Effective Fusion-based Approaches for Recommender Systems

Xin Xin

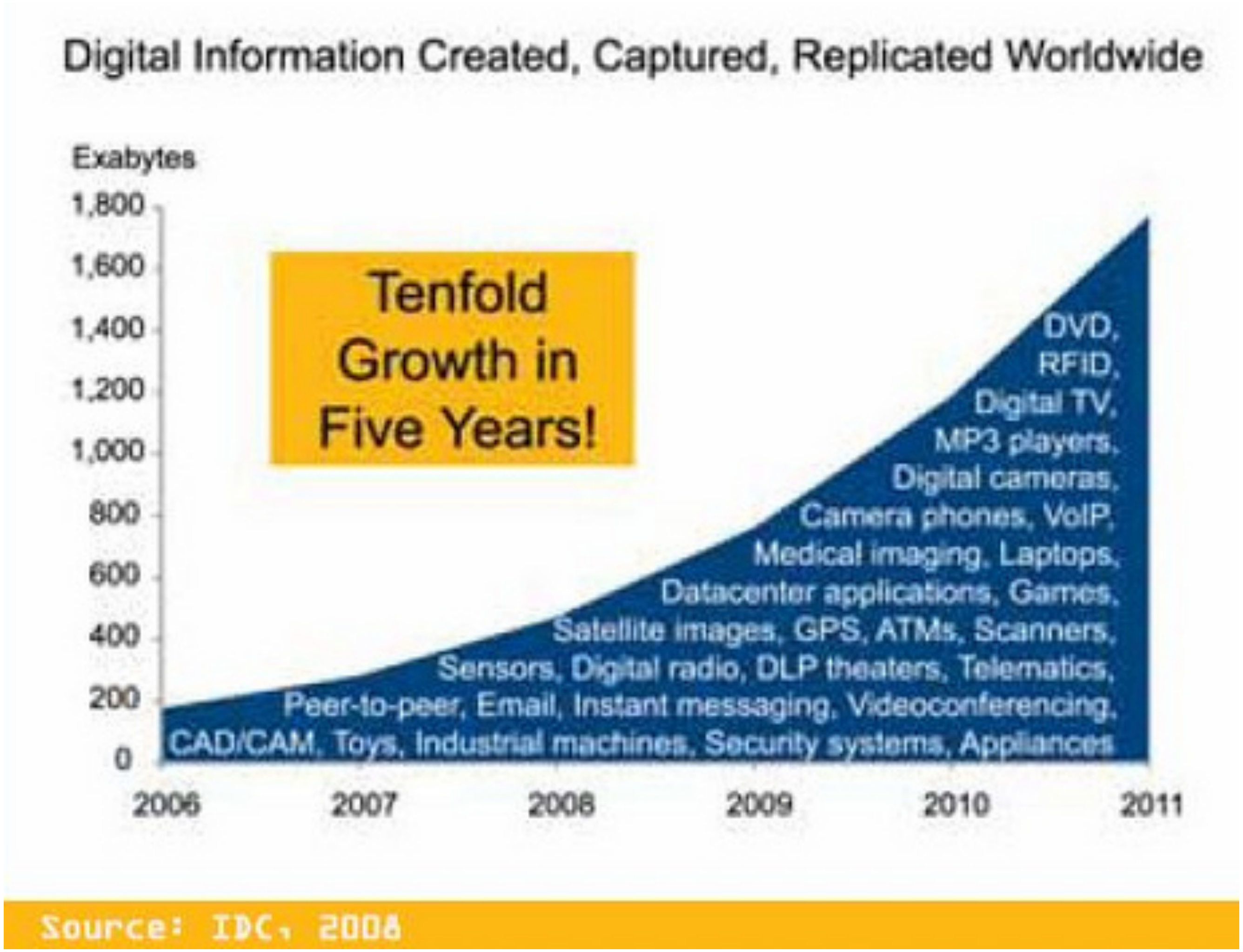
Supervisors: Prof. King and Prof. Lyu
Thesis Committees: Prof. Lam, Prof. Lui, and Prof. Yang

Department of Computer Science and Engineering
The Chinese University of Hong Kong
July 8, 2011

Outline

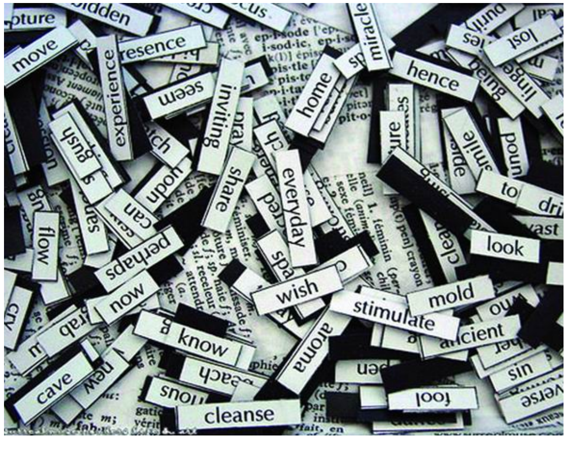
- Background of Recommender Systems
- Motivation of the Thesis
- Part 1: Relational ***Fusion*** of Multiple Features
- Part 2: Effective ***Fusion*** of Regression and Ranking
- Part 3: Effective ***Fusion*** of Quality and Relevance
- Part 4: Impression Efficiency Optimization
- Conclusion

