reputation Estimation
- Combine Ratings with Reviews
- Conclusion

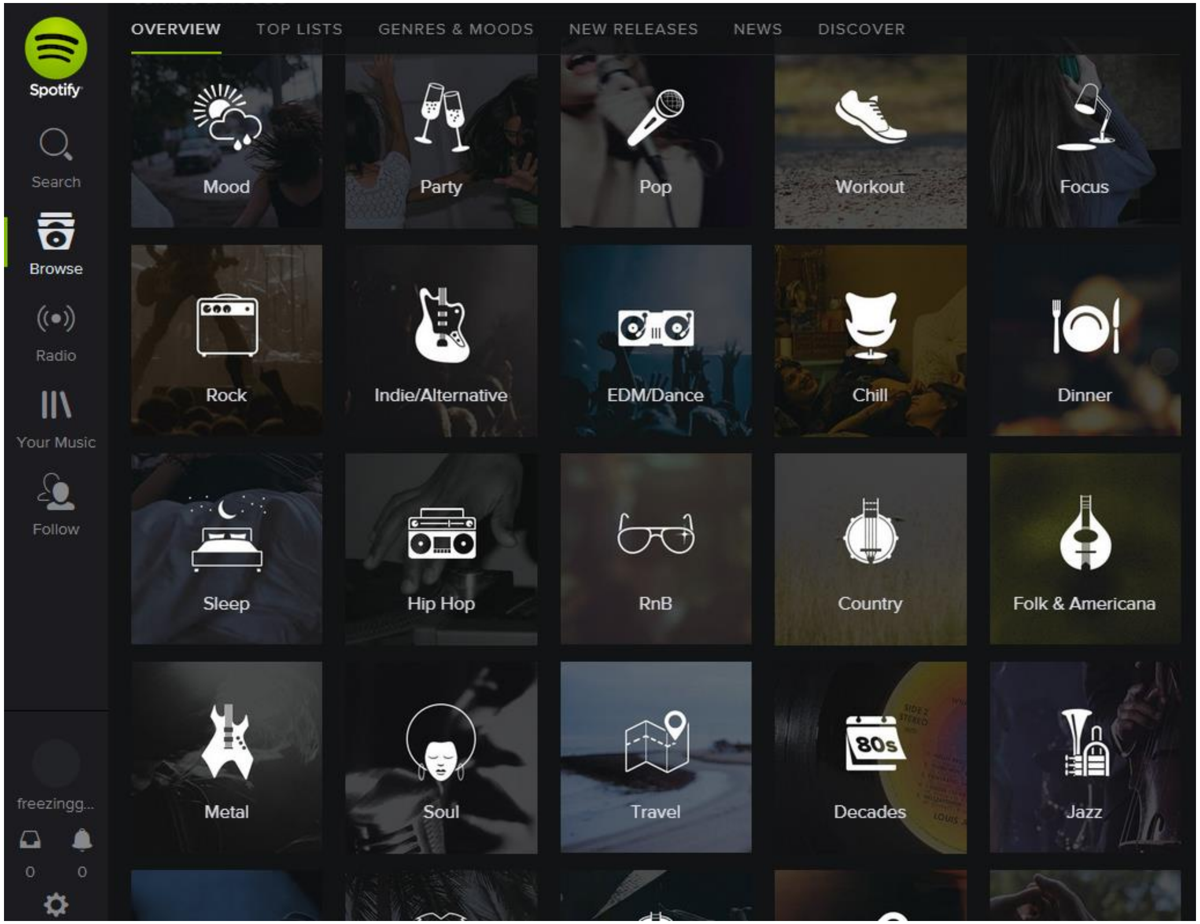
Introduction

Introduction



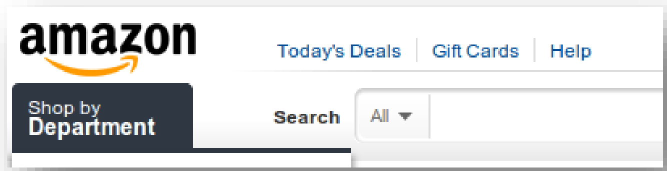
Introduction

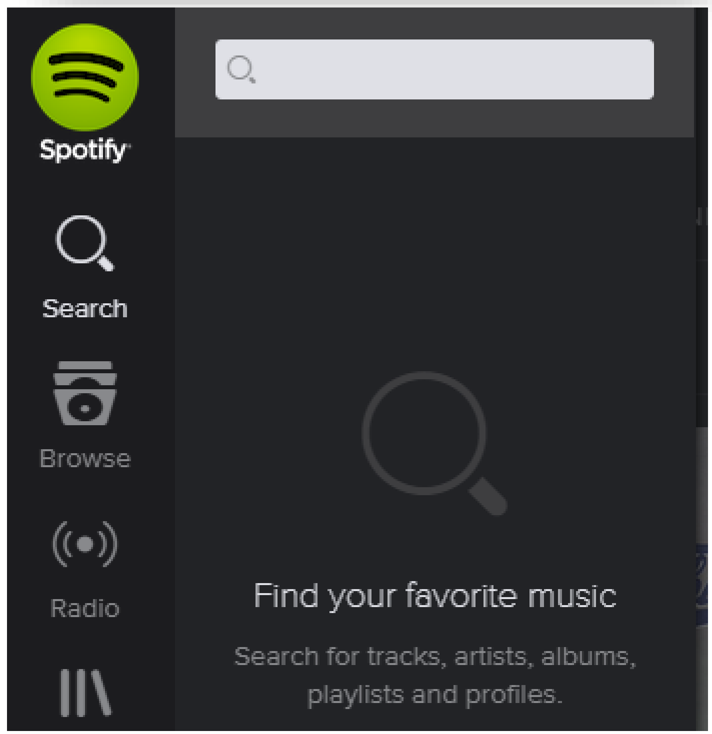
Introduction



Introduction

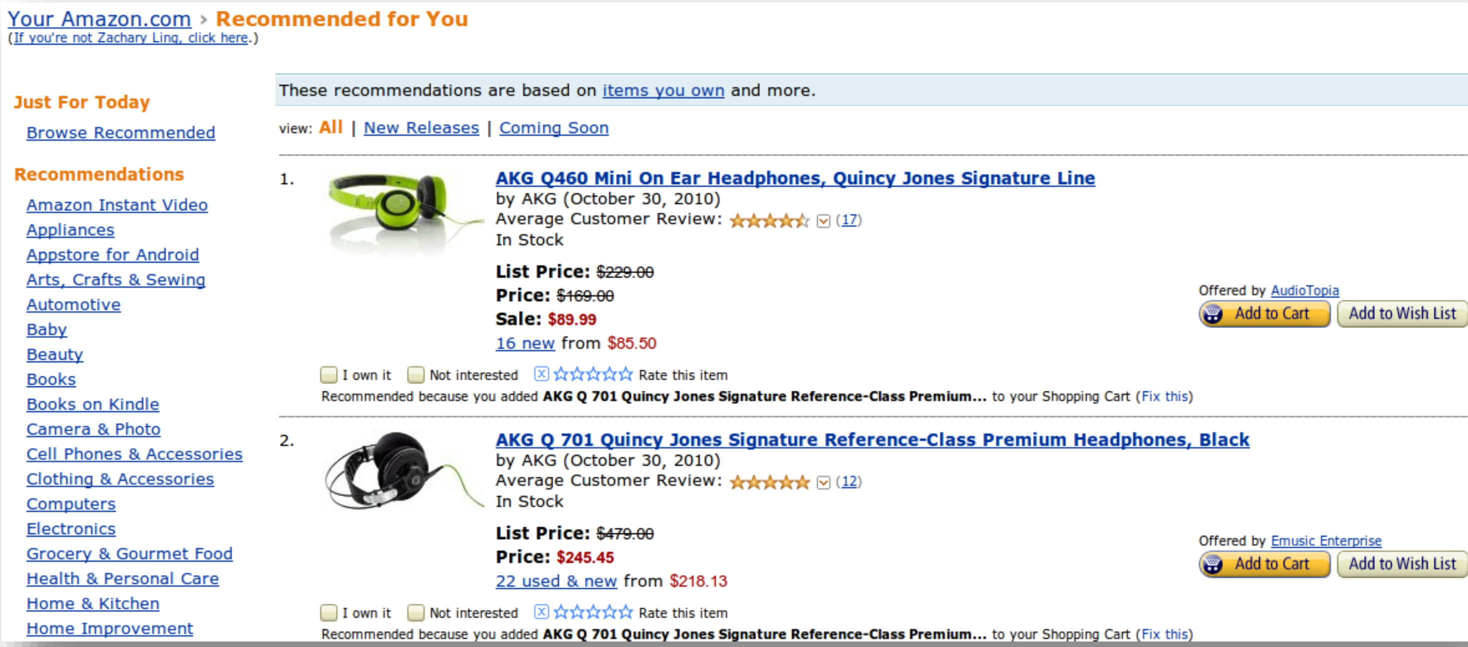
Introduction

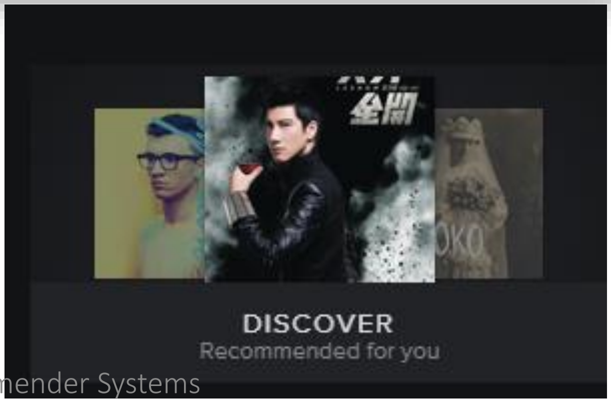




Introduction

Introduction

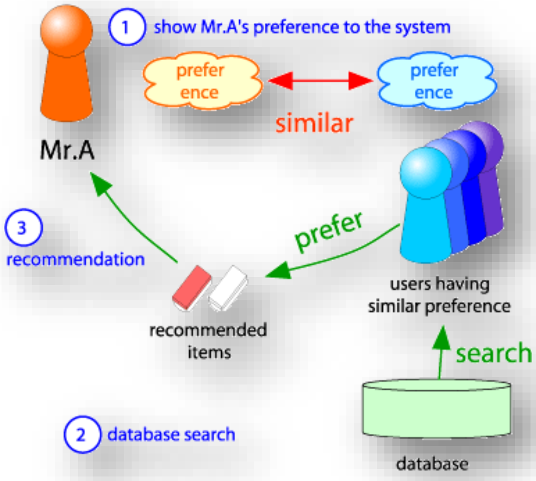




Recommender System Approaches

- Content based filtering
  - Content analyzer
  - Profile learner
  - Filtering component

- Collaborative filtering
  - Utilize other users' ratings to recommend
  - Neighborhood based
  - Model based



Recommender System Approaches

- Content based filtering
  - News recommendation
  - Pros
    - User independent
    - Explainable
    - New items
  - Cons
    - Domain dependent
    - Over-specialization
    - New users
- Collaborative filtering
  - Music movie recommendation
  - Pros
    - Domain independent
    - Discovery new items
    - Accurate
  - Cons
    - New items or users
    - Black box algorithm

Problem Statement

- Given N users' *partial* ratings on M items, collaborative filtering methods try to predict each users' preferences on each item.
- Notations
  - N users $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$, M items $\mathcal{I} = \{i_1, i_2, \cdots, i_M\}$, all items rated by u_i are denoted by $\mathcal{I}_i$, all users who have rated i_j are denoted by $\mathcal{U}_j$
  - Ratings are arranged in a partially observed matrix X, where X_{ij} denote the rating user u_i assigned to i_j
  - Alternatively, the ratings can be arranged in a set of triplets $(u, i, x) \in \mathcal{Q}$

	i1	i2	i3	i4	i5	i6	i7	i8
u1	5	2		3		4		
u2	4	3			5			
u3	4		2				2	4
u4								
u5	5	1	2		4	3		
u6	4	3		2	4		3	5

Problem Statement

- Given N users' *partial* ratings on M items, collaborative filtering methods try to predict each users' preferences on each item.
- Notations
  - N users $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$, M items $\mathcal{I} = \{i_1, i_2, \cdots, i_M\}$, all items rated by u_i are denoted by $\mathcal{I}_i$, all users who have rated i_j are denoted by $\mathcal{U}_j$
  - Ratings are arranged in a partially observed matrix X, where X_{ij} denote the rating user u_i assigned to i_j
  - Alternatively, the ratings can be arranged in a set of triplets $(u, i, x) \in \mathcal{Q}$

Ratings are arranged in a $N \times M$ matrix X

	i1	i2	i3	i4	i5	i6	i7	i8
u1	5	2		3		4		
u2	4	3			5			
u3	4		2				2	4
u4								
u5	5	1	2		4	3		
u6	4	3		2	4		3	5

Problem Statement

- Given N users' *partial* ratings on M items, collaborative filtering methods try to predict each users' preferences on each item.
- Notations
  - N users $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$, M items $\mathcal{I} = \{i_1, i_2, \cdots, i_M\}$, all items rated by u_i are denoted by $\mathcal{I}_i$, all users who have rated i_j are denoted by $\mathcal{U}_j$
  - Ratings are arranged in a partially observed matrix X, where X_{ij} denote the rating user u_i assigned to i_j
  - Alternatively, the ratings can be arranged in a set of triplets $(u, i, x) \in \mathcal{Q}$

Usually, we predict the rating values

	i1	i2	i3	i4	i5	i6	i7	i8
u1	5	2	?	3	?	4	?	?
u2	4	3	?	?	5	?	?	?
u3	4	?	2	?	?	?	2	4
u4	?	?	?	?	?	?	?	?
u5	5	1	2	?	4	3	?	?
u6	4	3	?	2	4	?	3	5

Neighborhood Based Methods

User Based Methods

- Leverage *similar users'* ratings

Item Based Methods

- Leverage *similar items'* ratings

	I1	I2	I3	I4
U1	1	5	4	?
U2	2	5	4	1
U3	4	2	1	4
U4	3	5	1	2
U5	4	3	1	4

Neighborhood Based Methods

User Based Methods

- Leverage *similar users'* ratings

Item Based Methods

- Leverage *similar items'* ratings

	I1	I2	I3	I4
U1	1	5	4	?
U2	2	5	4	1
U3	4	2	1	4
U4	3	5	1	2
U5	4	3	1	4

Neighborhood Based Methods

User Based Methods

- Leverage *similar users'* ratings

Item Based Methods

- Leverage *similar items'* ratings

	I1	I2	I3	I4
U1	1	5	4	?
U2	2	5	4	1
U3	4	2	1	4
U4	3	5	1	2
U5	4	3	1	4

Neighborhood Based Methods

User Based Methods

- Leverage *similar users'* ratings

Item Based Methods

- Leverage *similar items'* ratings

	I1	I2	I3	I4
U1	1	5	4	?
U2	2	5	4	1
U3	4	2	1	4
U4	3	5	1	2
U5	4	3	1	4

Neighborhood Based Methods

User Based Methods

- Leverage *similar users'* ratings

Item Based Methods

- Leverage *similar items'* ratings

	I1	I2	I3	I4
U1	1	5	4	?
U2	2	5	4	1
U3	4	2	1	4
U4	3	5	1	2
U5	4	3	1	4

Neighborhood Based Methods

User Based Methods

- Leverage *similar users'* ratings

Item Based Methods

- Leverage *similar items'* ratings

Neighborhood Based Methods

User Based Methods

- Leverage *similar users'* ratings

Item Based Methods

- Leverage *similar items'* ratings

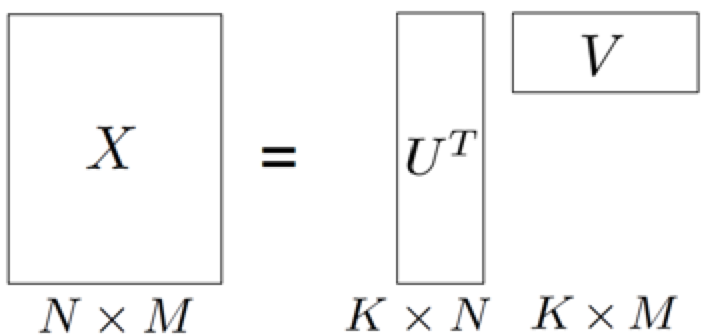
- Pros
  - Simple and easy to implement
  - Clear interpretation
- Cons
  - Manipulate ratings directly lead to high time complexity
  - Prone to sparseness problem

Model Based Methods

- Do not manipulate ratings directly
- Train a predefined compact model
- Usually efficient at prediction time
- Successful methods
  - Probabilistic latent semantic analysis
  - Matrix factorization based methods, etc.