Exponential Increase of Information



Information Overload

- Too much information
- Noises



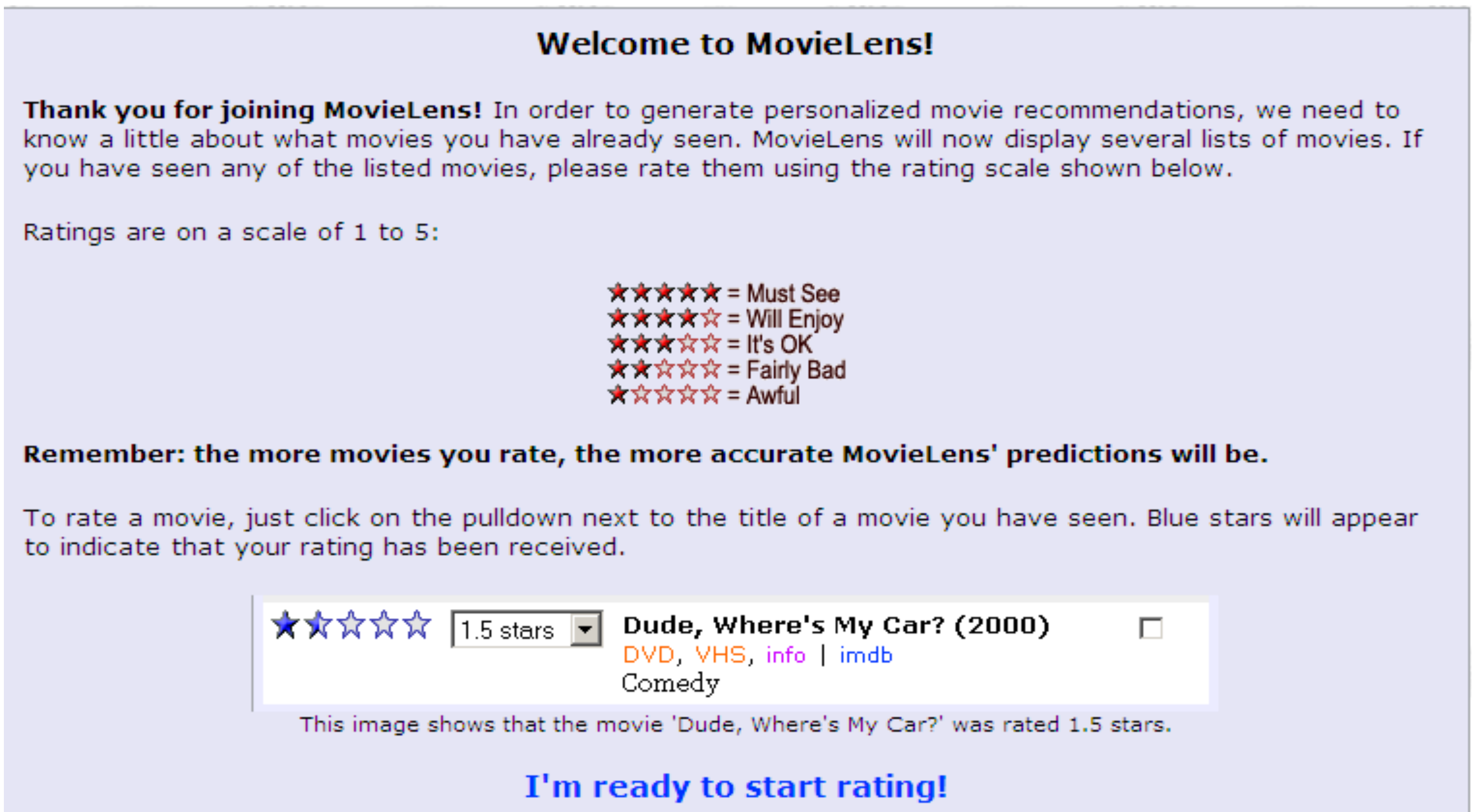
Recommender Systems

- To filter useful information for users

– Movie recommendation from MovieLens

Ratings in Recommender Systems

- Ratings
  - Recommendation results quality evaluation



Classical Regression Problem

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	y_{11}	y_{12}	r_{13}	y_{14}	y_{15}	y_{16}	y_{17}
u_2	y_{21}	y_{22}	y_{23}	y_{24}	r_{25}	y_{26}	y_{27}
u_3	y_{31}	y_{32}	y_{33}	y_{34}	y_{35}	y_{36}	r_{37}
u_4	y_{41}	y_{42}	r_{43}	y_{44}	y_{45}	y_{46}	r_{47}

Figure. User-item matrix in recommender systems

- Task: predict unrated user-item pairs

An Overview of Techniques

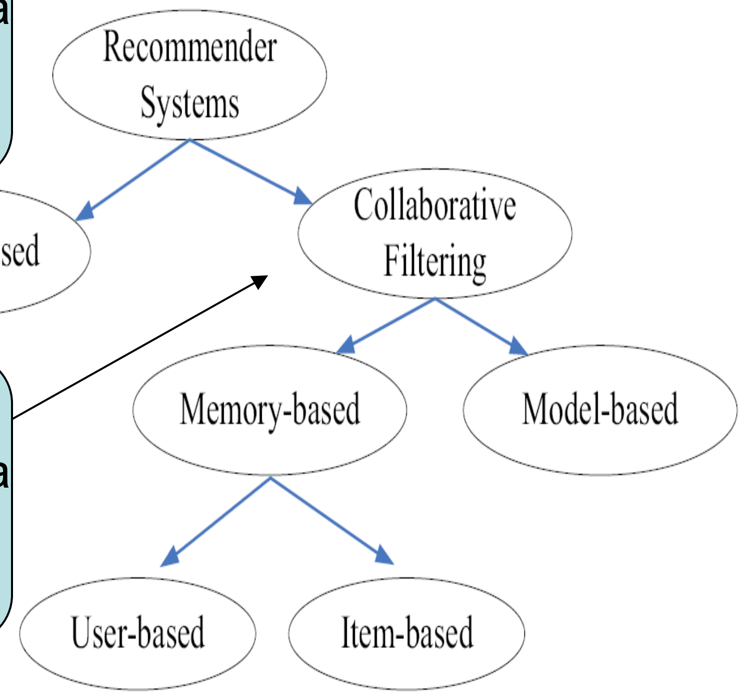
- Content-based

Main Idea: Content Features
Naive Method: If a user has given a high rating to a movie directed by Ang Lee, other movies directed by Ang Lee will be recommended to this user.