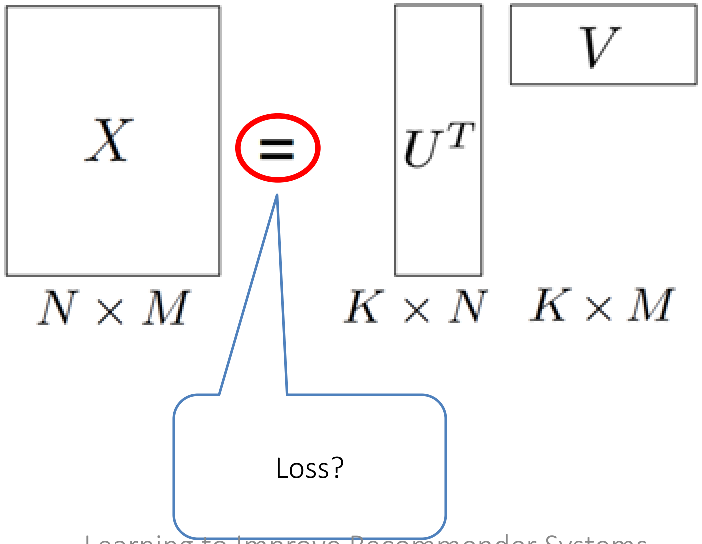
Matrix Factorization Based Methods

- Assumption
  - X has a low-rank structure
  - Users' preferences and items' features can be modeled using a few factors
  - User feature matrix $U \in \mathbb{R}^{K \times N}$
  - Item feature matrix $V \in \mathbb{R}^{K \times M}$



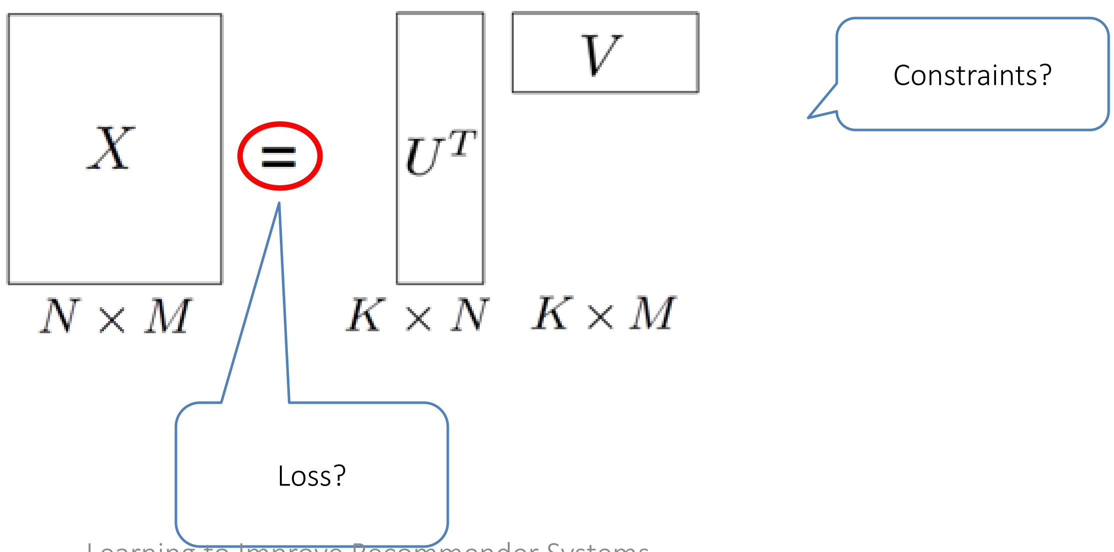
Matrix Factorization Based Methods

- Assumption
  - X has a low-rank structure
  - Users' preferences and items' features can be modeled using a few factors
  - User feature matrix $U \in \mathbb{R}^{K \times N}$
  - Item feature matrix $V \in \mathbb{R}^{K \times M}$



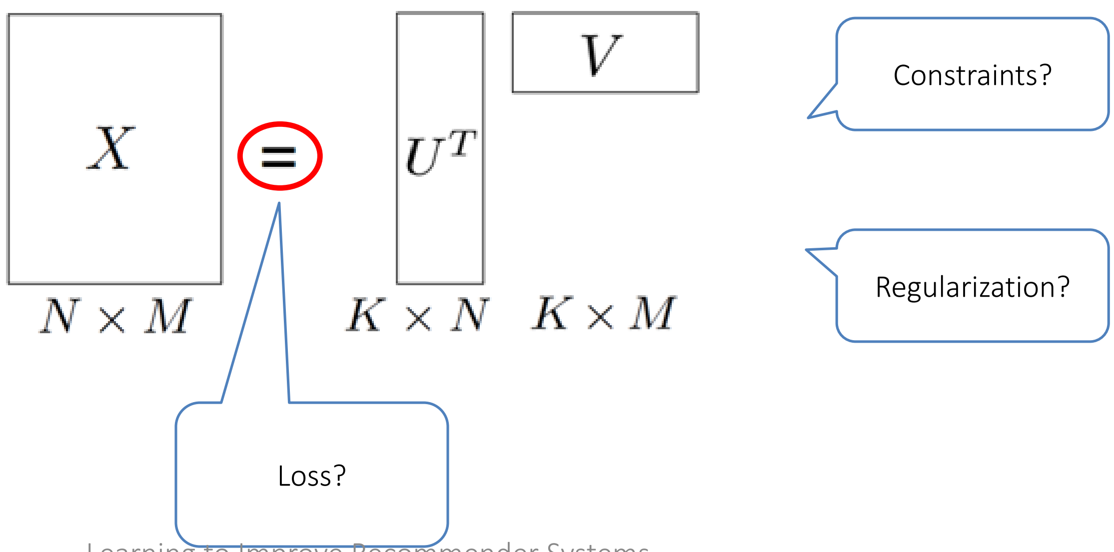
Matrix Factorization Based Methods

- Assumption
  - X has a low-rank structure
  - Users' preferences and items' features can be modeled using a few factors
  - User feature matrix $U \in \mathbb{R}^{K \times N}$
  - Item feature matrix $V \in \mathbb{R}^{K \times M}$



Matrix Factorization Based Methods

- Assumption
  - X has a low-rank structure
  - Users' preferences and items' features can be modeled using a few factors
  - User feature matrix $U \in \mathbb{R}^{K \times N}$
  - Item feature matrix $V \in \mathbb{R}^{K \times M}$



Matrix Factorization Based Methods

- Assumption
  - X has a low-rank structure
  - Users' preferences and items' features can be modeled using a few factors
  - User feature matrix $U \in \mathbb{R}^{K \times N}$
  - Item feature matrix $V \in \mathbb{R}^{K \times M}$

Methods	Loss	Constraints	Regularizations
SVD	L2 norm	None	None
L1-SVD	L1 norm	None	None
PMF	L2 norm	None	Frobenius Norm on U and V
NMF	L2 norm	U>0, V>0	None
MMMF	Hinge loss	None	Trace$(U^T V)$
RMF	Cross Entropy	None	Frobenius Norm on U and V

Loss?

Matrix Factorization Based Methods

- Assumption
  - X has a low-rank structure
  - Users' preferences and items' features can be modeled using a few factors
  - User feature matrix $U \in \mathbb{R}^{K \times N}$
  - Item feature matrix $V \in \mathbb{R}^{K \times M}$

Methods	Loss	Constraints	Regularizations
SVD	L2 norm	None	None
L1-SVD	L1 norm	None	None
PMF	L2 norm	None	Frobenius Norm on U and V
NMF	L2 norm	U>0, V>0	None
MMMF	Hinge loss	None	Trace$(U^T V)$
RMF	Cross Entropy	None	Frobenius Norm on U and V

Loss?

Probabilistic Matrix Factorization PMF

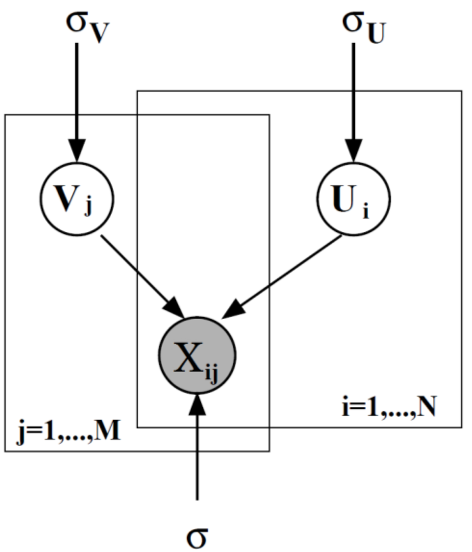
- Conditional distribution over observed ratings:

$$p(X|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(x_{ij}|g(U_i^T V_j),\sigma^2)]^{I_{ij}}$$

- Spherical Gaussian priors on user and item feature vectors:

$$p(U|\sigma_U^2) = \prod_{i=1}^{N}\mathcal{N}(U_i|0,\sigma_U^2)$$

$$p(V|\sigma_V^2) = \prod_{j=1}^{N}\mathcal{N}(V_j|0,\sigma_V^2)$$



- Maximize posterior:

$$p(U,V|X,\sigma^2,\sigma_U^2,\sigma_V^2) \propto p(X|U,V,\sigma^2)p(U|\sigma_U^2)p(V|\sigma_V^2)$$

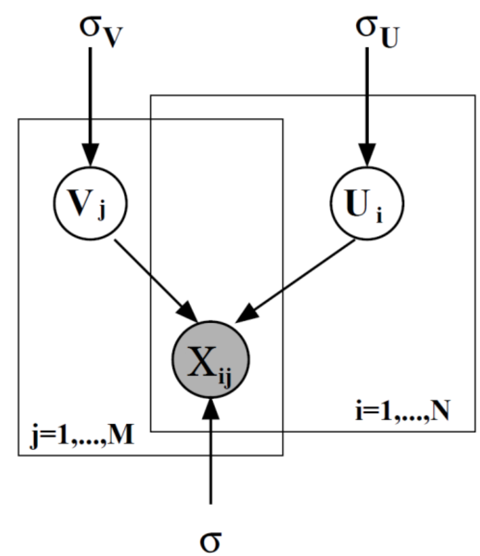
Probabilistic Matrix Factorization PMF

- Maximize

$$p(U, V|X, \sigma^2, \sigma_U^2, \sigma_V^2) \propto p(X|U, V, \sigma^2)p(U|\sigma_U^2)p(V|\sigma_V^2)$$

- Equivalent to minimize the following loss:

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(x_{ij} - g_{ij})^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$



- Using gradient descent to minimize loss:

$$U_i \leftarrow U_i - \eta\frac{\partial \mathcal{L}}{\partial U_i} \qquad \frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{M} I_{ij}(g_{ij} - x_{ij})g'_{ij}V_j + \lambda_U U_i$$

$$V_j \leftarrow V_j - \eta\frac{\partial \mathcal{L}}{\partial V_j} \qquad \frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{N} I_{ij}(g_{ij} - x_{ij})g'_{ij}U_i + \lambda_V V_j$$

Probabilistic Matrix Factorization PMF

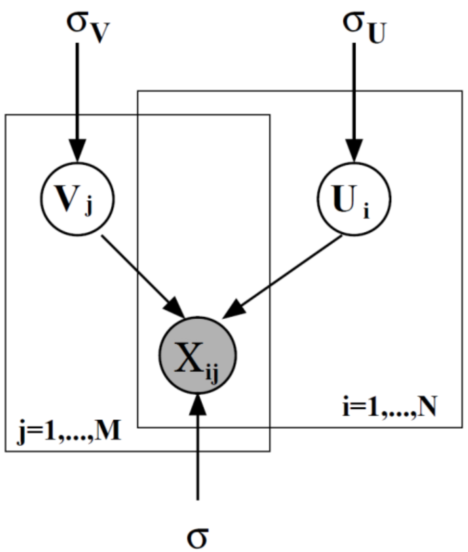
- Maximize

$$p(U, V|X, \sigma^2, \sigma_U^2, \sigma_V^2) \propto p(X|U, V, \sigma^2)p(U|\sigma_U^2)p(V|\sigma_V^2)$$

- Equivalent to minimize the following loss:

$$\mathcal{L} = \boxed{\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(x_{ij} - g_{ij})^2} + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$

Squared loss



- Using gradient descent to minimize loss:

$$U_i \leftarrow U_i - \eta \frac{\partial \mathcal{L}}{\partial U_i} \qquad \frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{M} I_{ij}(g_{ij} - x_{ij})g'_{ij}V_j + \lambda_U U_i$$

$$V_j \leftarrow V_j - \eta \frac{\partial \mathcal{L}}{\partial V_j} \qquad \frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{N} I_{ij}(g_{ij} - x_{ij})g'_{ij}U_i + \lambda_V V_j$$

Probabilistic Matrix Factorization PMF

- Maximize

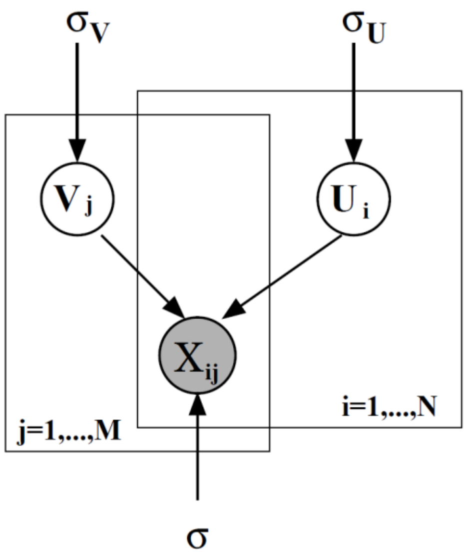
$$p(U,V|X,\sigma^2,\sigma_U^2,\sigma_V^2) \propto p(X|U,V,\sigma^2)p(U|\sigma_U^2)p(V|\sigma_V^2)$$

- Equivalent to minimize the following loss:

$$\mathcal{L} = \boxed{\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(x_{ij}-g_{ij})^2} + \boxed{\frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2}$$

Squared loss

Regularization



- Using gradient descent to minimize loss:

$$U_i \leftarrow U_i - \eta\frac{\partial \mathcal{L}}{\partial U_i} \qquad \frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{M} I_{ij}(g_{ij}-x_{ij})g'_{ij}V_j + \lambda_U U_i$$

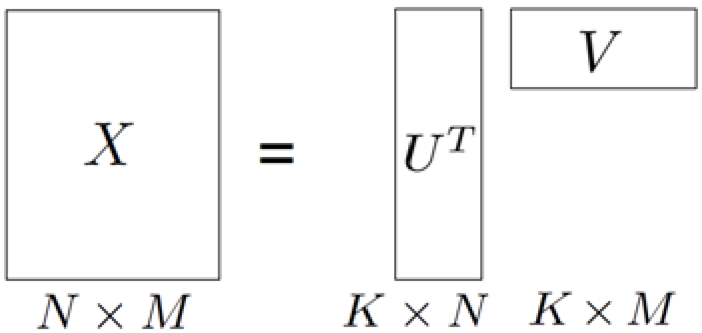
$$V_j \leftarrow V_j - \eta\frac{\partial \mathcal{L}}{\partial V_j} \qquad \frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{N} I_{ij}(g_{ij}-x_{ij})g'_{ij}U_i + \lambda_V V_j$$

Ranking Matrix Factorization RMF

- Top one probability
  - The probability that an item i being ranked on top

$$p_X(x_{ui}) = \frac{\exp(x_{ui})}{\sum_{k=1}^{M} I_{uk} \exp(x_{uk})}$$

$$p_{UV}(g_{ui}) = \frac{\exp(g_{ui})}{\sum_{k=1}^{M} I_{uk} \exp(g_{uk})}$$



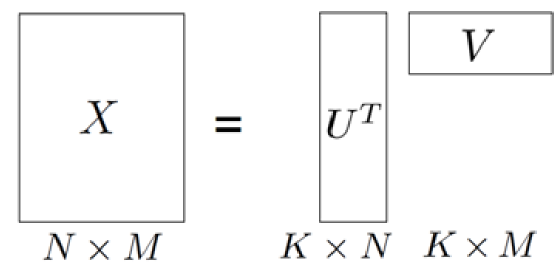
- Minimize cross entropy
  - Cross entropy measures the divergence between two distributions
  - Un-normalized KL-divergence

$$H(p,q) = E_p[-\log\ q] = -\sum_{x} p(x) \log q(x)$$

Ranking Matrix Factorization RMF

- Model loss is defined as:

$$\mathcal{L} = \sum_{i=1}^{N} \left\{ -\sum_{j=1}^{M} I_{ij} \frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(x_{ik})} \log \left\{ \frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(g_{ik})} \right\} \right\} + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2$$



- Using gradient descent to minimize:

$$U_i \leftarrow U_i - \eta \frac{\partial \mathcal{L}}{\partial U_i} \qquad \frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{M} I_{ij} \left\{ \frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(g_{ik})} - \frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(x_{ik})} \right\} g'_{ij} V_j + \lambda_U U_i$$

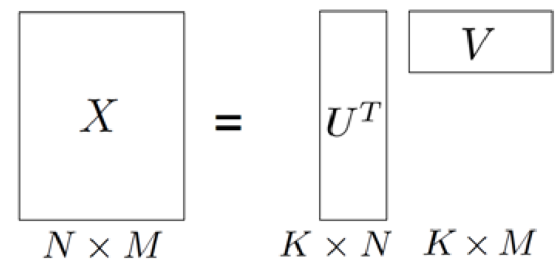
$$V_j \leftarrow V_j - \eta \frac{\partial \mathcal{L}}{\partial V_j} \qquad \frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{N} I_{ij} \left\{ \frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(g_{ik})} - \frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(x_{ik})} \right\} g'_{ij} U_i + \lambda_V V_j$$

Ranking Matrix Factorization RMF

- Model loss is defined as:

$$\mathcal{L} = \sum_{i=1}^{N}\left\{-\sum_{j=1}^{M} I_{ij}\frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik}\exp(x_{ik})}\log\left\{\frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik}\exp(g_{ik})}\right\}\right\} + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$

Cross Entropy



- Using gradient descent to minimize:

$$U_i \leftarrow U_i - \eta\frac{\partial\mathcal{L}}{\partial U_i}$$

$$V_j \leftarrow V_j - \eta\frac{\partial\mathcal{L}}{\partial V_j}$$

$$\frac{\partial\mathcal{L}}{\partial U_i} = \sum_{j=1}^{M} I_{ij}\left\{\frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik}\exp(g_{ik})} - \frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik}\exp(x_{ik})}\right\}g'_{ij}V_j + \lambda_U U_i$$

$$\frac{\partial\mathcal{L}}{\partial V_j} = \sum_{i=1}^{N} I_{ij}\left\{\frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik}\exp(g_{ik})} - \frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik}\exp(x_{ik})}\right\}g'_{ij}U_i + \lambda_V V_j$$

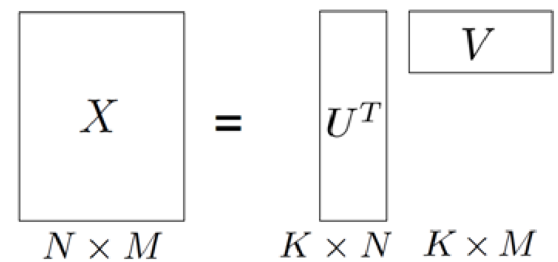
Ranking Matrix Factorization RMF

- Model loss is defined as:

$$\mathcal{L} = \sum_{i=1}^{N} \left\{ - \sum_{j=1}^{M} I_{ij} \frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(x_{ik})} \log \left\{ \frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(g_{ik})} \right\} \right\} + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2$$

Cross Entropy

Regularization



- Using gradient descent to minimize:

$$U_i \leftarrow U_i - \eta \frac{\partial \mathcal{L}}{\partial U_i} \qquad \frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{M} I_{ij} \left\{ \frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(g_{ik})} - \frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(x_{ik})} \right\} g'_{ij} V_j + \lambda_U U_i$$

$$V_j \leftarrow V_j - \eta \frac{\partial \mathcal{L}}{\partial V_j} \qquad \frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{N} I_{ij} \left\{ \frac{\exp(g_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(g_{ik})} - \frac{\exp(x_{ij})}{\sum_{k=1}^{M} I_{ik} \exp(x_{ik})} \right\} g'_{ij} U_i + \lambda_V V_j$$

Problems Faced by Recommender Systems

- Dynamic system are handled by static methods
  - Online learning algorithms
- Unrealistic implicit assumptions
  - Response aware methods
- Spammer problem
  - User reputation estimation framework and method
- Cold-start problem
  - Combine ratings with reviews

Outline

- Introduction and Background Review
- Online Collaborative Filtering
- Response Aware Collaborative Filtering
- User Reputation Estimation
- Combine Ratings with Reviews
- Conclusion

Motivation

In real-world recommender systems

- New ratings are collected constantly
  - Update the model
- New users
- New items
- Huge dataset

In laboratory simulated experiments

- Dataset is prepared beforehand
- No new ratings, users or items
- Relatively small dataset

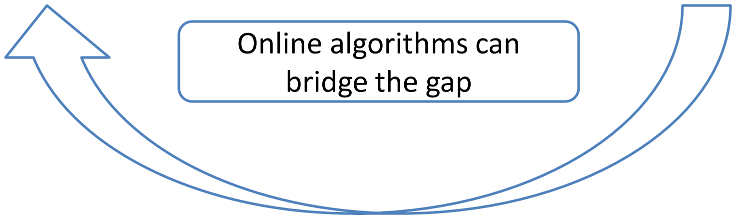
Motivation

In real-world recommender systems

- New ratings are collected constantly
  - Update the model
- New users
- New items
- Huge dataset

In laboratory simulated experiments

- Dataset is prepared beforehand
- No new ratings, users or items
- Relatively small dataset



Online Algorithms for PMF and RMF

- We propose two online algorithms respectively for both PMF and RMF
  - Stochastic gradient descent
    - Adjust model *stochastically* for each observation
  - Regularized dual averaging
    - Maintain an approximated average gradient
    - Solve an easy optimization problem at each iteration

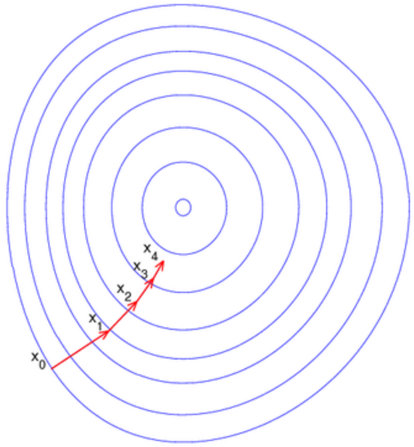
Stochastic Gradient Descent PMF

- Recall the loss function for PMF

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(x_{ij} - g_{ij})^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$

- Squared loss can be dissected and associated with each observation triplet $(u, i, x) \in Q$

$$\mathcal{L}_{(u,i,x)} = (x_{ui} - g_{ui})^2 + \frac{\lambda_U}{2}\|U_u\|_2^2 + \frac{\lambda_V}{2}\|V_i\|_2^2$$



- Update model using gradient of this loss:

$$U_u \leftarrow U_u - \eta((g_{ui} - x)g'_{ui}V_i + \lambda_U U_u),$$

$$V_i \leftarrow V_i - \eta((g_{ui} - x)g'_{ui}U_u + \lambda_V V_i),$$

Regularized Dual Averaging PMF

- Maintain the approximated average gradient

$$\boxed{Y_{U_u}} \leftarrow \frac{t_u - 1}{t_u} Y_{U_u} + \frac{1}{t_u}(g_{ui} - x) g'_{ui} V_i$$

$$\boxed{\Sigma_{i \in I_u} (g_{ui} - x) g'_{ui} V_i / \mathrm{t_u}}$$

$$Y_{V_i} \leftarrow \frac{t_v - 1}{t_v} Y_{V_i} + \frac{1}{t_v}(g_{ui} - x) g'_{ui} U_u$$

Regularized Dual Averaging PMF

- Maintain the approximated average gradient

Number of items rated by u

$$Y_{U_u} \leftarrow \frac{t_u - 1}{t_u} Y_{U_u} + \frac{1}{t_u}(g_{ui} - x) g'_{ui} V_i$$

$\Sigma_{i \in I_u} (g_{ui} - x) g'_{ui} V_i / \mathrm{t_u}$

$$Y_{V_i} \leftarrow \frac{t_v - 1}{t_v} Y_{V_i} + \frac{1}{t_v}(g_{ui} - x) g'_{ui} U_u$$

Regularized Dual Averaging PMF

- Maintain the approximated average gradient

Number of items rated by u

Previous gradient

$$Y_{U_u} \leftarrow \frac{t_u - 1}{t_u} Y_{U_u} + \frac{1}{t_u}(g_{ui} - x)g'_{ui}V_i$$

$\Sigma_{i \in I_u}(g_{ui} - x)g'_{ui}V_i/\mathrm{t_u}$

$$Y_{V_i} \leftarrow \frac{t_v - 1}{t_v} Y_{V_i} + \frac{1}{t_v}(g_{ui} - x)g'_{ui}U_u$$

Regularized Dual Averaging PMF

- Maintain the approximated average gradient

Number of items rated by u

Previous gradient

$$Y_{U_u} \leftarrow \frac{t_u - 1}{t_u} Y_{U_u} + \frac{1}{t_u}(g_{ui} - x) g'_{ui} V_i$$

$\Sigma_{i \in I_u} (g_{ui} - x) g'_{ui} V_i / \mathrm{t_u}$

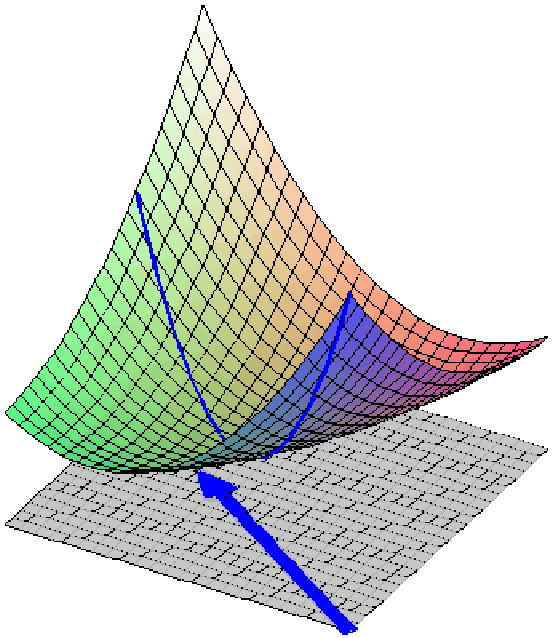
Gradient due to new observation $(u, i, x) \in Q$

$$Y_{V_i} \leftarrow \frac{t_v - 1}{t_v} Y_{V_i} + \frac{1}{t_v}(g_{ui} - x) g'_{ui} U_u$$

Regularized Dual Averaging PMF

- Solve the following optimization problem to obtain
  - New user feature vector U_u
  - New item feature vector V_i

$$U_u = \arg\min_w \{Y_{U_u}^T w + \lambda_U \|w\|_2^2\}$$
$$V_i = \arg\min_w \{Y_{V_i}^T w + \lambda_V \|w\|_2^2\}$$



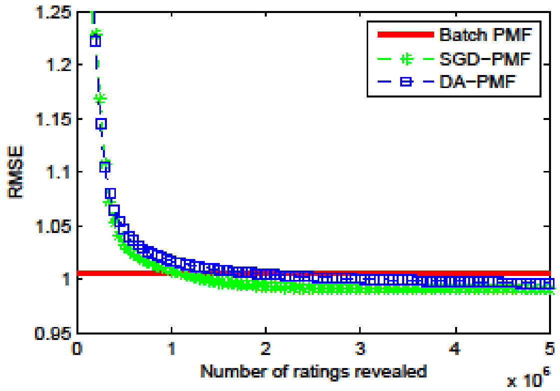
Experiments and Results

- We conduct experiments on real life data set
  - MovieLens, Yahoo! Music and Jester

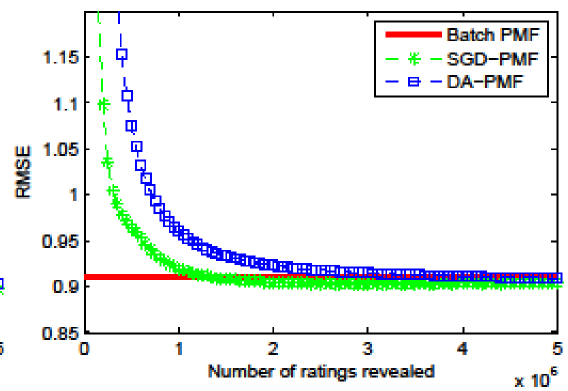
Dataset	Users	Movies	Ratings	Rating Range
MovieLens	6040	3900	1,000,209	1-5
Yahoo! Music	1,000,990	624,961	252,800,275	1-100
Jester	24,938	100	1,810,455	-10-10

  - Three settings
    - T1: 10% training, 90% testing
    - T5: 50% training, 50% testing
    - T9: 90% training, 10% testing

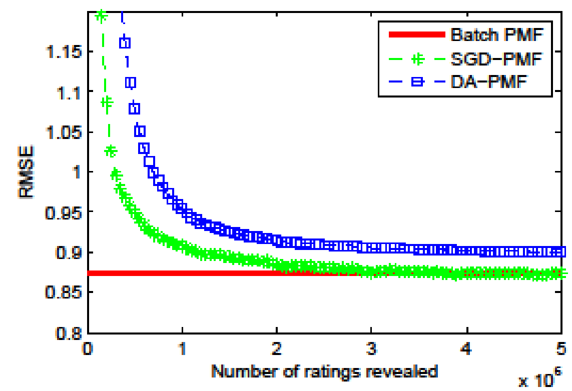
Online versus Batch Algorithms



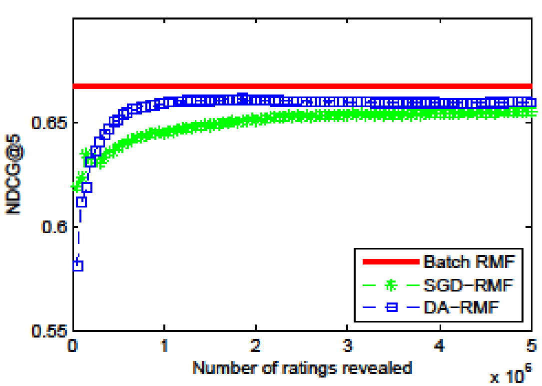
T1



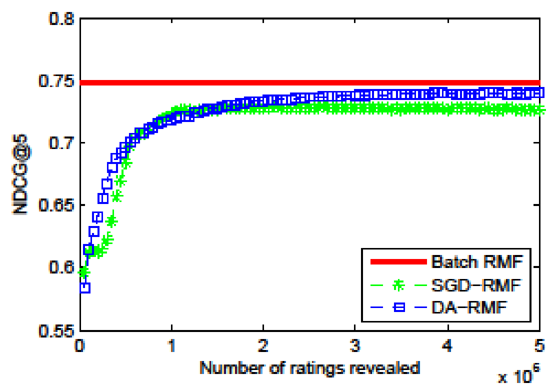
T5



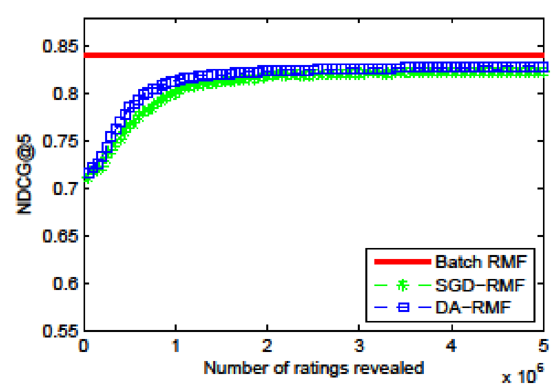
T9



T1



T5



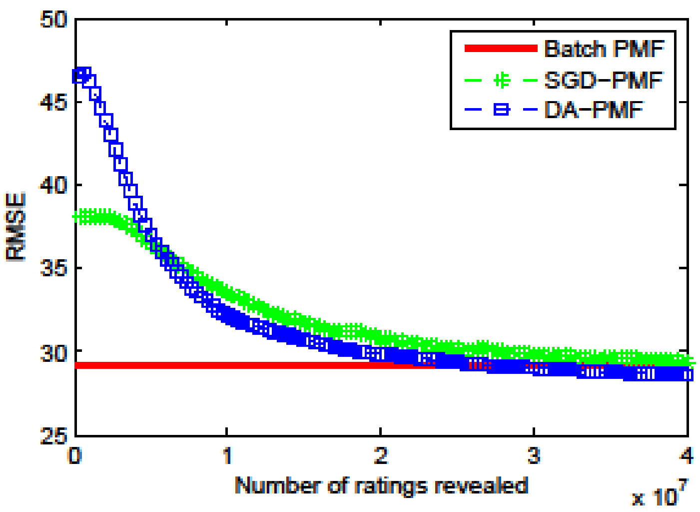
T9

Scalability to Large Dataset

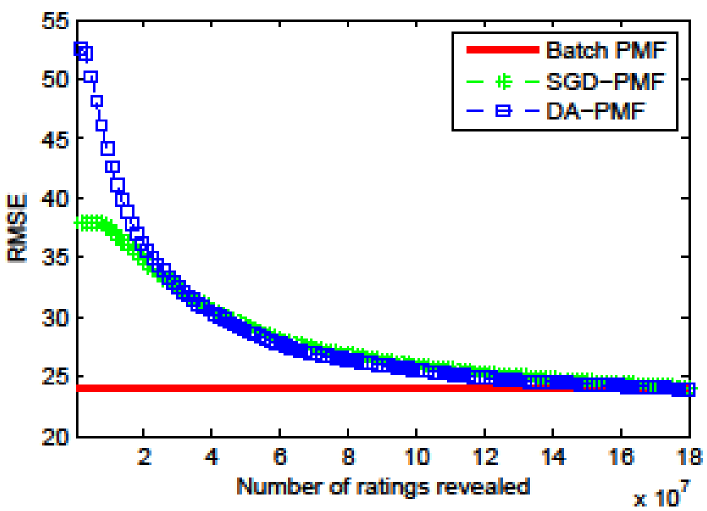
- Experiment environment
  - Linux workstation (Xeon Dual Core 2.4 GHz, 32 GB RAM)
  - Batch PMF: 8 hours for 120 iteration
  - Online PMF: 10 minutes

Scalability to Large Dataset

- Experiment environment
  - Linux workstation (Xeon Dual Core 2.4 GHz, 32 GB RAM)
  - Batch PMF: 8 hours for 120 iteration
  - Online PMF: 10 minutes



T1



T5

Scalability to Large Dataset

- Experiment environment
  - Linux workstation (Xeon Dual Core 2.4 GHz, 32 GB RAM)
  - Batch PMF: 8 hours for 120 iteration
  - Online PMF: 10 minutes

Outline

- Introduction and Background Review
- Online Collaborative Filtering
- Response Aware Collaborative Filtering
- User Reputation Estimation
- Combine Ratings with Reviews
- Conclusion

Unrealistic Assumptions

- Implicit assumption of previous CF methods
  - All response or random response

Unrealistic Assumptions

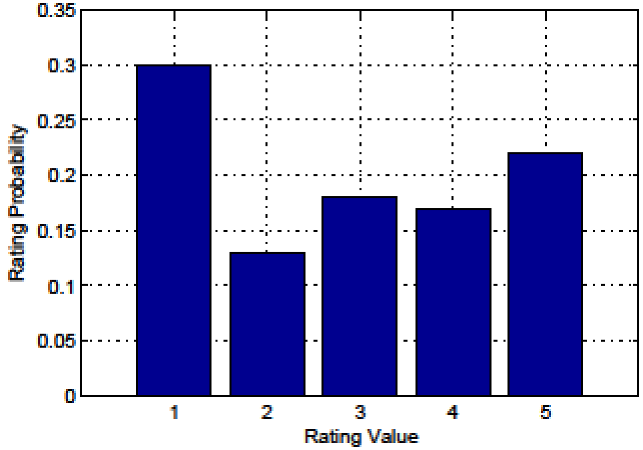
- Implicit assumption of previous CF methods
  - All response or random response

	I1	I2	I3	I4	I5
U1	5	4			
U2		5		4	
U3	4			4	
U4	5		5		
U5		4			5

Unrealistic Assumptions

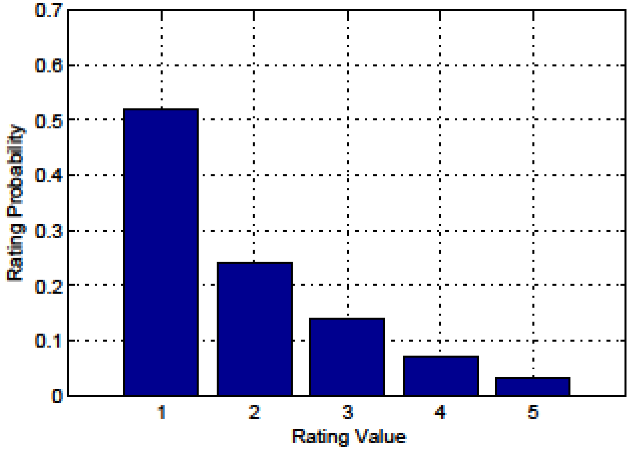
Rating value distribution of user selected items

- A lot of high rating items



Rating value distribution of randomly selected items

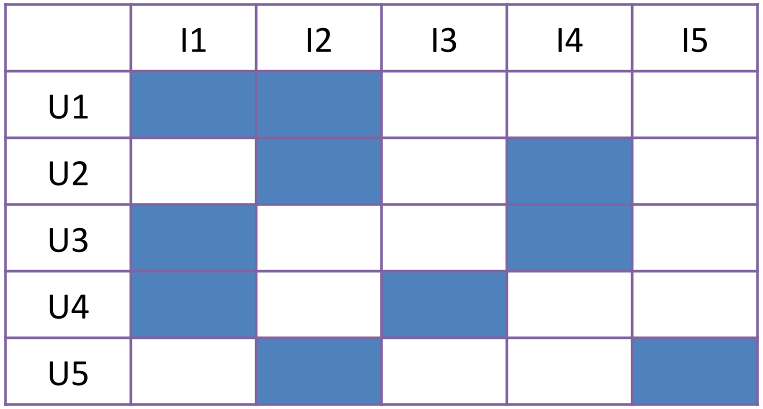
- Very few high rating items



Response Aware Collaborative Filtering

- Information embedded in ratings
  - Rating value indicate preferences
  - Rating response patterns

	I1	I2	I3	I4	I5
U1	5	4			
U2		5		4	
U3	4			4	
U4	5		5		
U5		4			5



Missing Data Theory

- Two step procedure

Data Generation Model

- Generate full data matrix
- $P(X|\theta)$

Data Observation Model

- Model observation process
- $P(R|X,\mu)$

$$\begin{aligned} P(R, X|\mu, \theta) &= P(R|X, \mu, \theta)P(X|\mu, \theta) \\ &= P(R|X, \mu)P(X|\theta), \end{aligned}$$

Missing Data Theory

- Two step procedure

Data Generation Model

- Generate full data matrix
- $P(X|\theta)$

Data Observation Model

- Model observation process
- $P(R|X,\mu)$

	I1	I2	I3	I4	I5
U1	5	4	1	1	2
U2	3	5	2	4	4
U3	4	1	3	4	1
U4	5	3	5	2	3
U5	2	4	1	3	5

X

$$P(R, X|\mu, \theta) = P(R|X, \mu, \theta)P(X|\mu, \theta)$$
$$= P(R|X, \mu)P(X|\theta),$$

Missing Data Theory

- Two step procedure

Data Generation Model

- Generate full data matrix
- $P(X|\theta)$

Data Observation Model

- Model observation process
- $P(R|X,\mu)$

	I1	I2	I3	I4	I5
U1	5	4	1	1	2
U2	3	5	2	4	4
U3	4	1	3	4	1
U4	5	3	5	2	3
U5	2	4	1	3	5

X

	I1	I2	I3	I4	I5
U1	1	1	0	0	0
U2	0	1	0	1	0
U3	1	0	0	1	0
U4	1	0	1	0	0
U5	0	1	0	0	1

R

$$P(R, X|\mu, \theta) = P(R|X, \mu, \theta)P(X|\mu, \theta)$$
$$= P(R|X, \mu)P(X|\theta),$$

Missing Data Theory

- Three missing data assumptions
  - Missing Completely At Random (MCAR)

$$P(R|X,\mu) = P(R|\mu)$$

Example: Response is determined by a Bernoulli tail with success probability μ

  - Missing At Random (MAR)

$$P(R|X,\mu) = P(R|X_{obs},\mu)$$

  - Not Missing At Random (NMAR)
    - If Both MCAR and MAR fail to hold

Missing Data Theory

- Three missing data assumptions
  - Missing Completely At Random (MCAR)

$$P(R|X, \mu) = P(R|\mu)$$

Example: Response is determined by a Bernoulli tail with success probability μ

  - Missing At Random (MAR)

$$P(R|X, \mu) = P(R|X_{obs}, \mu)$$

What does it mean?