    - [Deshpande 2004], [Sarwar 2001]
  - Hybrid
    - [Wang 2006], [Ma 2007]
- Model-based

Main Idea: Common Behavior Patterns
Naive Method: If a user has given high a rating to a movie A; and many users who like A also like B. Movie B will be recommended to the user

  - Model-based
    - [Salakhutdinov 2008]
    - [Koren 2008]
    - [Koren 2010]
    - [Weimer 2007]



Applications of Recommender Systems

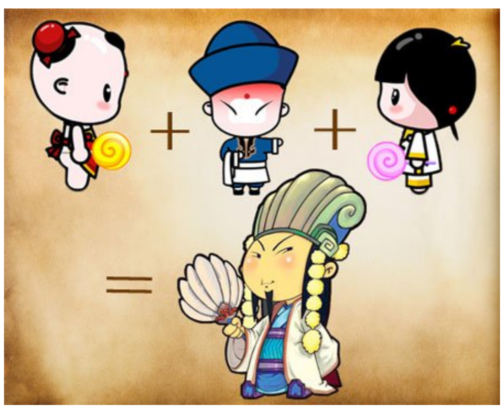
System	**Content**
Amazon	books, CDs, others
Epinions	books, CDs, others
MovieLens	movie
Netflix	dvd
Yahoo! Music	music
Grundy	books
Video Recommender	video
Ringo	music
PHOAKS	textual information
Jester	jokes
Fab System	Web page

Outline

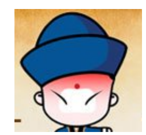
- Background of Recommender Systems
- Motivation of the Thesis
- Part 1: Relational Fusion of Multiple Features
- Part 2: Effective Fusion of Regression and Ranking
- Part 3: Effective Fusion of Quality and Relevance
- Part 4: Impression Efficiency Optimization
- Conclusion

Fusion-based Approaches

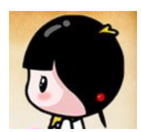
- To combine multiple information and algorithms to get the better performance
- Two heads are better than one