  - Not Missing At Random (NMAR)
    - If Both MCAR and MAR fail to hold

Missing Data Theory

- Three missing data assumptions
  - Missing Completely At Random (MCAR)

$$P(R|X,\mu) = P(R|\mu)$$

Example: Response is determined by a Bernoulli tail with success probability μ

  - Missing At Random (MAR)

$$P(R|X,\mu) = P(R|X_{obs},\mu)$$

What does it mean?

  - Not Missing At Random (NMAR)
    - If Both MCAR and MAR fail to hold

Example: Response is related to the rating value

Missing Data Theory

- If MAR fail to hold, ML learns biased data model parameter θ

$$
\begin{aligned}
& \mathcal{L}(\mu, \theta | X_{obs}, R) = P(R, X_{obs} | \mu, \theta) \\
= & \int_{X_{mis}} P(R, X | \mu, \theta) dX_{mis} \\
= & \int_{X_{mis}} P(R | X, \mu) P(X | \theta) dX_{mis} \\
= & \int_{X_{mis}} P(R | X_{obs}, \mu) P(X | \theta) dX_{mis} \\
= & P(R | X_{obs}, \mu) \int_{X_{mis}} P(X | \theta) dX_{mis} \\
= & P(R | X_{obs}, \mu) P(X_{obs} | \theta) \\
\propto & P(X_{obs} | \theta).
\end{aligned}
$$

Missing Data Theory

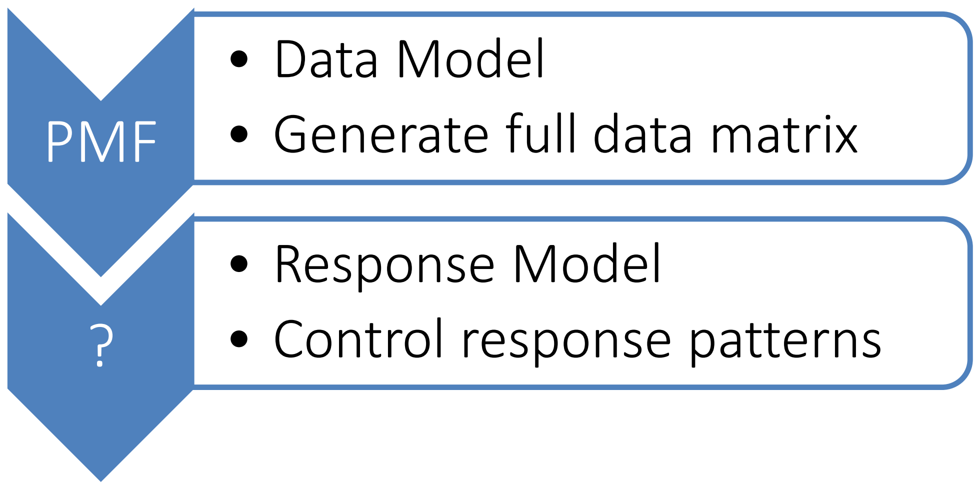
- If MAR fail to hold, ML learns biased data model parameter θ

$$
\begin{aligned}
& \mathcal{L}(\mu, \theta | X_{obs}, R) = P(R, X_{obs} | \mu, \theta) \\
= & \int_{X_{mis}} P(R, X | \mu, \theta) dX_{mis} \\
= & \int_{X_{mis}} P(R | X, \mu) P(X | \theta) dX_{mis} \\
= & \int_{X_{mis}} P(R | X_{obs}, \mu) P(X | \theta) dX_{mis} \\
= & P(R | X_{obs}, \mu) \int_{X_{mis}} P(X | \theta) dX_{mis} \\
= & P(R | X_{obs}, \mu) P(X_{obs} | \theta) \\
\propto & P(X_{obs} | \theta).
\end{aligned}
$$

Definition of MAR appears naturally

Response Aware PMF

- Follow the two steps procedure under matrix factorization framework



$$P(R, X|U, V, \mu, \sigma^2) = P(R|X, U, V, \mu, \sigma^2)P(X|U, V, \sigma^2)$$

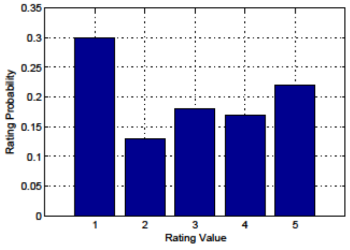
Response Models

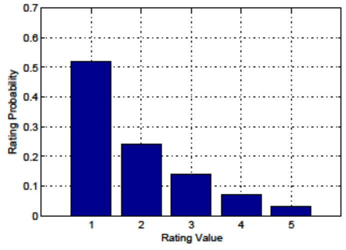
Rating dominant response model

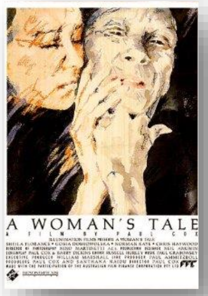
- Rating value alone determines the response

Context aware response model

- Context aware
  - Rating value
  - Heavy rater vs. light rater
  - Hot item vs. obscure item







Response Models

- We use Bernoulli distribution to model the response probability $P\left(R_{ij} \middle| X_{ij}, \mathrm{U_i}, \mathrm{V_j}, \mu, \sigma^2\right) \sim \mathrm{Bernoulli}(\mu)$

- Rating Dominant
  - μ is determined by the rating value alone
  - $R_{ij} \sim \mathrm{Bernoulli}(\mu_{\mathrm{X_{ij}}})$
  - Only D different μs

- Context Aware
  - μ is determined by rating value, user and item
  - $R_{ij} \sim \mathrm{Bernoulli}(\mu_{ijk})$
  - $\mu_{ijk} \sim \frac{1}{1+\exp\{-(\delta_k + \Theta_U U_i + \Theta_V V_j)\}}$

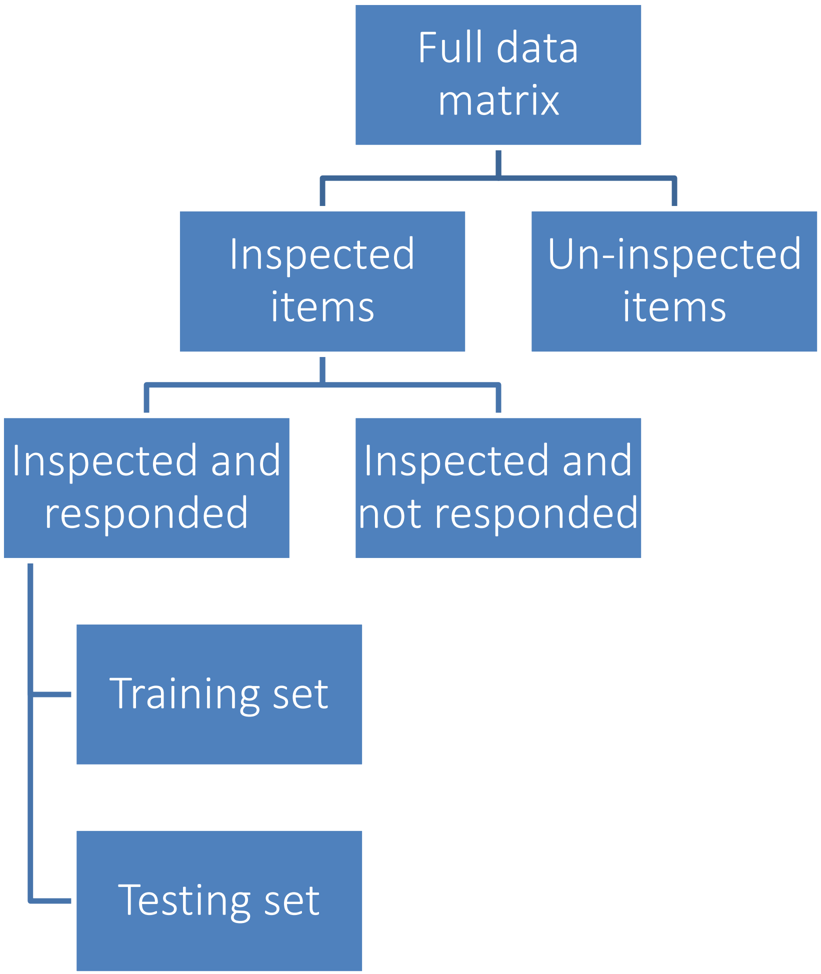
- Both can be learned using alternative gradient descent

Experiments and Results

- We conduct experiments on both synthetic and real-world datasets
  - Synthetic dataset
  - Yahoo! Music ratings for user selected and randomly selected songs
- We device three protocols to simulate various conditions
  - Traditional
  - Realistic
  - Adversarial

Users	Items	Collected ratings	Survey users	Survey ratings
15,400	1,000	311,704	5,400	54,000

Generation of Synthetic Dataset



$$U_i \sim \mathcal{N}(\mathbf{0}_K, \sigma_U^2 \mathbf{I}_K), \quad i = 1, \ldots, N,$$
$$V_j \sim \mathcal{N}(\mathbf{0}_K, \sigma_V^2 \mathbf{I}_K), \quad j = 1, \ldots, M,$$
$$X_{ij} = \lceil g(U_i^T V_j) \times D \rceil.$$

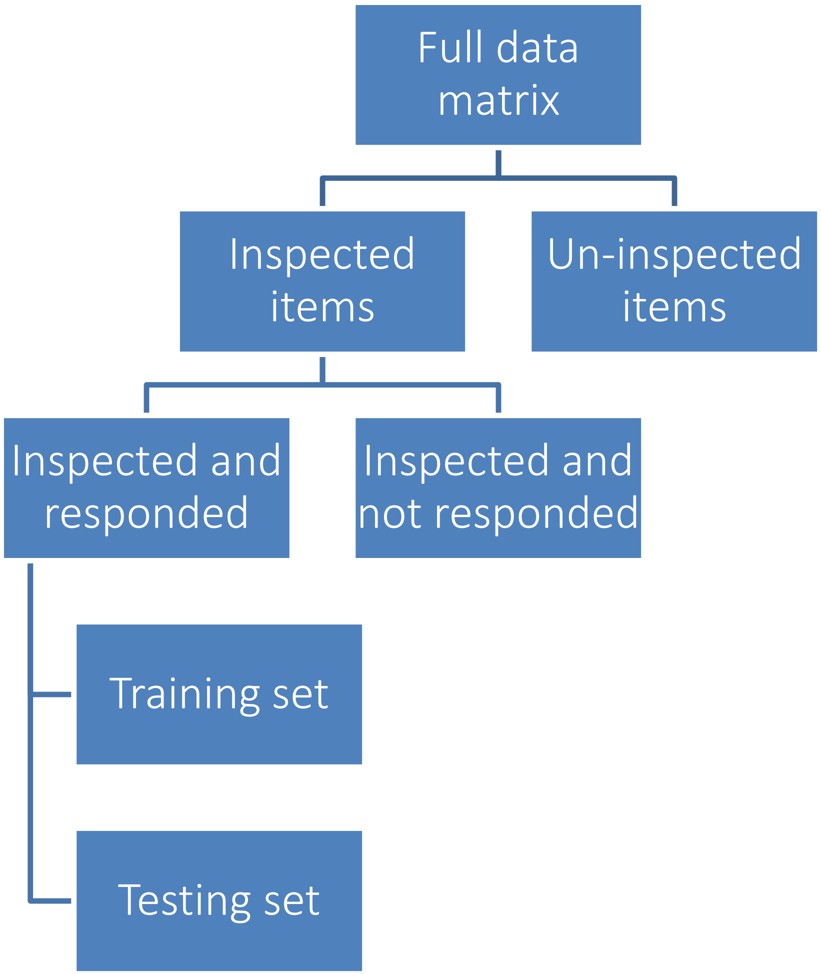
Bernoulli trail $P_{inspect}$

Bernoulli trail $P_{X_{ij}}$

N	M	D	K	$P_{inspect}$
1000	1000	5	5	0.3
P_1	P_2	P_3	P_4	P_5
0.073	0.068	0.163	0.308	0.931

Random 90% 10% partition

Generation of Synthetic Dataset



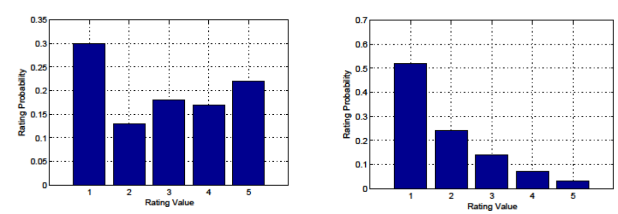
$$U_i \sim \mathcal{N}(\mathbf{0}_K, \sigma_U^2 \mathbf{I}_K), \quad i = 1, \ldots, N,$$
$$V_j \sim \mathcal{N}(\mathbf{0}_K, \sigma_V^2 \mathbf{I}_K), \quad j = 1, \ldots, M,$$
$$X_{ij} = \lceil g(U_i^T V_j) \times D \rceil.$$

Bernoulli trail $P_{inspect}$

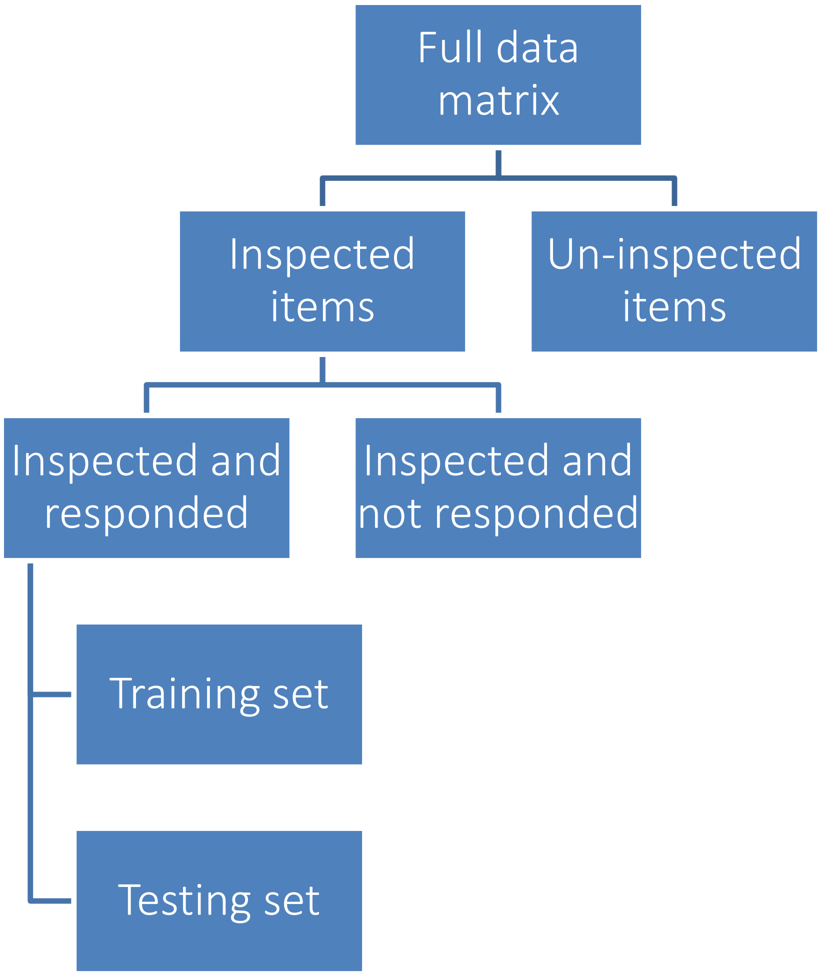
Bernoulli trail $P_{X_{ij}}$

N	M	D	K	$P_{inspect}$
1000	1000	5	5	0.3
P_1	P_2	P_3	P_4	P_5
0.073	0.068	0.163	0.308	0.931

Random 90% 10% partition



Generation of Synthetic Dataset



$$U_i \sim \mathcal{N}(\mathbf{0}_K, \sigma_U^2 \mathbf{I}_K), \quad i = 1, \ldots, N,$$
$$V_j \sim \mathcal{N}(\mathbf{0}_K, \sigma_V^2 \mathbf{I}_K), \quad j = 1, \ldots, M,$$
$$X_{ij} = \lceil g(U_i^T V_j) \times D \rceil.$$

Bernoulli trail $P_{inspect}$

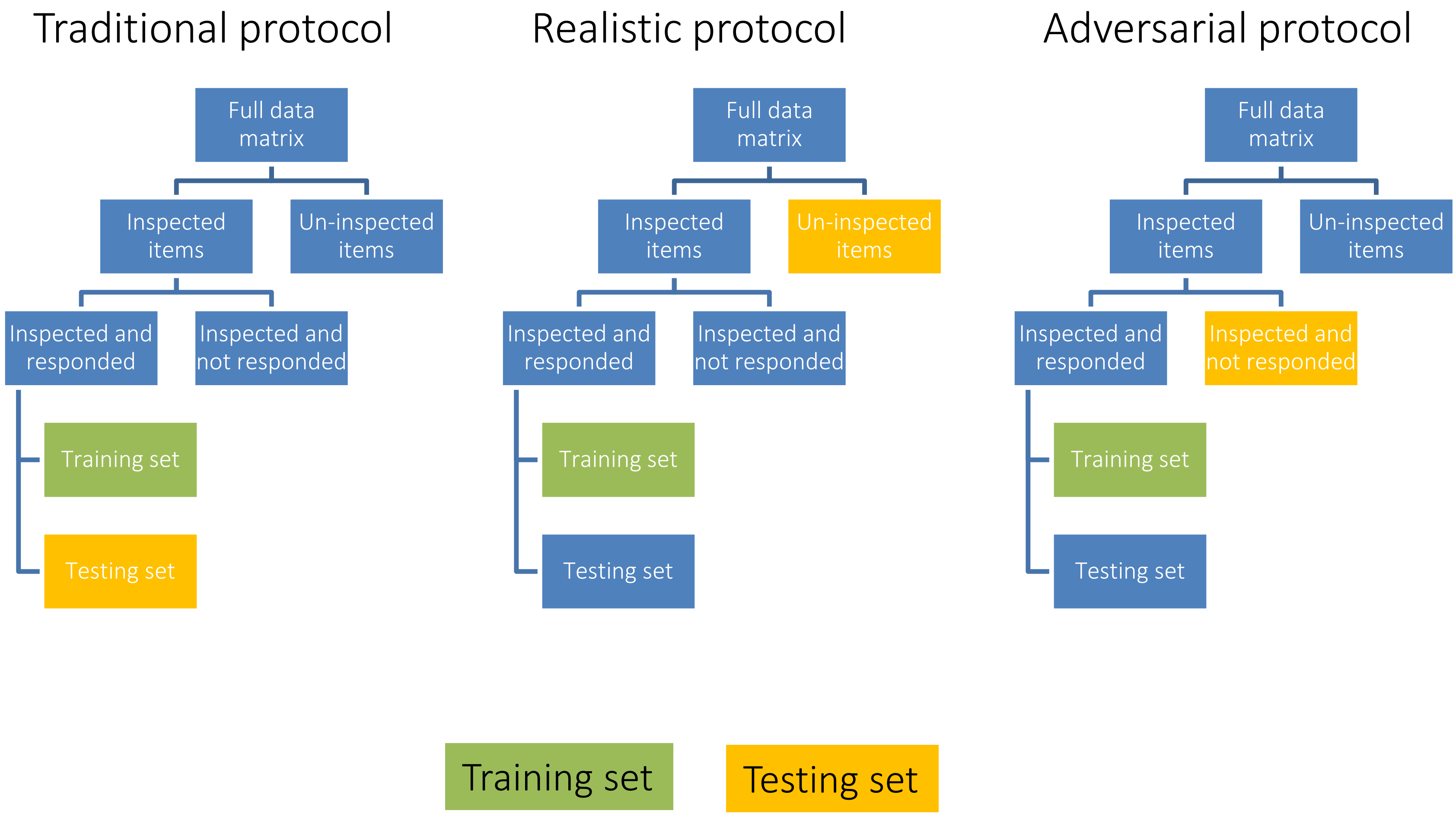
Bernoulli trail $P_{X_{ij}}$

N	M	D	K	$P_{inspect}$
1000	1000	5	5	0.3

P_1	P_2	P_3	P_4	P_5
0.073	0.068	0.163	0.308	0.931

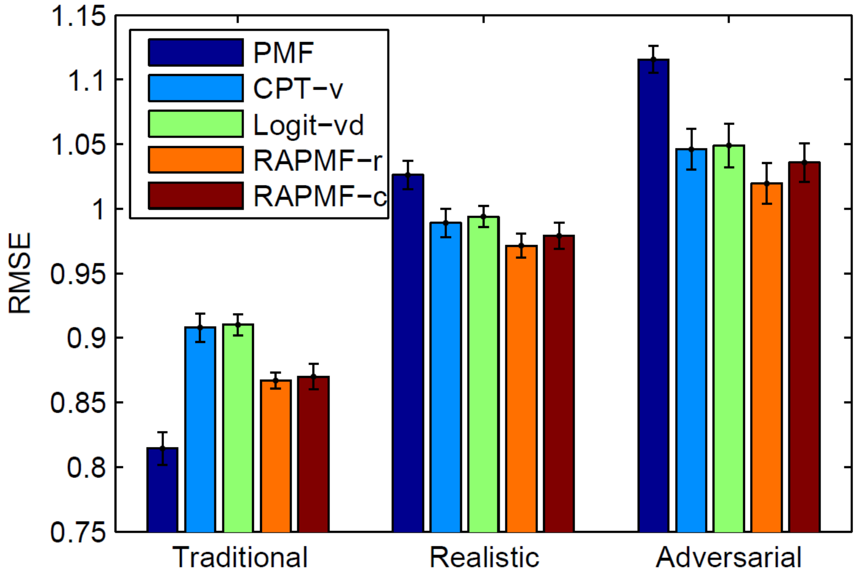
Random 90% 10% partition

Three Protocols

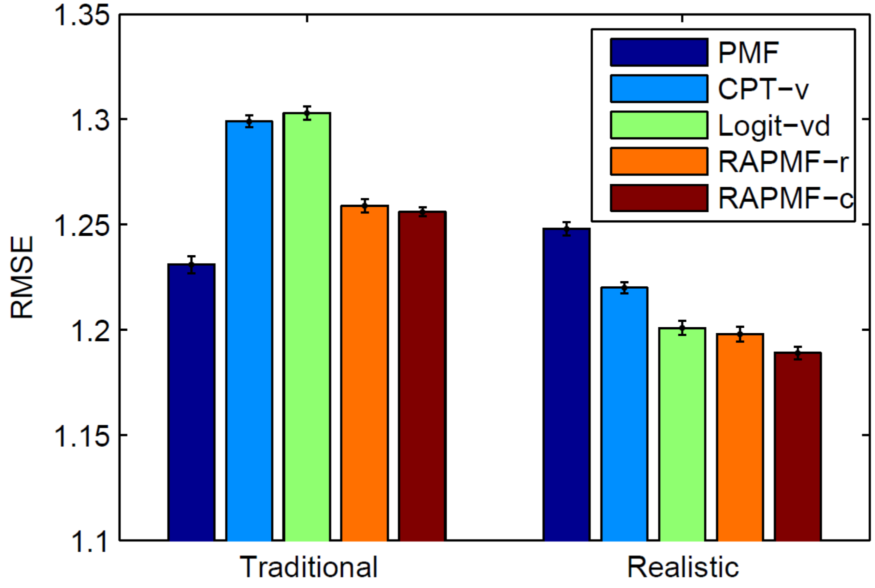


Results

- Performance of our proposed model versus various baseline models



Synthetic dataset



Yahoo dataset

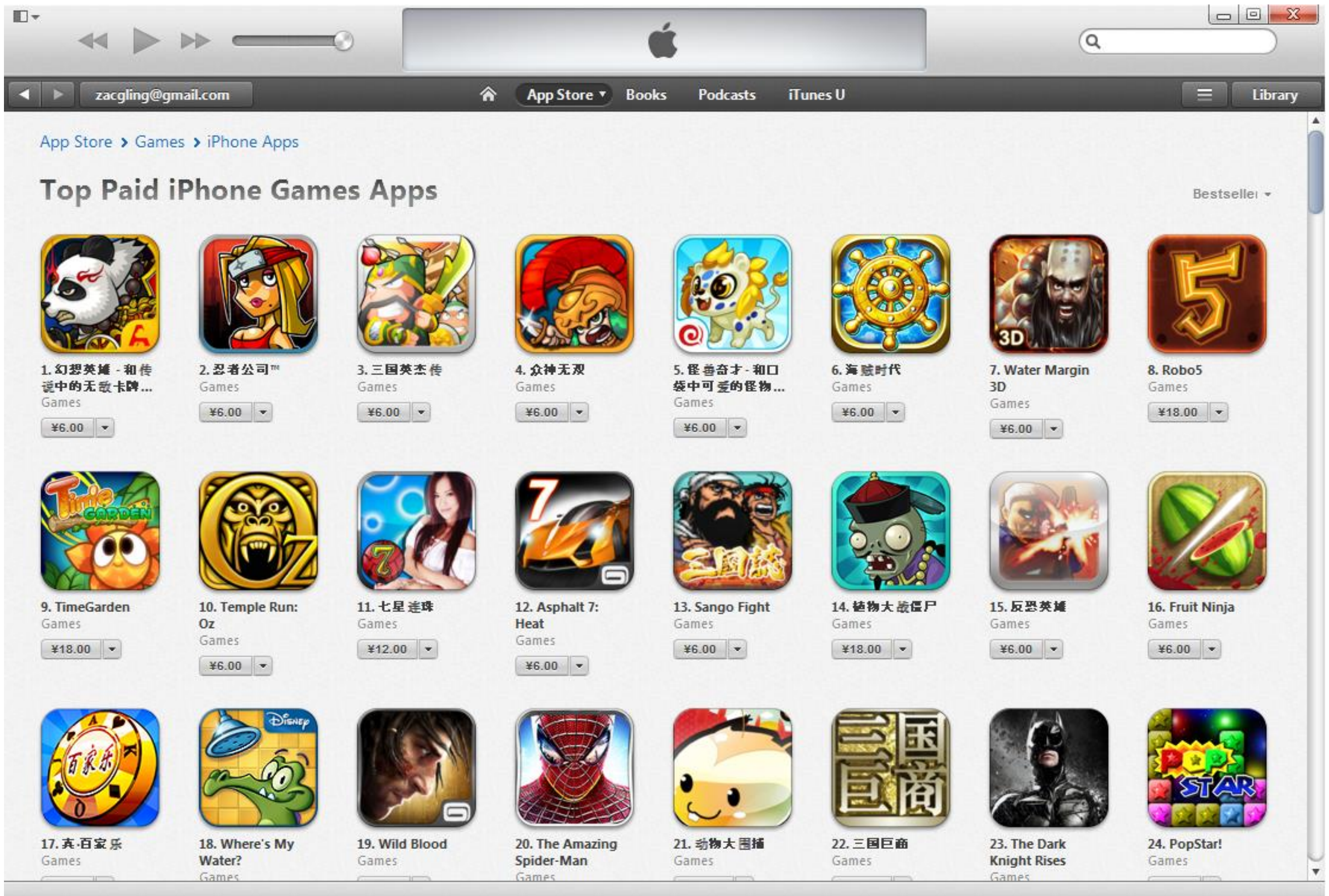
Outline

- Introduction and Background Review
- Online Collaborative Filtering
- Response Aware Collaborative Filtering
- User Reputation Estimation
- Combine Ratings with Reviews
- Conclusion

Spammer Problem

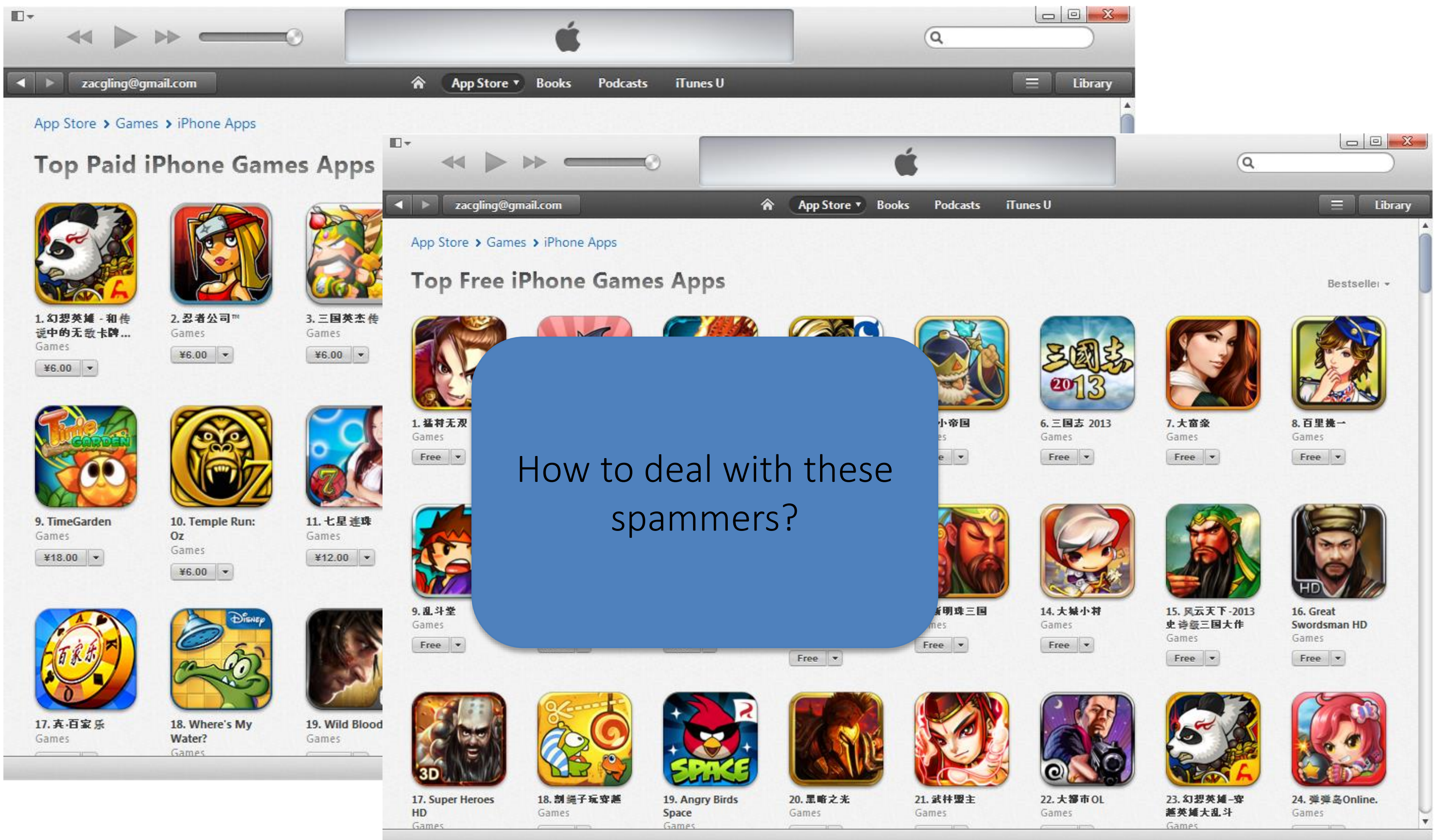
Spammer Problem

Spammer Problem



Spammer Problem

Spammer Problem



How to deal with these spammers?

Spammer Problem

Spammer Problem

Problem Statement

- Reputation estimation in online rating system
  - Given N users $\{u_1, u_2, \cdots, u_N\}$ ratings on M items and arrange them in a partially observed matrix R
  - Calculate reputation scores $\{c_1, c_2, \cdots, c_N\}$, where the score $c \in [0,1]$, for all the N users such that a normal user u_i should have a large c_i and a spam user u_j should have a small c_j

Reputation Estimation Framework

- Require three ingredients to work
  - Prediction Model
    - Provide reasonable model for normal users
    - Collaborative filtering methods can be readily used
  - Penalty Function
    - Summarize unexpectedness of a user
  - Link Function
    - Link the unexpectedness of a user to the reputation of the user

Prediction Model

- Let's assume that a normal user's behavior is modeled by $\mathcal{H}$, and the observed rating r_{ij} is a Gaussian R.V. centered at $\mathcal{H}(i,j)$, with variance σ^2

$$r_{ij} \sim \mathcal{N}(\mathcal{H}(i,j), \sigma^2)$$

- Then the log-likelihood of observing r_{ij} given $\mathcal{H}(i,j)$ is

$$\begin{aligned}
\mathcal{L}_{ij} &= \log(P(r_{ij}|\mathcal{H}(i,j))) \\
&= \log(\mathcal{N}(r_{ij}|\mathcal{H}(i,j), \sigma^2)) \\
&= C - \frac{1}{2\sigma^2}(r_{ij} - \mathcal{H}(i,j))^2
\end{aligned}$$

Prediction Model

- Let's assume that a normal user's behavior is modeled by $\mathcal{H}$, and the observed rating r_{ij} is a Gaussian R.V. centered at $\mathcal{H}(i,j)$,with variance σ^2

$$r_{ij} \sim \mathcal{N}(\mathcal{H}(i,j), \sigma^2)$$

- Then the log-likelihood of observing r_{ij} given $\mathcal{H}(i,j)$ is

$$\begin{aligned}\mathcal{L}_{ij} &= \log(P(r_{ij}|\mathcal{H}(i,j))) \\ &= \log(\mathcal{N}(r_{ij}|\mathcal{H}(i,j), \sigma^2)) \\ &= C - \frac{1}{2\sigma^2}\boxed{(r_{ij} - \mathcal{H}(i,j))^2}\end{aligned}$$

Prediction Model

- The *unexpectedness* of observing r_{ij}, based on $\mathcal{H}$ is

$$s_{ij} = (r_{ij} - \mathcal{H}(i,j))^2$$

- Related to self-information under mild condition
  - Self-information is a measure of the information content associated with the outcome of a random variable
  - The larger the self-information, the more surprising it is
  - Measure the "discordant" of r_{ij} with all other known ratings as seen by $\mathcal{H}$

Penalty Function

- Summarize the set of unexpectedness $\{s_{ij}\}$ into one quantity s_i or s_j
- Sample penalty function, arithmetic mean

$$s_i = \frac{1}{\|\mathcal{I}_i\|} \sum_{j \in \mathcal{I}_i} s_{ij}$$

Link function

- Relate the unexpectedness s_i to the reputation c_i
- Convenient to require that c_i lie in $[0,1]$
- Sample link function

$$c_i = 1 - \frac{s_i}{s_{max}}$$

Link function

- Relate the unexpectedness s_i to the reputation c_i
- Convenient to require that c_i lie in $[0,1]$
- Sample link function

$$c_i = 1 - \frac{s_i}{s_{max}}$$

Maximum possible value of s_i

Adaptability of the Framework

- The framework can capture existing reputation estimation methods

Algorithms	Prediction model	Penalty function	Link function
Mizzaro's algorithm	$\mathcal{H}(i,j) = \Sigma_{i\in\mathcal{U}_j} c_i r_{ij} / \Sigma_{i\in\mathcal{U}_j} c_i$	$s_j = \Sigma_{i\in\mathcal{U}_j} c_i$	$c_i = \frac{\sum_{j\in\mathcal{I}_i} s_j (1-\sqrt{\sqrt{s_{ij}}/s_{max}})}{\sum_{j\in\mathcal{I}_j} s_j}$
Laureti's algorithm	Same as above	$s_i = \frac{1}{\lVert\mathcal{I}_i\rVert} \sum_{j\in\mathcal{I}_i} s_{ij}$	$c_i = (s_i + \epsilon)^{-\beta}$
De Kerchove's algorithm	Same as above	Same as above	$c_i = 1 - k \times s_i$
Li's L1-AVG algorithm	$\mathcal{H}(i,j) = \frac{1}{\lVert\mathcal{U}_j\rVert} \sum_{i\in\mathcal{U}_j} r_{ij} c_i$	$\frac{1}{\lVert\mathcal{I}_i\rVert} \sum_{j\in\mathcal{I}_i} \sqrt{s_{ij}}$	$1 - \lambda s_i$
Li's L2-AVG algorithm	Same as above	$\frac{1}{\lVert\mathcal{I}_i\rVert} \sum_{j\in\mathcal{I}_i} s_{ij}$	$1 - \frac{\lambda}{2} s_i$
Li's L1-MAX algorithm	Same as above	$\max_{j\in\mathcal{I}_i} \sqrt{s_{ij}}$	$1 - \lambda s_i$
Li's L2-MAX algorithm	Same as above	$\max_{j\in\mathcal{I}_i} s_{ij}$	$1 - \frac{\lambda}{2} s_i$
Li's L1-MIN algorithm	Same as above	$\min_{j\in\mathcal{I}_i} \sqrt{s_{ij}}$	$1 - \lambda s_i$
Li's L2-MIN algorithm	Same as above	$\min_{j\in\mathcal{I}_i} s_{ij}$	$1 - \frac{\lambda}{2} s_i$

Adaptability of the Framework

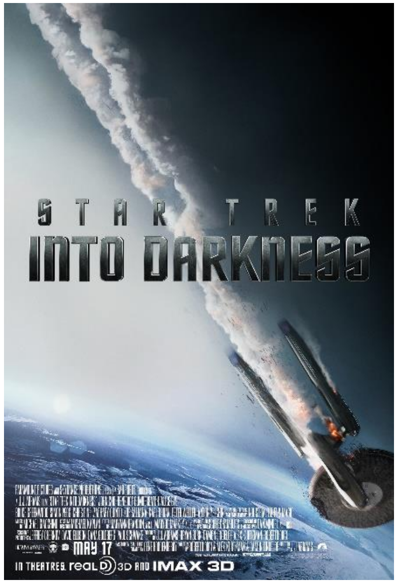
- As we can see, all the mentioned previous work can be captured as special cases of our framework
  - They all use “reputation weighted average” as the predictor (item centric model)
  - It naturally assumes that an item has an intrinsic quality

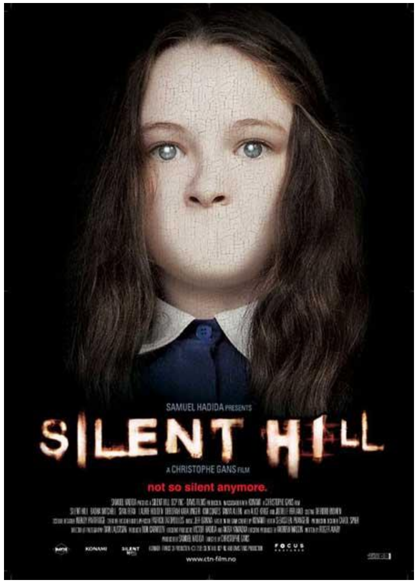
$$\mathcal{H}(i,j) = \sum_{i \in \mathcal{U}_j} c_i r_{ij} / \sum_{i \in \mathcal{U}_j} c_i$$

  - The intrinsic quality view may not suitable for all cases

Intrinsic View versus Taste View

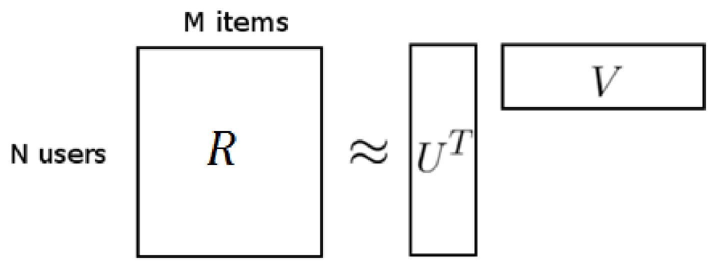
- Depending on the situation, taste view might be more appropriate





Reputation Estimation using Matrix Factorization

- We plug-in a well-studied personalized model as the prediction model
  - Low-rank matrix factorization model



$$\mathcal{L} = \frac{1}{2} \sum_{(i,j,r)\in\mathcal{Q}} (r - U_i^T V_j)^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2$$

Reputation Estimation using Matrix Factorization

- Penalty function

$$s_i = \frac{1}{\|\mathcal{I}_i\|} \sum_{j \in \mathcal{I}_i} s_{ij}$$

- Link function

$$c_i = 1 - s_i$$

We assume that the ratings have been mapped to $[0,1]$ as a pre-processing step. So that $s_i, c_i \in [0,1]$.