Ratings



Social Information

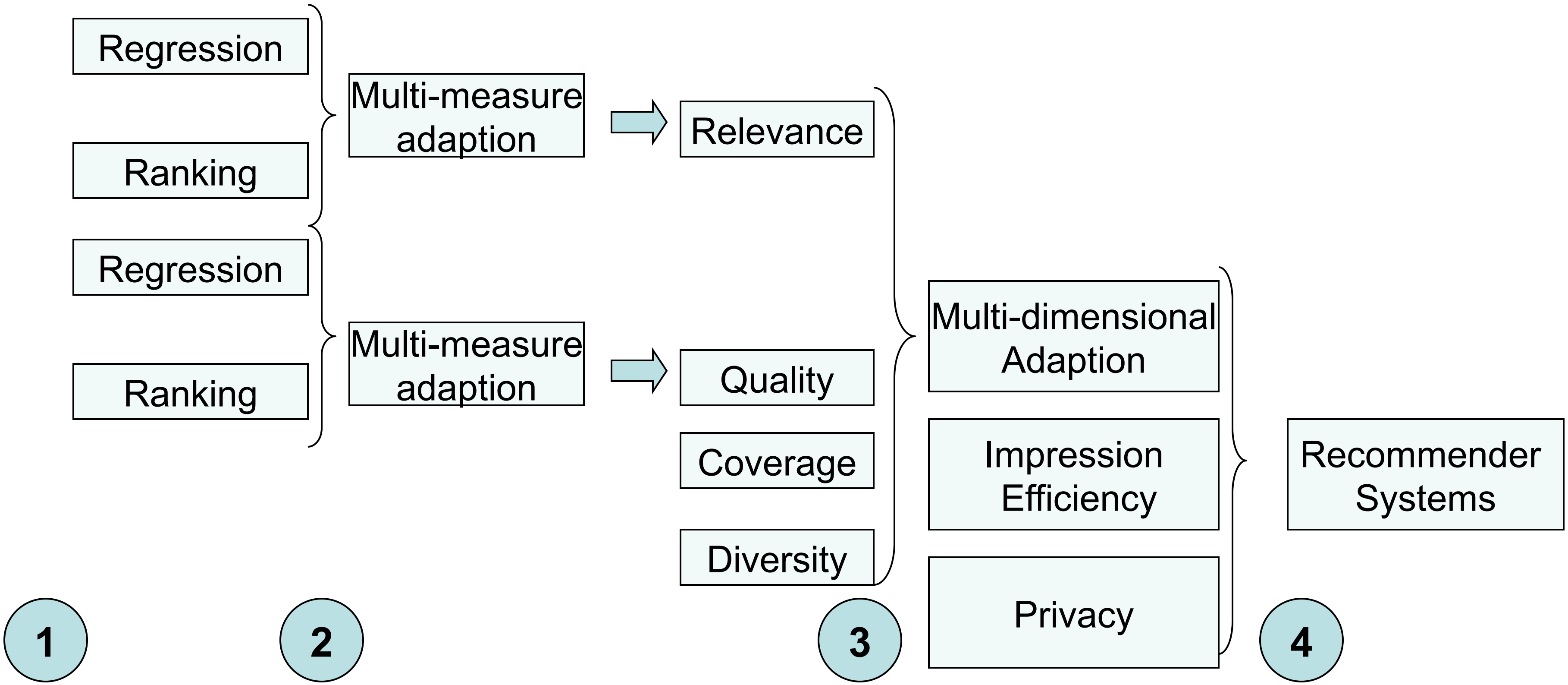


Location

Better Recommendation

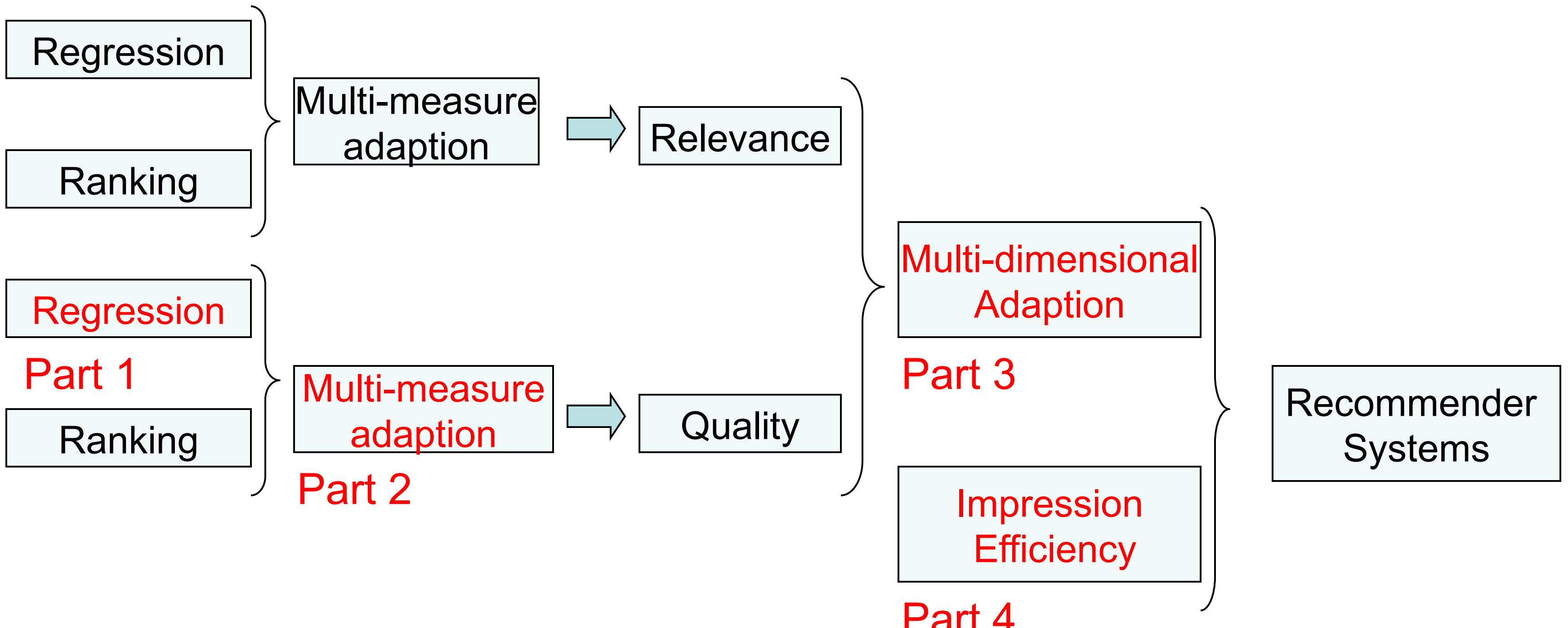
- Fusion is effective
  - Reports on competitions such as Netflix [Koren 2008], KDD CUP [Kurucz 2007][Rosset 2007]

Roadmap of the Thesis (1): Evaluation Structure of Recommender Systems



1. Single measure and single dimension
2. Multi-measure adaption
3. Multi-dimensional adaption
4. Success of recommender systems

Roadmap of the Thesis (2): Summary

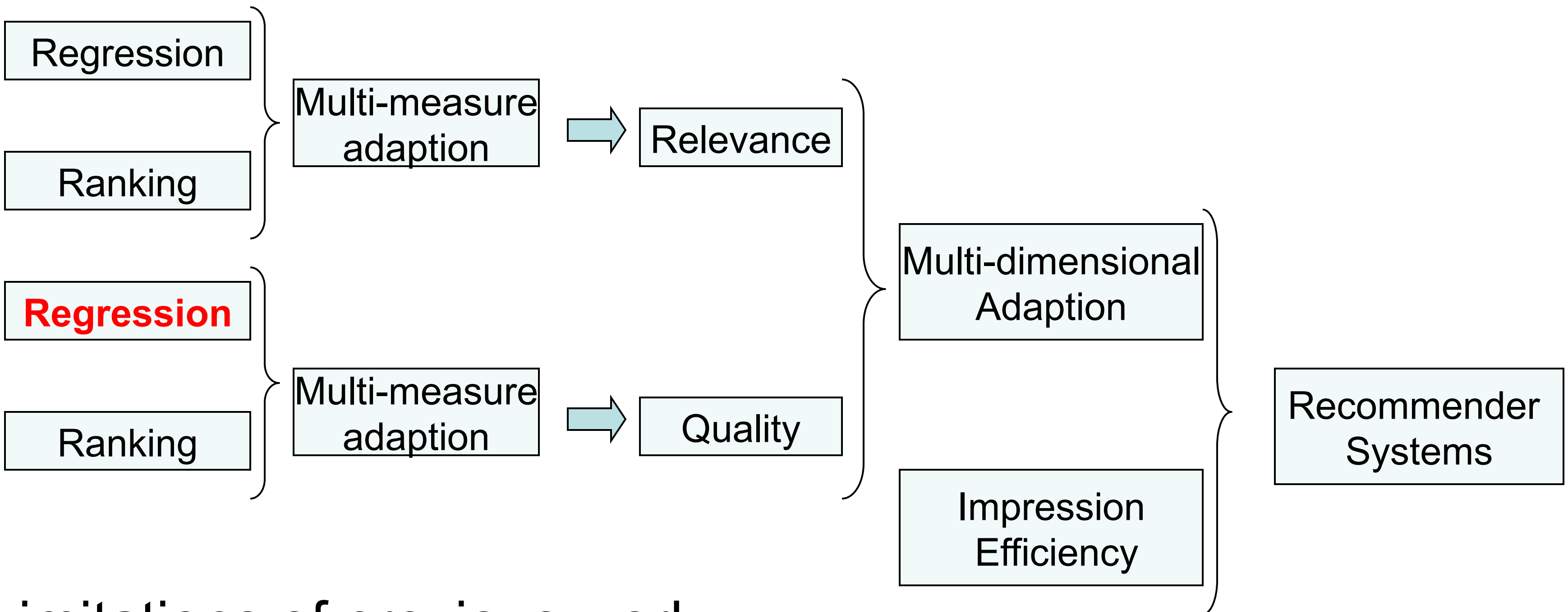


- Part 1: Relational fusion of multiple features
  - Full and oral paper in CIKM 2009 (cited count: 14)
- Part 2: Effective fusion of regression and ranking
  - Submitted to CIKM 2011
- Part 3: Effective fusion of quality and relevance
  - Full paper in WSDM 2011
- Part 4: Impression efficiency optimization
  - Prepared for WWW 2012