Experiments

- Dataset
  - There is no publicly available rating dataset with ground truth spammer label
  - We take MovieLens dataset as our base dataset
    - We simulate spam users' behavior using several spamming strategies
    - Data in MovieLens comes from an academic recommender system, it is more likely the users are not spam users

Experiments

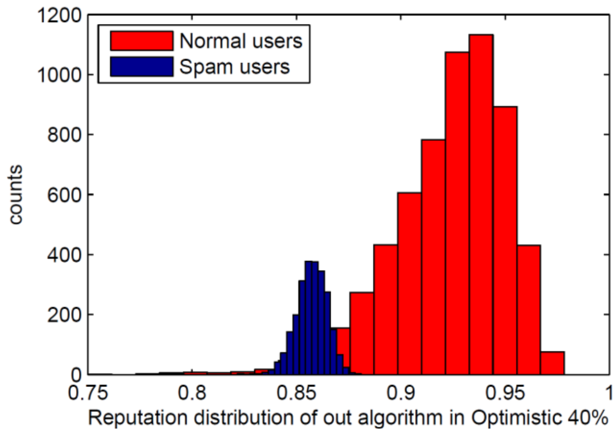
- Dataset
  - Spamming strategies
    - Random spamming
      - Random attacks
    - Semi-random spamming
      - Average attacks
    - Optimistic spamming
      - Bandwagon attacks
    - Pessimistic spamming
      - Nuke attacks
  - Spammer level
    - 10%, 20%, 30% and 40% (as to normal users)

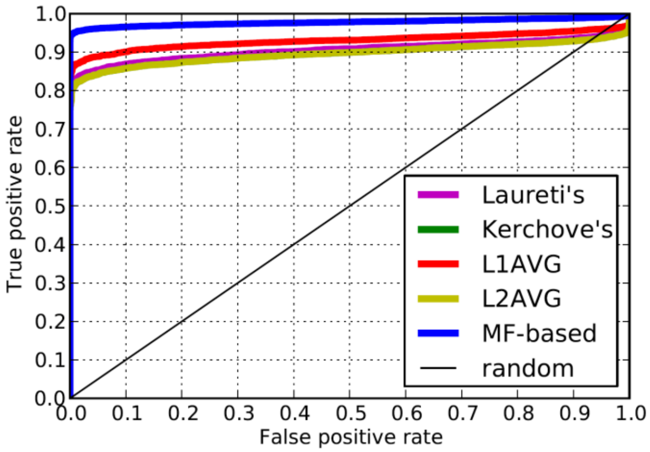
	i1	i2	i3	i4	i5	i6
Normal u1	5	3	4	3	1	4
Normal u2	4	3	5	5	1	5
Random	1	4	3	2	5	3
Semi-random	5	3	4	2	1	4
Optimistic	5	3	4	5	5	5
Pessimistic	5	3	4	1	1	1

Sample spamming data

Experiments

- Evaluation Methods
  - We use Area under the ROC Curve (AUC) to measure the performance





Experiments

- Results

Type	Random				Semi-random			
Percentage	10	20	30	40	10	20	30	40
Laureti's	0.9806	0.9803	0.9797	0.979	0.9241	0.924	0.9248	0.9248
Kerchove's	0.9793	0.9791	0.9785	0.9777	0.9227	0.9231	0.9239	0.9239
L1-AVG	0.9791	0.9789	0.978	0.9769	0.9098	0.9111	0.9118	0.9115
L2-AVG	0.979	0.9788	0.9782	0.9773	0.9224	0.9228	0.9237	0.9237
MF-based	**0.9893**	**0.9896**	**0.9896**	**0.9892**	**0.9685**	**0.9676**	**0.9673**	**0.9668**
Improvement	**0.89%**	**0.95%**	**1.01%**	**1.04%**	**4.80%**	**4.72%**	**4.60%**	**4.54%**

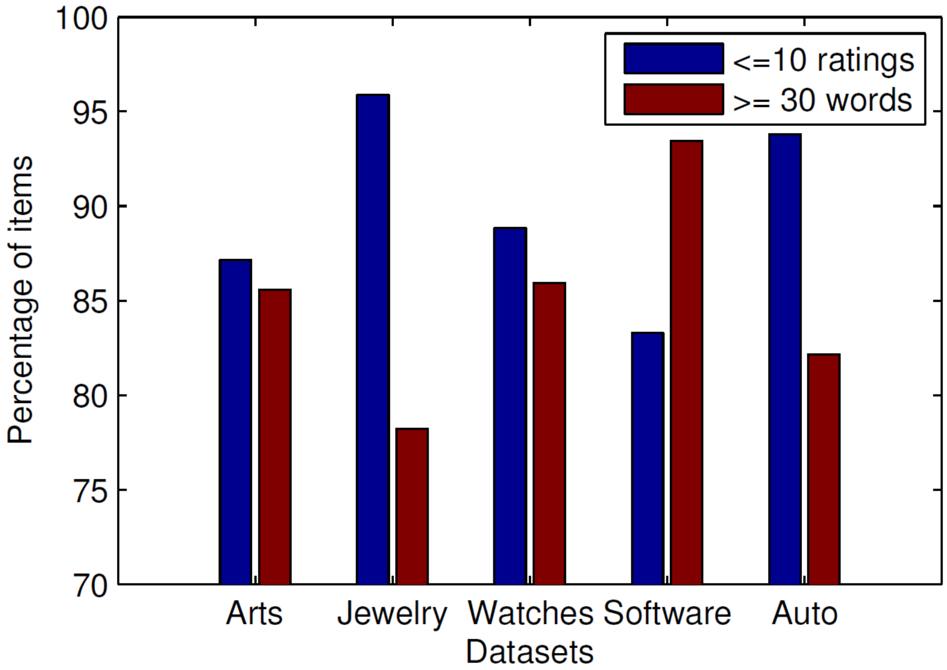
Type	Optimistic				Pessimistic			
Percentage	10	20	30	40	10	20	30	40
Laureti's	0.9464	0.9298	0.9166	0.9047	0.9926	0.9914	0.9902	0.9887
Kerchove's	0.9428	0.9234	0.909	0.896	0.991	0.9885	0.9858	0.9829
L1-AVG	0.9578	0.9465	0.9376	0.9295	0.99	0.9875	0.9847	0.9817
L2-AVG	0.9425	0.9231	0.9088	0.8959	0.9902	0.9873	0.9841	0.9807
MF-based	**0.9884**	**0.9858**	**0.9814**	**0.9774**	**0.9939**	**0.9938**	**0.9937**	**0.9936**
Improvement	**3.19%**	**4.15%**	**4.67%**	**5.15%**	**0.13%**	**0.24%**	**0.35%**	**0.50%**

Outline

- Introduction and Background Review
- Online Collaborative Filtering
- Response Aware Collaborative Filtering
- User Reputation Estimation
- Combine Ratings with Reviews
- Conclusion

Cold-start Problem

- Cold-start problem
  - Recommender system has too little information concerning a user or an item to make accurate predictions
  - Severe problems in real system



Reasons for Recommendation

Reasons for Recommendation

- Why these items are recommended?
- Explanations on why such items are recommended can be useful.
- Existing recommender systems do not provide adequate explanations.

Reasons for Recommendation

- Why these items are recommended?
- Explanations on why such items are recommended can be useful.
- Existing recommender systems do not provide adequate explanations.

Reasons for Recommendation

- Why these items are recommended?
- Explanations on why such items are recommended can be
- Existing ovide adequa

Reasons for Recommendation

- Why these items are recommended?
- Explanations on why such items are recommended can be
- Existin vide adequa

Reasons for Recommendation

- Why these items are recommended?
- Explanations on why such items are recommended can be
- Existing vide adequa

Reasons for Recommendation

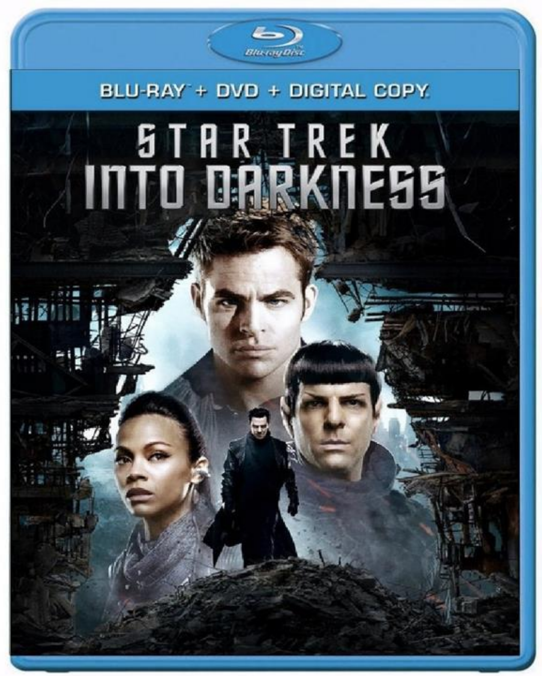
- Why these items are recommended?
- Explanations on why such items are recommended can be useful.
- Existing recommender systems do not provide adequate explanations.

Reasons for Recommendation

- Why these items are recommended?
- Explanations on why such items are recommended can be useful.
- Existing recommender systems do not provide adequate explanations.

Reviews Can Help

Reviews Can Help



Reviews Can Help

★★★★★ **Did not disappoint life-long Trekkie**

By Emily Eagon on July 27, 2013

Format: DVD

I had a major argument with a fellow Trekkie about the merits of this film. He continued to argue that the movie was good until the end, in which case it was a cop out of something that had already been done before (those who have seen other Star Trek motion pictures know what I'm talking about. Being sensitive to spoilers) This was my argument:

Yes it does mirror some previously established Star Trek plots, but the twists that accompanied the mirages are COMPLETELY important to what makes this film unique. The changes that were made to story lines from the original series completely change the way that the characters react and open them up to future discoveries that could not have been made in the original series (I'm mostly referring to Spock's emotional availability)

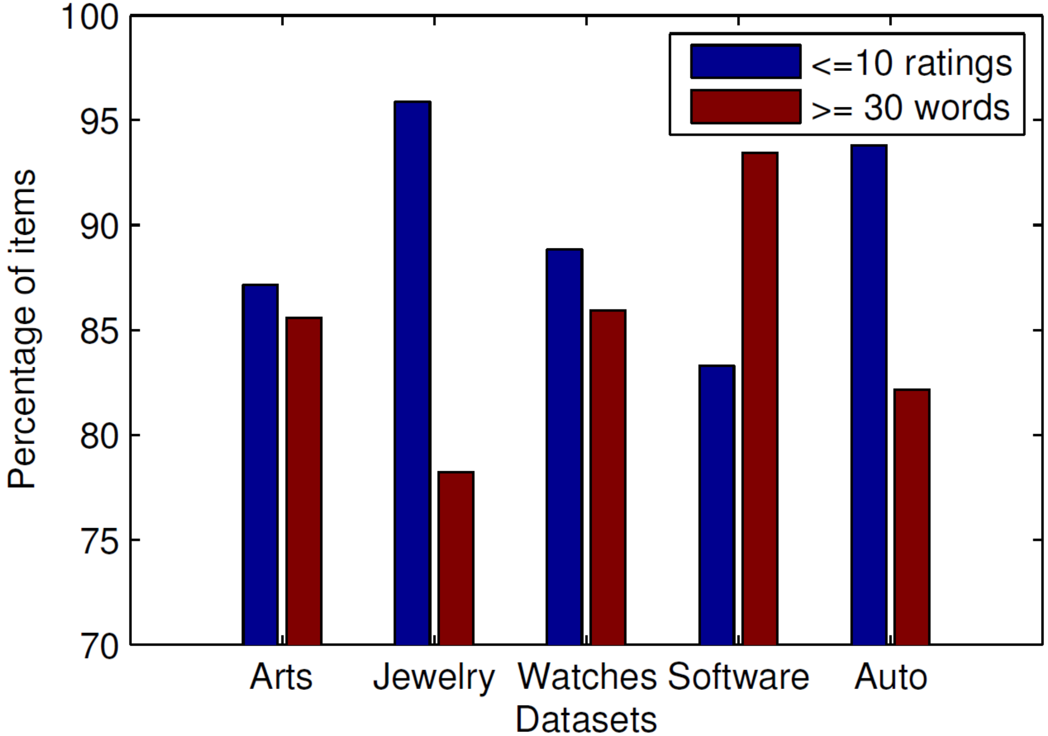
Even in the tiniest details it connects to the original series, down to the Tribbles, making any Trekkie feel right at home for the majority of the movie. The film was filled with the sass, wit, and banter that the characters in this show are known for and keep the audience on their toes with the surprises built in.

Maybe one or two other times in my life have I wanted to stand up in the theater or my living room (or wherever I was watching whatever I was watching) and root for a character so badly. The line from the trailer sums it all up. "Is there anything you would not do for your family?" This movie shows exactly how much of a family they truly are and I could not have been happier with this film.

By the way I NEVER see a movie multiple times in theaters due to the obscene prices, but I was willing to go three times to see this film, if that tells you anything.

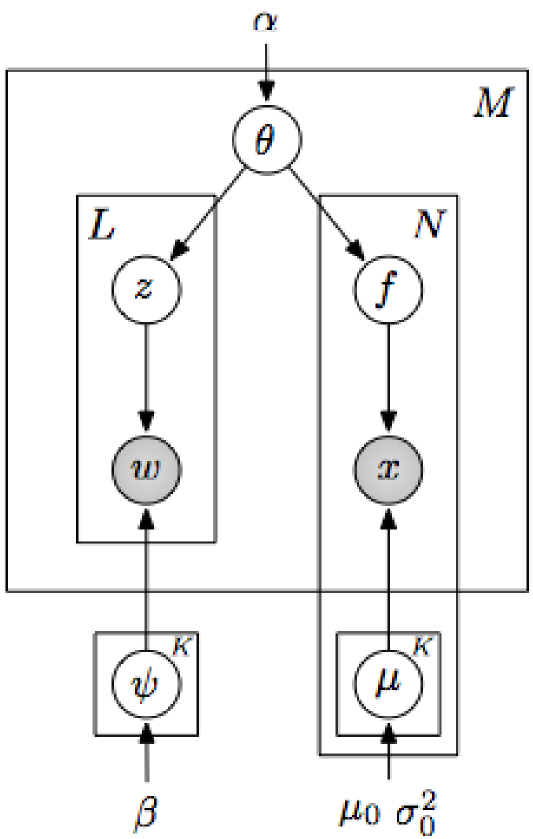
Reviews Can Help

Reviews Can Help



Ratings Meet Reviews, A Combined Approach to Recommend

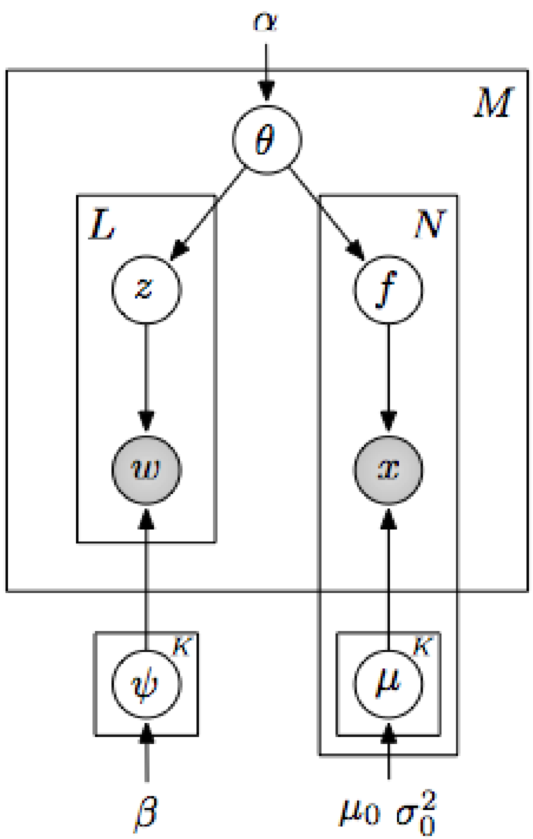
- Our model, RMR
  - Use mixture of Gaussians rather than matrix factorization to model ratings
  - Use LDA to model reviews
  - Combine ratings and reviews by sharing the same topic distribution



Ratings Meet Reviews

- Generative Process:

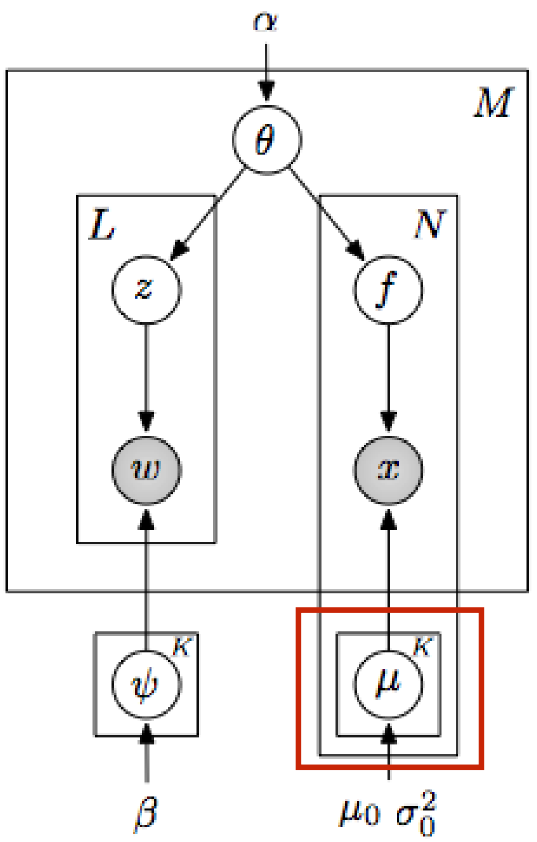
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

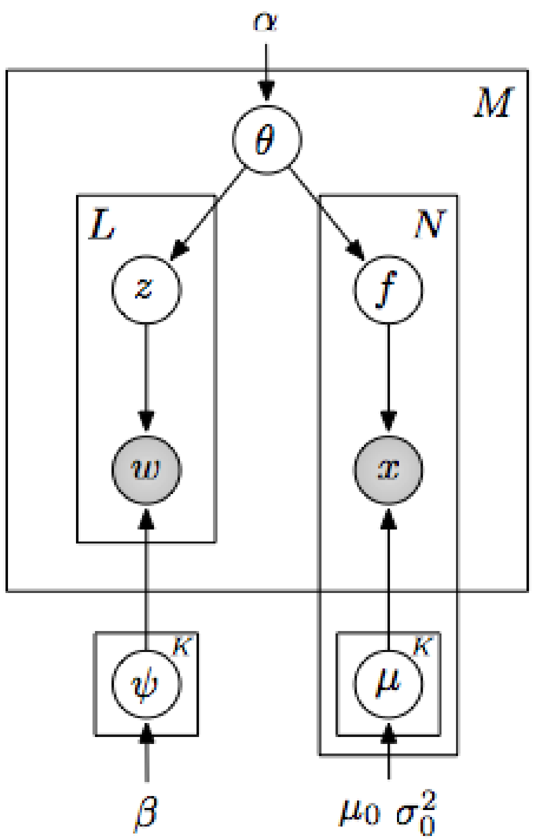
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

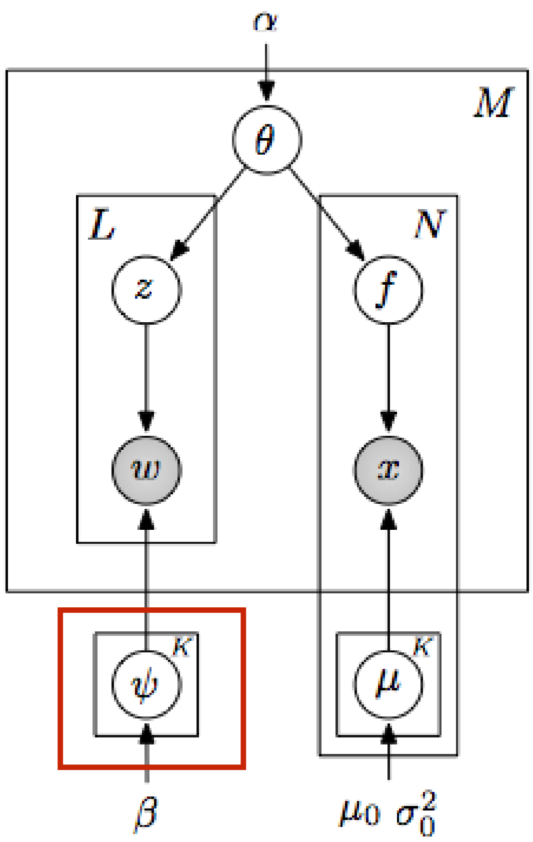
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

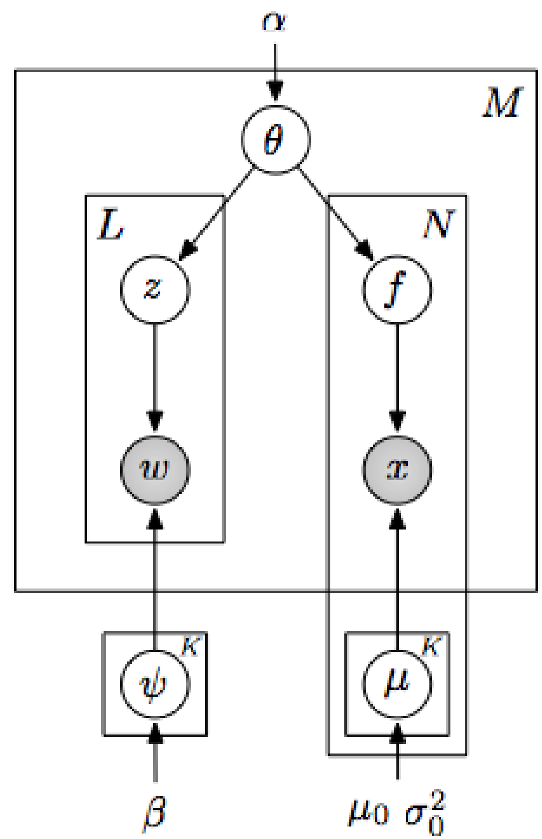
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

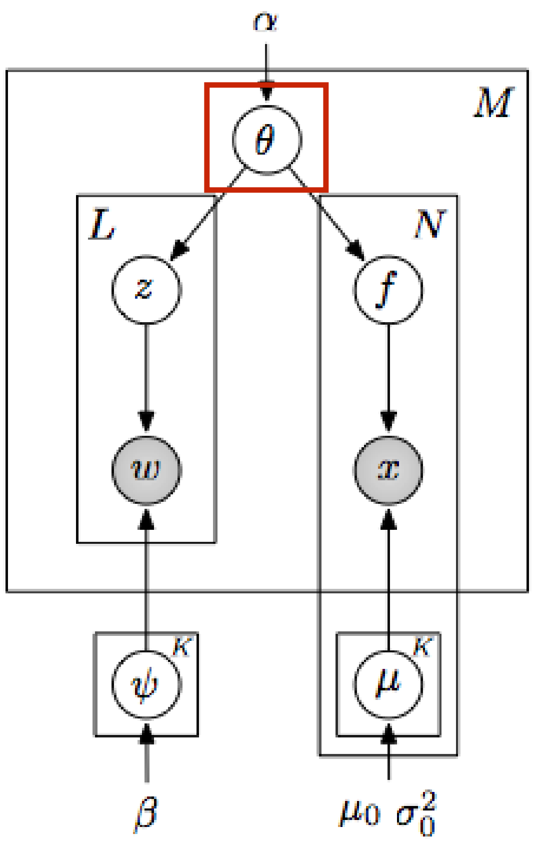
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

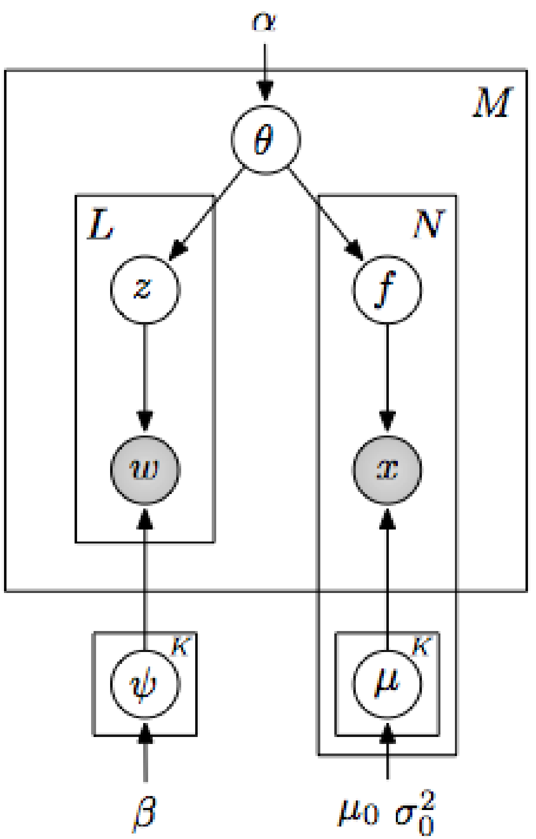
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim \text{Gaussian}(\mu_0, \sigma_0^2)$
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim \text{Dirichlet}(\beta)$
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim \text{Dirichlet}(\alpha)$
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim \text{Multinomial}(\theta_v)$
      ii. Draw word $w_{v,n} \sim \text{Multinomial}(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim \text{Multinomial}(\theta_v)$
      ii. Draw the rating $x_{v,u} \sim \text{Gaussian}(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

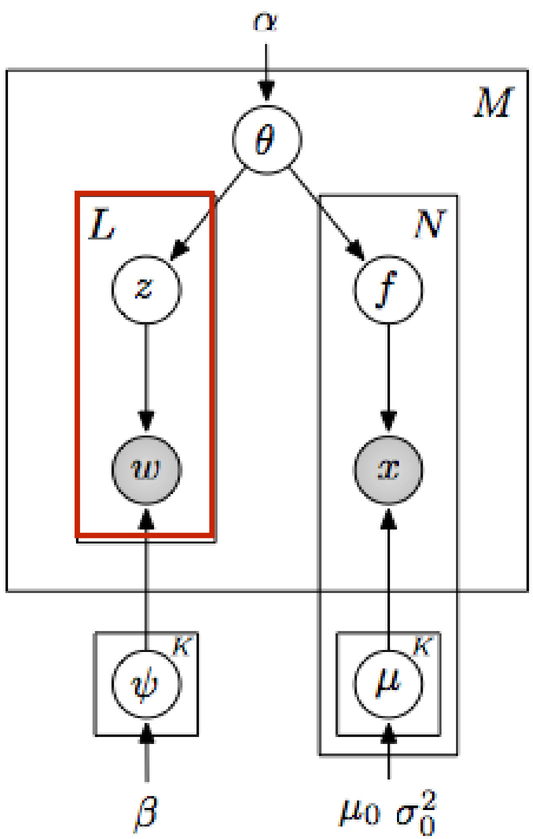
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

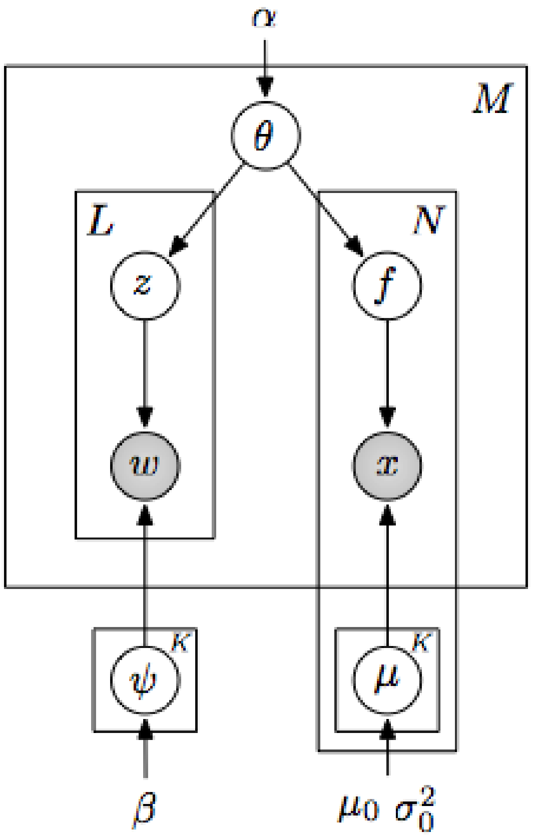
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim \text{Gaussian}(\mu_0, \sigma_0^2)$
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim \text{Dirichlet}(\beta)$
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim \text{Dirichlet}(\alpha)$
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim \text{Multinomial}(\theta_v)$
      ii. Draw word $w_{v,n} \sim \text{Multinomial}(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim \text{Multinomial}(\theta_v)$
      ii. Draw the rating $x_{v,u} \sim \text{Gaussian}(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

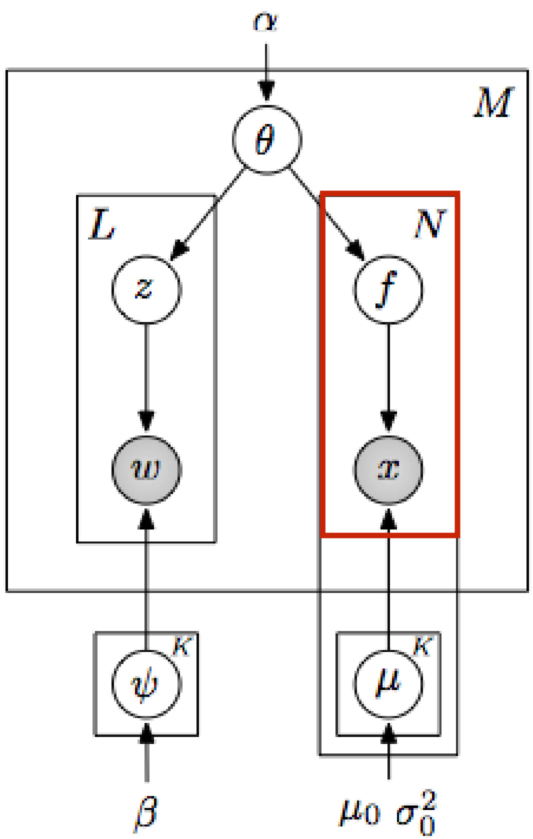
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.



Ratings Meet Reviews

- Generative Process:

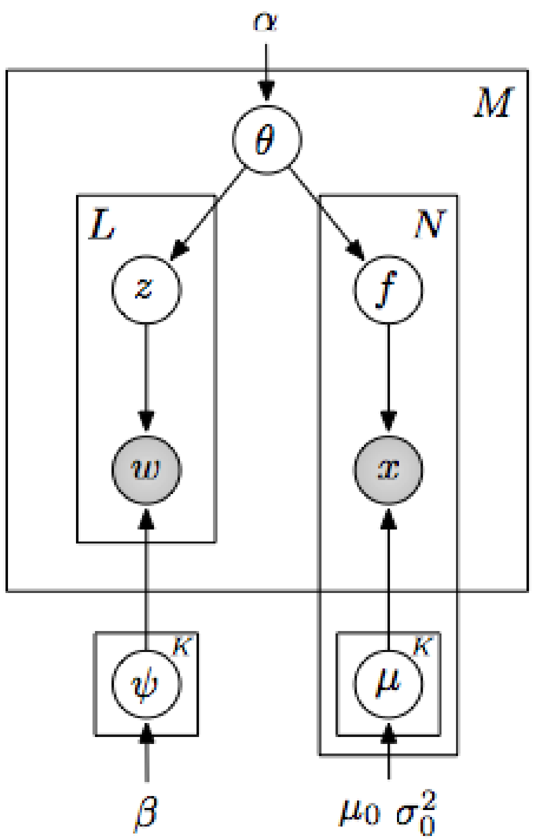
1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim \text{Gaussian}(\mu_0, \sigma_0^2)$
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim \text{Dirichlet}(\beta)$
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim \text{Dirichlet}(\alpha)$
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim \text{Multinomial}(\theta_v)$
      ii. Draw word $w_{v,n} \sim \text{Multinomial}(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim \text{Multinomial}(\theta_v)$
      ii. Draw the rating $x_{v,u} \sim \text{Gaussian}(\mu_{u,f_{v,u}}, \sigma^2)$.



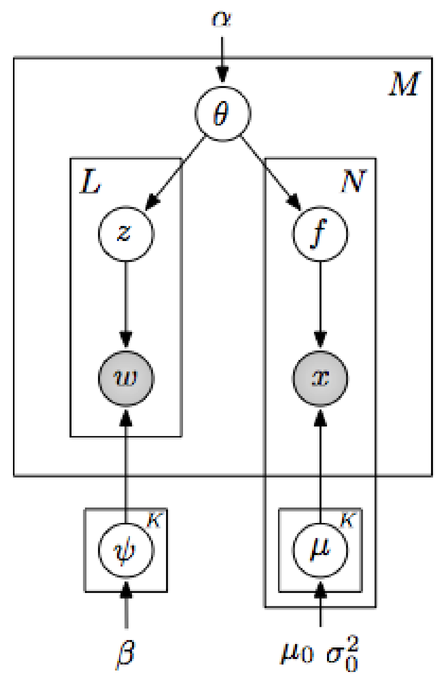
Ratings Meet Reviews

- Generative Process:

1. For each user $u \in \mathcal{U}$:
   (a) For each latent topic dimension $k \in [1, K]$:
      i. Draw $\mu_{u,k} \sim$ Gaussian(μ_0, σ_0^2)
2. For each latent topic dimension $k \in [1, K]$:
   (a) Draw $\psi_k \sim$ Dirichlet(β)
3. For each item $v \in \mathcal{V}$:
   (a) Draw topic mixture proportion $\theta_v \sim$ Dirichlet(α)
   (b) For each description word $w_{v,n}$:
      i. Draw topic assignment $z_{v,n} \sim$ Multinomial(θ_v)
      ii. Draw word $w_{v,n} \sim$ Multinomial$(\psi_{z_{v,n}})$
   (c) For each observed rating assigned by u to v:
      i. Draw topic assignment $f_{v,u} \sim$ Multinomial(θ_v)
      ii. Draw the rating $x_{v,u} \sim$ Gaussian$(\mu_{u,f_{v,u}}, \sigma^2)$.