Outline

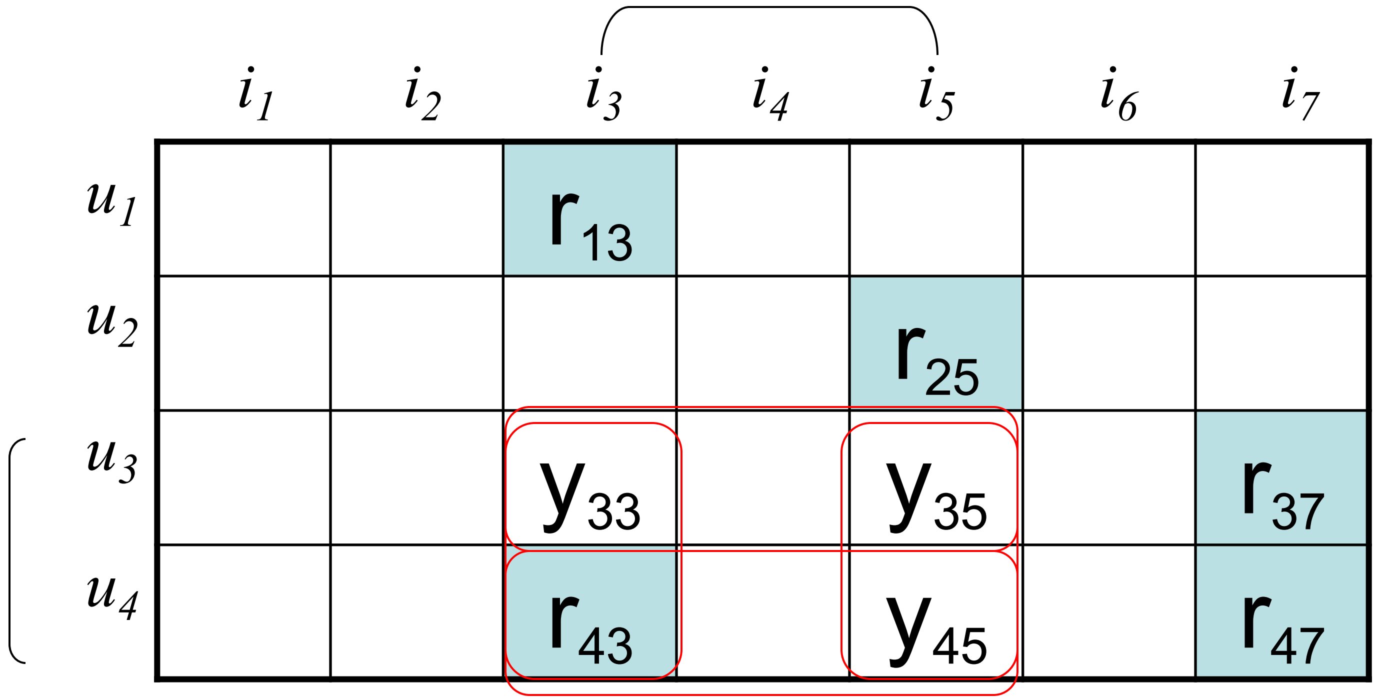
- Background of Recommender Systems
- Motivation of the Thesis
- Part 1: Relational Fusion of Multiple Features
- Part 2: Effective Fusion of Regression and Ranking
- Part 3: Effective Fusion of Quality and Relevance
- Part 4: Impression Efficiency Optimization
- Conclusion

Part 1: Relational Fusion of Multiple Features



- Limitations of previous work
  - Lack of relational dependency
  - Difficulty for integrating features

Limitation (1): Lack of Relational Dependency within Predictions

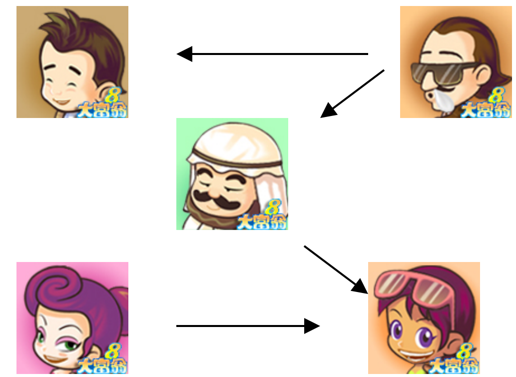


- Heuristic Fusion [Wang 2006]
  - Difficult to measure similarity between y_{35} and y_{43}
  - Cannot guarantee the nearness between y_{35} and y_{33} (or y_{35} and y_{45})
- EMDP [Ma 2007]
  - Error propagation

Limitation (2): Difficult to Integrate Features into an Unified Approach

Content information of items

Profile information of users



Trust relationship

- Linear Integration [Ma 2007]
  - Difficult to calculate the feature function weights

Our Solution: Multi-scale Continuous Conditional Random Fields (MCCRF)

- Propose to utilize MCCRF as a relational fusion-based approach, which is extended from single-scale continuous conditional random fields
- Relational dependency within predictions can be modeled by the Markov property
- Feature weights are globally optimized

Relational Recommendation Formulation

Let X denote observations.

Let Y denote predictions.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	y_{11}	y_{12}	r_{13}	y_{14}	y_{15}	y_{16}	y_{17}
u_2	y_{21}	y_{22}	y_{23}	y_{24}	r_{25}	y_{26}	y_{27}
u_3	y_{31}	y_{32}	y_{33}	y_{34}	y_{35}	y_{36}	r_{37}
u_4	y_{41}	y_{42}	r_{43}	y_{44}	y_{45}	y_{46}	r_{47}

Traditional Recommendation

$$y_{(l,m)} = f(X)$$

Relational Recommendation

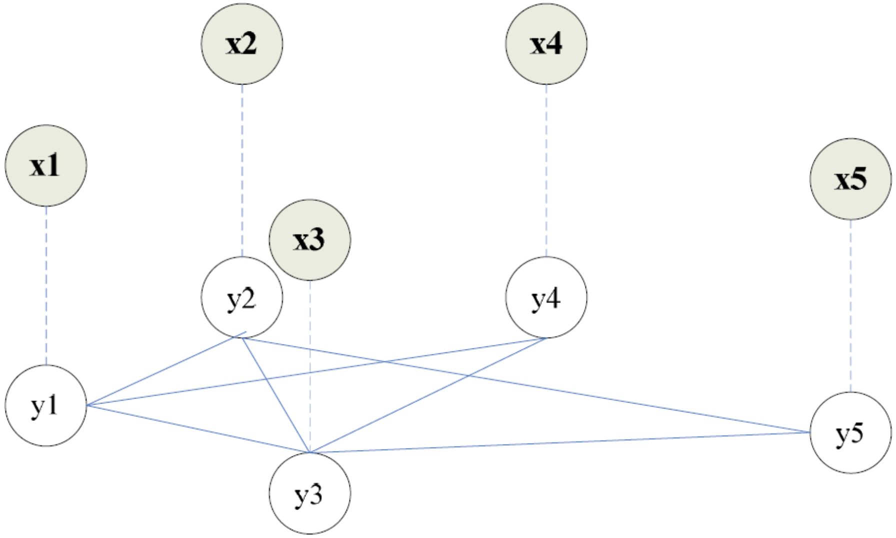
$$Y = f(X)$$

$$y_{(l,m)} = f(X, y_{(-l,-m)})$$

Traditional Single-scale Continuous Conditional Random Fields

$$p(Y|X) = \frac{1}{Z_{sgl}(X)} \exp\left\{\sum_m \alpha \cdot H(y_m, X) + \sum_{m,n} \beta \cdot G(y_m, y_n, X)\right\}$$