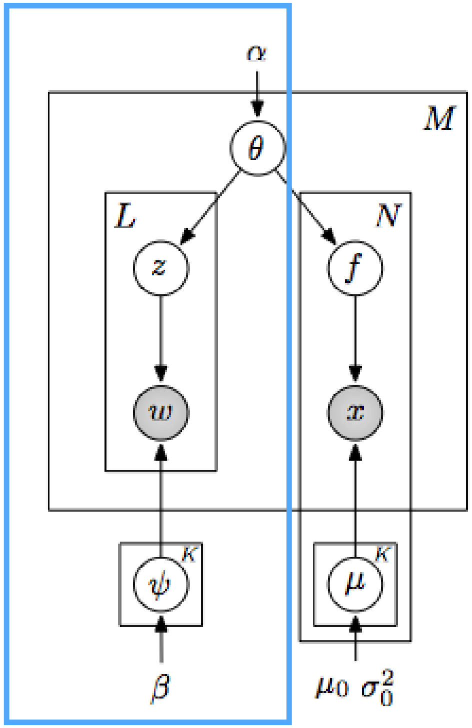


Ratings Meet Reviews



$$P(\mathbf{w},\mathbf{x}|\Theta;\alpha,\beta,\mu_0,\sigma_0^2,\sigma^2) \propto \prod_{j=1}^{M} P(\theta_j|\alpha) \prod_{i\in\mathcal{U}_j} \left(\prod_{l=1}^{L_{i,j}} \sum_{z=1}^{K} P(z|\theta_j) P(w_l|\psi_z) \right) \left(\sum_{f=1}^{K} P(f|\theta_j) P(x_{i,j}|\mu_{i,f},\sigma^2) \right)$$

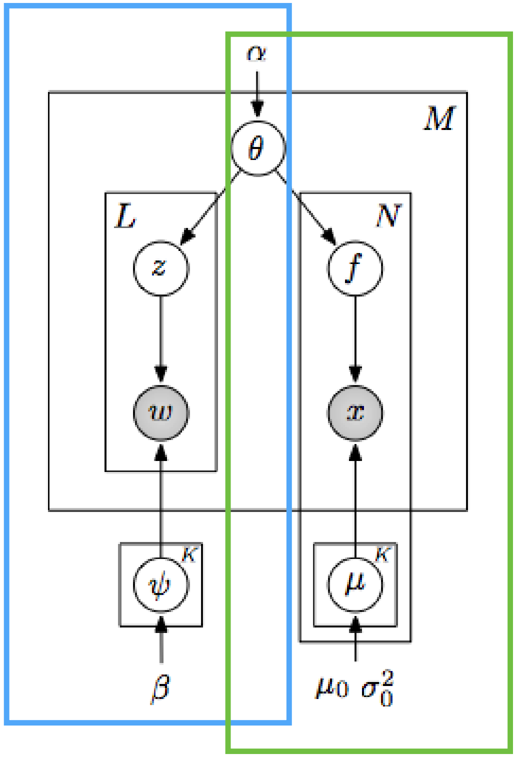
Ratings Meet Reviews



Use LDA to model the reviews

$$P(\mathbf{w},\mathbf{x}|\Theta;\alpha,\beta,\mu_0,\sigma_0^2,\sigma^2) \propto \prod_{j=1}^{M} P(\theta_j|\alpha) \prod_{i\in\mathcal{U}_j} \left(\prod_{l=1}^{L_{i,j}} \sum_{z=1}^{K} P(z|\theta_j)P(w_l|\psi_z) \right) \left(\sum_{f=1}^{K} P(f|\theta_j)P(x_{i,j}|\mu_{i,f},\sigma^2) \right)$$

Ratings Meet Reviews



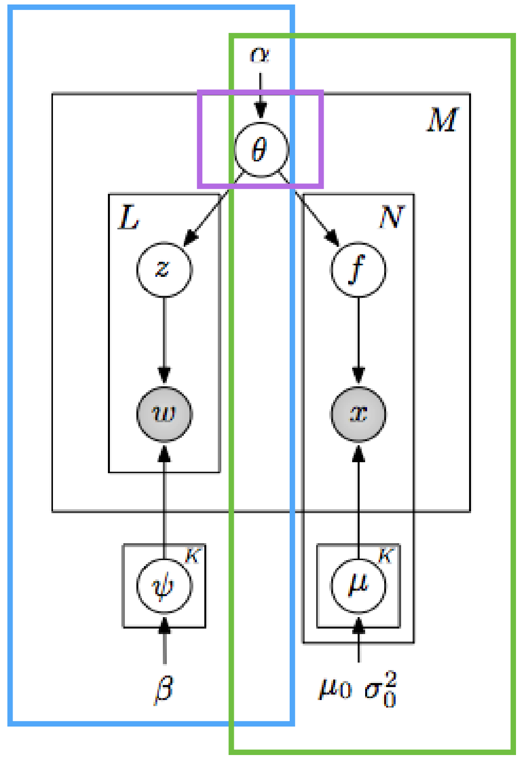
Use LDA to model the reviews

Use mixture of Gaussians to model the ratings

$$P(\mathbf{w}, \mathbf{x} | \Theta; \alpha, \beta, \mu_0, \sigma_0^2, \sigma^2) \propto \prod_{j=1}^{M} P(\theta_j | \alpha) \prod_{i \in \mathcal{U}_j} \left(\prod_{l=1}^{L_{i,j}} \sum_{z=1}^{K} P(z|\theta_j) P(w_l|\psi_z) \right) \left(\sum_{f=1}^{K} P(f|\theta_j) P(x_{i,j}|\mu_{i,f}, \sigma^2) \right)$$

Ratings Meet Reviews

Use LDA to model the reviews



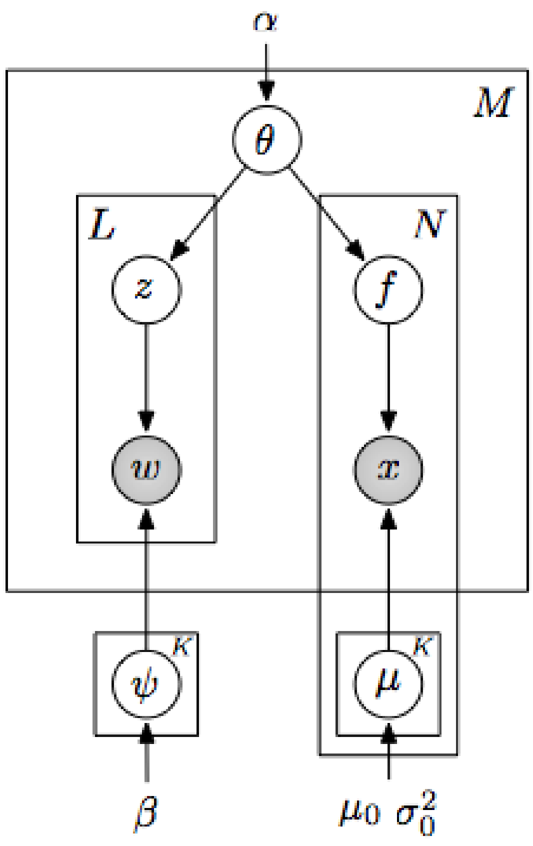
Use the same topic distribution to connect rating part and review part

Use mixture of Gaussians to model the ratings

$$P(\mathbf{w}, \mathbf{x} | \Theta; \alpha, \beta, \mu_0, \sigma_0^2, \sigma^2) \propto \prod_{j=1}^{M} P(\theta_j|\alpha) \prod_{i \in \mathcal{U}_j} \left(\prod_{l=1}^{L_{i,j}} \sum_{z=1}^{K} P(z|\theta_j) P(w_l|\psi_z) \right) \left(\sum_{f=1}^{K} P(f|\theta_j) P(x_{i,j}|\mu_{i,f}, \sigma^2) \right)$$

Ratings Meet Reviews

- We developed Collapsed Gibbs Sampler for RMR
- Space Complexity $O((M + N + V) \times K)$
- Time Complexity $O(K)$



Experiments

- How RMR performs compared with other models?
- How can “cold-start” items/users benefit from the incorporation of reviews?
- Can we learn interpretable latent topics?

Categories	Users	Items	Ratings	Comment words
27	6,643,669	2,441,053	34,686,880	4,053,795,667

Statistics of Amazon dataset

Experiments

Dataset	a MF	b LDAMF	c CTR	d HFT	e RMR	Improvement of RMR versus min(a,b)	c	d
Arts	1.565 (0.04)	1.575 (0.04)	1.471 (0.04)	1.390 (0.04)	**1.371 (0.04)**	14.15%	7.29%	1.39%
Jewelry	1.257 (0.03)	1.279 (0.03)	1.206 (0.03)	1.177 (0.02)	**1.160 (0.02)**	8.36%	3.97%	1.47%
Industrial Scientific	0.461 (0.02)	0.462 (0.02)	0.382 (0.02)	**0.359 (0.02)**	0.362 (0.02)	27.35%	5.52%	-0.83%
Watches	1.535 (0.03)	1.518 (0.03)	1.491 (0.03)	1.488 (0.03)	**1.458 (0.02)**	4.12%	2.26%	2.06%
Cell Phones and Accessories	2.230 (0.04)	2.308 (0.04)	2.177 (0.04)	2.135 (0.03)	**2.085 (0.03)**	6.95%	4.41%	2.40%
Musical Instruments	1.506 (0.02)	1.520 (0.02)	1.422 (0.02)	1.395 (0.02)	**1.374 (0.02)**	9.61%	3.49%	1.53%
Software	2.409 (0.02)	2.214 (0.02)	2.254 (0.02)	2.219 (0.02)	**2.173 (0.02)**	1.89%	3.73%	2.12%
Gourmet Foods	1.515 (0.01)	1.491 (0.01)	1.482 (0.01)	**1.457 (0.01)**	1.465 (0.01)	1.77%	1.16%	-0.55%
Office Products	1.814 (0.01)	1.796 (0.01)	1.733 (0.01)	1.669 (0.01)	**1.638 (0.01)**	9.65%	5.80%	1.89%
Automotive	1.570 (0.01)	1.585 (0.01)	1.492 (0.01)	1.432 (0.01)	**1.403 (0.01)**	11.90%	6.34%	2.07%
Patio	1.771 (0.01)	1.793 (0.01)	1.720 (0.01)	1.698 (0.01)	**1.669 (0.01)**	6.11%	3.06%	1.74%
Pet Supplies	1.700 (0.01)	1.700 (0.01)	1.613 (0.01)	1.583 (0.01)	**1.562 (0.01)**	8.83%	3.27%	1.34%
Beauty	1.399 (0.01)	1.414 (0.01)	1.361 (0.01)	1.358 (0.01)	**1.334 (0.01)**	4.87%	2.02%	1.80%
Shoes	0.305 (0.00)	0.335 (0.00)	0.271 (0.00)	**0.247 (0.00)**	0.251 (0.00)	21.51%	7.97%	-1.59%
Kindle Store	1.553 (0.01)	1.561 (0.01)	1.457 (0.01)	1.437 (0.01)	**1.412 (0.01)**	9.99%	3.19%	1.77%
Clothing and Accessories	0.393 (0.00)	0.406 (0.00)	0.355 (0.00)	0.349 (0.00)	**0.336 (0.00)**	16.96%	5.65%	3.87%
Health	1.615 (0.01)	1.608 (0.01)	1.552 (0.01)	1.538 (0.01)	**1.512 (0.01)**	6.35%	2.65%	1.72%
Toys and Games	1.467 (0.01)	1.395 (0.01)	1.389 (0.01)	**1.370 (0.01)**	1.372 (0.01)	1.68%	1.24%	-0.15%
Tools and Home Improvement	1.600 (0.01)	1.610 (0.01)	1.513 (0.01)	1.510 (0.01)	**1.491 (0.01)**	7.31%	1.48%	1.27%
Sports and Outdoors	1.219 (0.01)	1.223 (0.01)	1.150 (0.01)	1.138 (0.01)	**1.129 (0.01)**	7.97%	1.86%	0.80%
Video Games	1.610 (0.01)	1.608 (0.01)	1.572 (0.01)	1.528 (0.01)	**1.510 (0.01)**	6.49%	4.11%	1.19%
Home and Kitchen	1.628 (0.05)	1.610 (0.05)	1.577 (0.05)	1.531 (0.04)	**1.501 (0.04)**	7.26%	5.06%	2.00%
Amazon Instant Video	1.330 (0.01)	1.328 (0.01)	1.291 (0.01)	**1.260 (0.01)**	1.270 (0.01)	4.57%	1.65%	-0.79%
Electronics	1.828 (0.00)	1.823 (0.00)	1.764 (0.00)	**1.722 (0.00)**	**1.722 (0.00)**	5.87%	2.44%	0.00%
Music	**0.956 (0.00)**	0.958 (0.00)	0.959 (0.00)	0.980 (0.00)	0.959 (0.00)	-0.31%	0.00%	2.19%
Movies and TV	1.119 (0.00)	1.117 (0.00)	**1.114 (0.00)**	1.119 (0.00)	1.120 (0.00)	-0.27%	-0.54%	-0.09%
Books	1.107 (0.00)	1.109 (0.00)	**1.106 (0.00)**	1.138 (0.00)	1.113 (0.00)	-0.54%	-0.63%	2.25%
Average on all datasets						7.79%	3.28%	1.22%

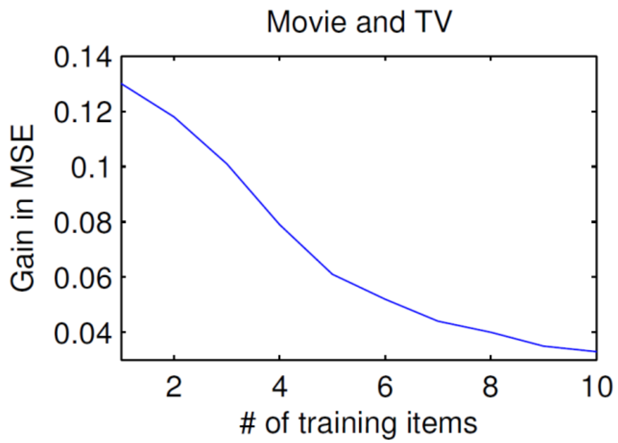
Experiments

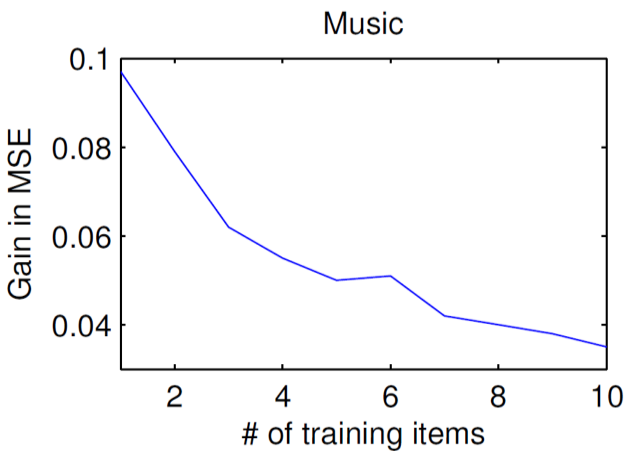
- Performs the best on 19 out of 27 categories
- Performs better on 26 out of 27 datasets compared with matrix factorization
- On average, improve 7.8% over MF, 3.3% over CTR and 1.2% over HFT

	a	b	c	d	e	Improvement of RMR versus		
Dataset	MF	LDAMF	CTR	HFT	RMR	min(a,b)	c	d
Arts	1.565 (0.04)	1.575 (0.04)	1.471 (0.04)	1.390 (0.04)	**1.371 (0.04)**	14.15%	7.29%	1.39%
Jewelry	1.257 (0.03)	1.279 (0.03)	1.206 (0.03)	1.177 (0.02)	**1.160 (0.02)**	8.36%	3.97%	1.47%
Industrial Scientific	0.461 (0.02)	0.462 (0.02)	0.382 (0.02)	**0.359 (0.02)**	0.362 (0.02)	27.35%	5.52%	-0.83%
Watches	1.535 (0.03)	1.518 (0.03)	1.491 (0.03)	1.488 (0.03)	**1.458 (0.02)**	4.12%	2.26%	2.06%
Cell Phones and Accessories	2.230 (0.04)	2.308 (0.04)	2.177 (0.04)	2.135 (0.03)	**2.085 (0.03)**	6.95%	4.41%	2.40%
Musical Instruments	1.506 (0.02)	1.520 (0.02)	1.422 (0.02)	1.395 (0.02)	**1.374 (0.02)**	9.61%	3.49%	1.53%
Software	2.409 (0.02)	2.214 (0.02)	2.254 (0.02)	2.219 (0.02)	**2.173 (0.02)**	1.89%	3.73%	2.12%
Gourmet Foods	1.515 (0.01)	1.491 (0.01)	1.482 (0.01)	**1.457 (0.01)**	1.465 (0.01)	1.77%	1.16%	-0.55%
Office Products	1.814 (0.01)	1.796 (0.01)	1.733 (0.01)	1.669 (0.01)	**1.638 (0.01)**	9.65%	5.80%	1.89%
Automotive	1.570 (0.01)	1.585 (0.01)	1.492 (0.01)	1.432 (0.01)	**1.403 (0.01)**	11.90%	6.34%	2.07%
Patio	1.771 (0.01)	1.793 (0.01)	1.720 (0.01)	1.698 (0.01)	**1.669 (0.01)**	6.11%	3.06%	1.74%
Pet Supplies	1.700 (0.01)	1.700 (0.01)	1.613 (0.01)	1.583 (0.01)	**1.562 (0.01)**	8.83%	3.27%	1.34%
Beauty	1.399 (0.01)	1.414 (0.01)	1.361 (0.01)	1.358 (0.01)	**1.334 (0.01)**	4.87%	2.02%	1.80%
Shoes	0.305 (0.00)	0.335 (0.00)	0.271 (0.00)	**0.247 (0.00)**	0.251 (0.00)	21.51%	7.97%	-1.59%
Kindle Store	1.553 (0.01)	1.561 (0.01)	1.457 (0.01)	1.437 (0.01)	**1.412 (0.01)**	9.99%	3.19%	1.77%
Clothing and Accessories	0.393 (0.00)	0.406 (0.00)	0.355 (0.00)	0.349 (0.00)	**0.336 (0.00)**	16.96%	5.65%	3.87%
Health	1.615 (0.01)	1.608 (0.01)	1.552 (0.01)	1.538 (0.01)	**1.512 (0.01)**	6.35%	2.65%	1.72%
Toys and Games	1.467 (0.01)	1.395 (0.01)	1.389 (0.01)	**1.370 (0.01)**	1.372 (0.01)	1.68%	1.24%	-0.15%
Tools and Home Improvement	1.600 (0.01)	1.610 (0.01)	1.513 (0.01)	1.510 (0.01)	**1.491 (0.01)**	7.31%	1.48%	1.27%
Sports and Outdoors	1.219 (0.01)	1.223 (0.01)	1.150 (0.01)	1.138 (0.01)	**1.129 (0.01)**	7.97%	1.86%	0.80%
Video Games	1.610 (0.01)	1.608 (0.01)	1.572 (0.01)	1.528 (0.01)	**1.510 (0.01)**	6.49%	4.11%	1.19%
Home and Kitchen	1.628 (0.05)	1.610 (0.05)	1.577 (0.05)	1.531 (0.04)	**1.501 (0.04)**	7.26%	5.06%	2.00%
Amazon Instant Video	1.330 (0.01)	1.328 (0.01)	1.291 (0.01)	**1.260 (0.01)**	1.270 (0.01)	4.57%	1.65%	-0.79%
Electronics	1.828 (0.00)	1.823 (0.00)	1.764 (0.00)	**1.722 (0.00)**	**1.722 (0.00)**	5.87%	2.44%	0.00%
Music	**0.956 (0.00)**	0.958 (0.00)	0.959 (0.00)	0.980 (0.00)	0.959 (0.00)	-0.31%	0.00%	2.19%
Movies and TV	1.119 (0.00)	1.117 (0.00)	**1.114 (0.00)**	1.119 (0.00)	1.120 (0.00)	-0.27%	-0.54%	-0.09%
Books	1.107 (0.00)	1.109 (0.00)	**1.106 (0.00)**	1.138 (0.00)	1.113 (0.00)	-0.54%	-0.63%	2.25%
Average on all datasets						7.79%	3.28%	1.22%

Experiments

- Cold-start Settings
  - Items with fewer ratings gain more from the reviews





Experiments

- Interpretability
  - We recommend "Star Trek" to you because you are interested in "batman, effects, alien, harry, matrix, edition"

Top words in category Software

roxio	quicken	leopard	office	suse
contacted	son	os	excel	accounts
perfect	pick	parallels	2007	2004
burning	given	apple	student	nav
dvds	spanish	turbo	activation	federal
care	starting	tiger	microsoft	symantec

Top words in category Movie & TV

workout	season	batman	disney	godzilla
yoga	match	effects	christmas	hitchcock
workouts	episodes	alien	animation	kidman
videos	seasons	harry	kids	murder
exercises	vs	matrix	shrek	densel
cardio	episode	edition	animated	nicole

Overview

- Introduction and Background Review
- Online Collaborative Filtering
- Response Aware Collaborative Filtering
- User Reputation Estimation
- Combine Ratings with Reviews
- Conclusion

Conclusion

- We propose methods to improve recommender systems
  - Online learning algorithms
    - Bridge the gap between real system and experiments
    - Scale to large datasets
    - Incorporate new users or items effortlessly
  - Response aware PMF
    - Drop unrealistic assumptions
    - Improve prediction accuracy

Conclusion

- We propose methods to improve recommender systems
  - Reputation estimation methods
    - Propose general extensible framework
    - Propose matrix factorization based methods
    - Show better discrimination ability
  - Combine ratings with reviews
    - Utilize review data to alleviate cold-start problem
    - Tag latent dimension with words to produce reasons for recommendation

Questions?