$$Z_{sgl}(X) = \int_y \exp\left\{\sum_m \alpha \cdot H(y_m, X) + \sum_{m,n} \beta \cdot G(y_m, y_n, X)\right\} dy$$



$$h_{t1}(y_m, X) = -(y_m - x_{m,t1})^2,$$

$$g_{t2}(y_m, y_n, X) = -\frac{1}{2} M_{m,n,t2} (y_m - y_n)^2$$

Feature example:

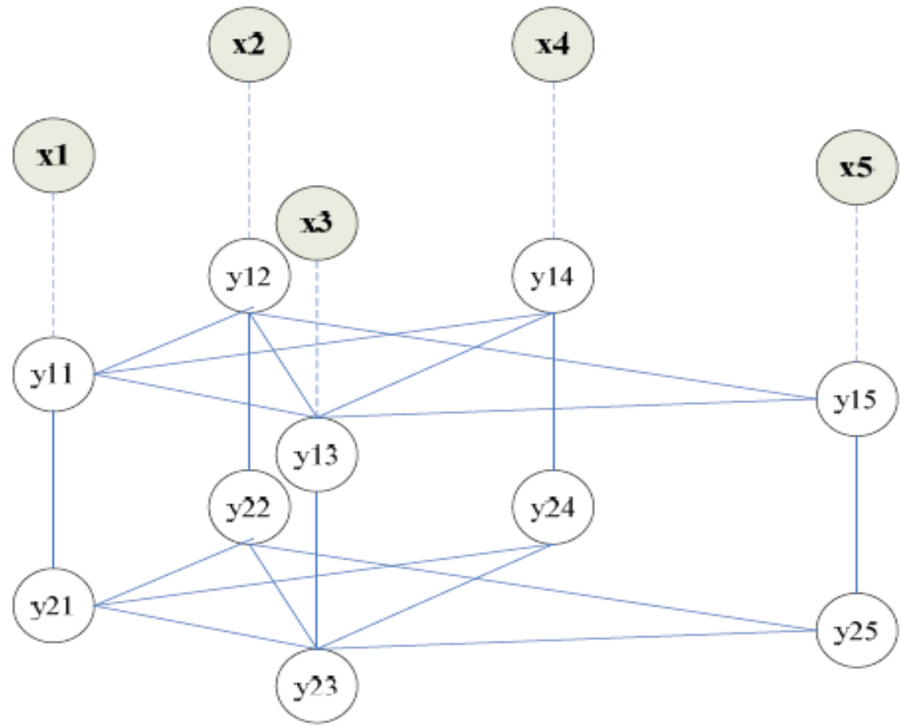
$x_{m,t1}$ ⇒ Avg. Rate for item *m*

$M_{m,n,t2}$ ⇒ Similarity between item *m* and item *n*

Multi-scale Continuous Conditional Random Fields

$$
\begin{aligned}
p(Y|X) = \frac{1}{Z_{mul}(X)} \exp\Big\{ & \sum_l \sum_m \alpha \cdot H(y_{l,m}, X) \\
& + \sum_l \sum_{m,n} \beta \cdot G(y_{l,m}, y_{l,n}, X) \\
& + \sum_m \sum_{l,j} \gamma \cdot R(y_{l,m}, y_{j,m}, X) \Big\}
\end{aligned}
$$

$$
\begin{aligned}
Z_{mul}(X) = \int_y \exp\Big\{ & \sum_l \sum_m \alpha \cdot H(y_{l,m}, X) \\
& + \sum_l \sum_{m,n} \beta \cdot G(y_{l,m}, y_{l,n}, X) \\
& + \sum_m \sum_{l,j} \gamma \cdot R(y_{l,m}, y_{j,m}, X) \Big\} dy
\end{aligned}
$$



$$
\begin{aligned}
h_{t1}(y_{l,m}, X) &= -(y_{l,m} - x_{l,m,t1})^2 \\
g_{t2}(y_{l,m}, y_{l,n}, X) &= -\frac{1}{2} M_{m,n,t2}(y_{l,m} - y_{l,n})^2 \\
r_{t_3}(y_{l,m}, y_{j,m}, X) &= -\frac{1}{2} U_{l,j,t3}(y_{l,m} - y_{j,m})^2
\end{aligned}
$$

Feature example:

- $x_{l,m,t1}$ ⇒ Avg. Rate for user *l*
- $M_{m,n,t2}$ ⇒ Similarity between item *m* and item *n*
- $U_{l,j,t3}$ ⇒ trust between user *l* and user *j*

Features

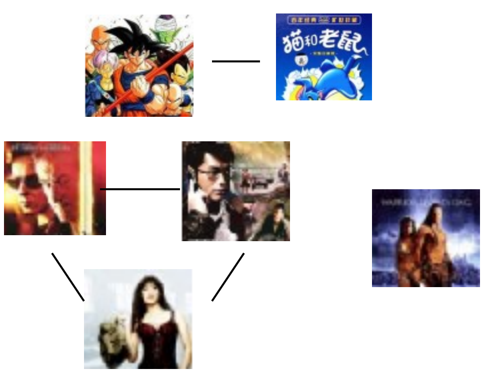
Local features

Avg. rating of the same occupation

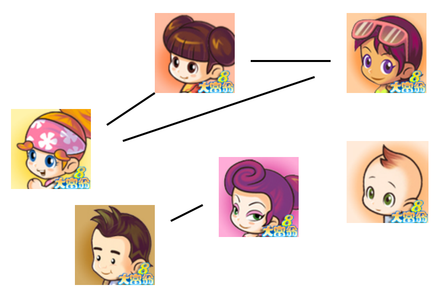
Avg. rating of the same age and gender

Avg. rating of the same genre

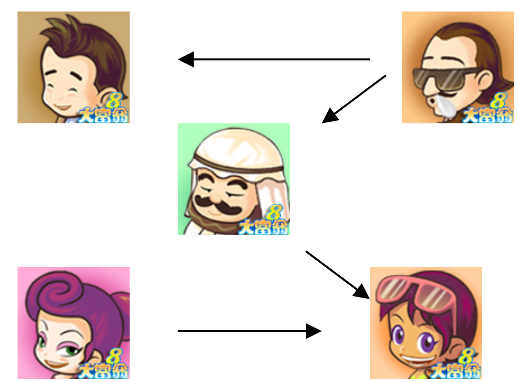
Relational features



Similarity among items



Similarity among users



Trust relation

Algorithms-Training and Inference

$$p(Y|X) = \frac{1}{Z_{mul}(X)} \exp\left\{\sum_{l}\sum_{m} \alpha \cdot H(y_{l,m}, X) + \sum_{l}\sum_{m,n} \beta \cdot G(y_{l,m}, y_{l,n}, X) + \sum_{m}\sum_{l,j} \gamma \cdot R(y_{l,m}, y_{j,m}, X)\right\}$$

Training process

Objective Function：

$$L_\lambda = \sum_{k=0}^{N} \log p_\lambda(y_k|x_k) = \sum_{k}^{N} [\lambda \cdot F(y_k, x_k) - \log Z_\lambda(x_k)]$$

$$L_\lambda = L'_{\lambda'} = \sum_{k}^{N} [e^{\lambda'} \cdot F(y_k, x_k) - \log Z_{e^{\lambda'}}(x_k)]$$

Gradient：

$$\nabla L'_{\lambda'} = e^{\lambda'} \cdot \sum_{k=0}^{N} \left[F(y_k, x_k) - E_{p_{\lambda'}(Y|x_k)}\left(F(Y, x_k)\right)\right]$$

Gibbs Sampling：

$$E_{p_\lambda(Y|x_k)}(F(Y|x_k)) = \frac{1}{S}\left(\sum_{1}^{S} F(\tilde{y}, x_k)\right)$$

$$P(y_{l,m}|y_{-l,-m}, X) = \frac{P(y_{l,m}, y_{-l,-m}|X)}{\int_{y_{l,m}} P(y_{l,m}, y_{-l,-m}|X) dy_{l,m}}$$

Inference process

Objective Function：

$$\hat{y} = \arg\max_{y} p(y|x)$$

Simulated Annealing：

$$p_i(\tilde{y}|x) = p^{1/T(i)}(\tilde{y}|x)$$

Experiment-Setup

- Datasets
  - MovieLens
  - Epinions

Statistics	MovieLens	Epinions
Min. Num. of Ratings/User	20	1
Min. Num. of Ratings/Item	1	1
Max. Num. of Ratings/User	737	1022
Max. Num. of Ratings/Item	583	2018
Avg. Num. of Ratings/User	106.04	16.55
Avg. Num. of Ratings/Item	59.45	4.76

- Metrics
  - Mean Absolute Error (MAE)
  - Root Mean Square Error (RMSE)

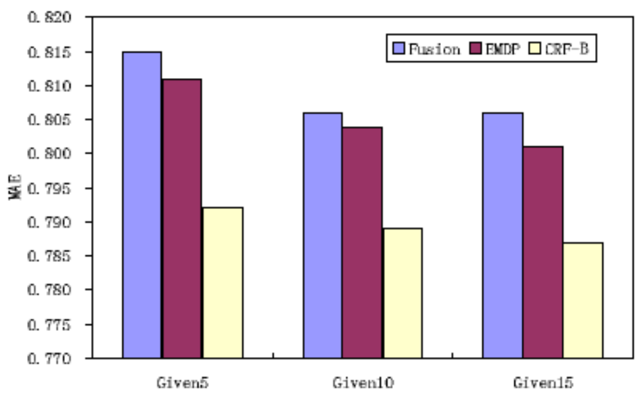
$$MAE = \frac{\sum |R_{u,i} - \tilde{R}_{u,i}|}{N}$$

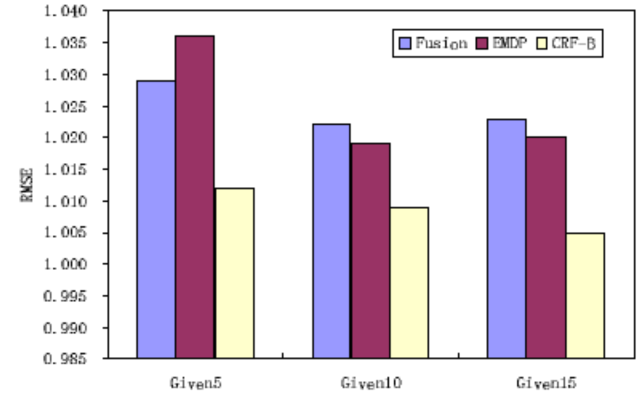
$$RMSE = \sqrt{\frac{\sum (R_{u,i} - \tilde{R}_{u,i})^2}{N}}$$

- Baselines
  - EPCC: combination of UPCC and IPCC (memory)
  - Aspect Model (AM): classical latent method (model)
  - Fusion: directly find similar users’ similar items
  - EMDP: two rounds prediction

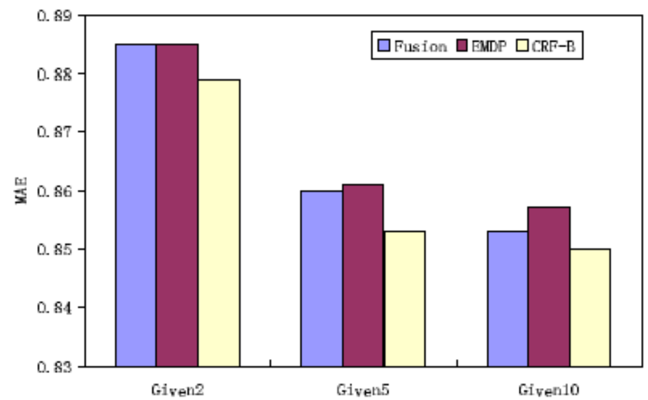
Effectiveness of Dependency

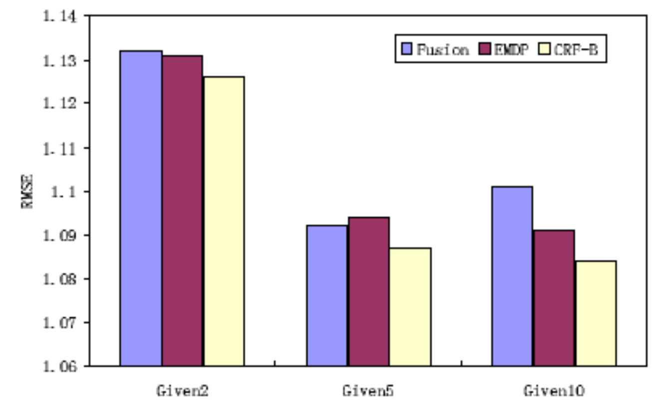
MovieLens





Epinions

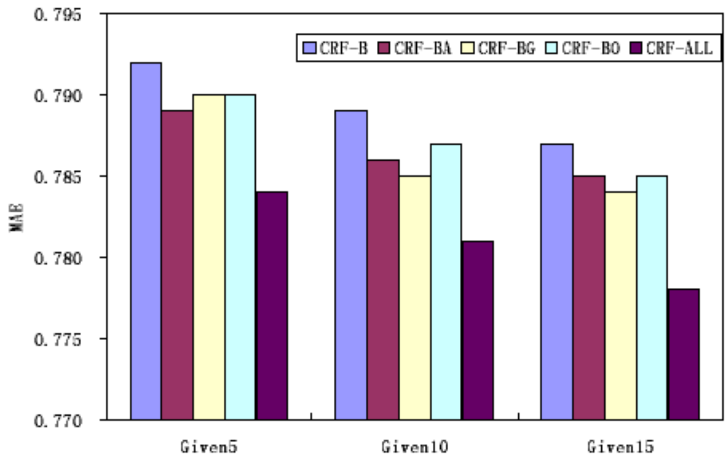


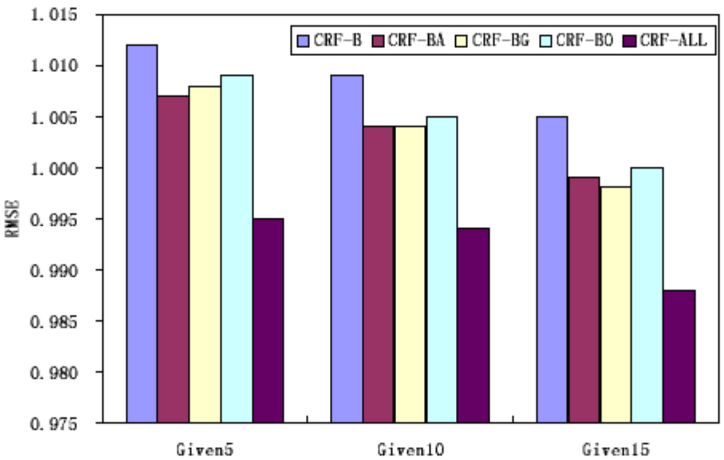


- Our MCCRF approach outperforms others consistently
  - Fusion [Wang 2006] calculate inaccurate similarity between two predictions
  - EMDP [Ma 2007] has the error propagation problem

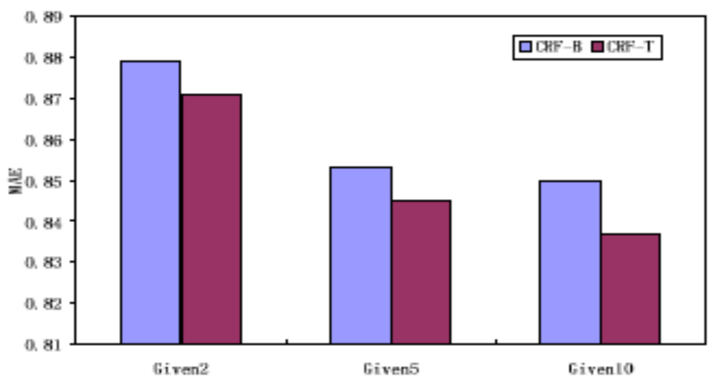
Effectiveness of Features

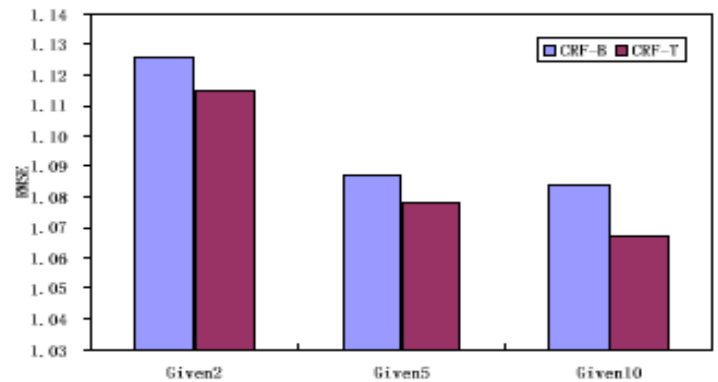
MovieLens





Epinions





- Approaches with more features perform better
  - Two heads are better than one
  - MCCRF is effective in fusion of multiple features

Overall Performance

Table. Performance in MovieLens dataset

Methods	MAE			RMSE		
	Given5	Given10	Given15	Given5	Given10	Given15
EPCC	0.835	0.830	0.815	1.065	1.059	1.033
AM	0.827	0.819	0.816	1.041	1.031	1.025
Fusion	0.815	0.806	0.805	1.029	1.024	1.022
EMDP	0.811	0.804	0.801	1.036	1.019	1.020
MCCRF	**0.784**	**0.781**	**0.778**	**0.995**	**0.994**	**0.988**

Table. Performance in Epinions dataset

Methods	MAE			RMSE		
	Given2	Given5	Given10	Given2	Given5	Given10
EPCC	0.887	0.867	0.858	1.136	1.105	1.092
AM	0.893	0.885	0.863	1.132	1.131	1.101
Fusion	0.885	0.860	0.853	1.132	1.092	1.101
EMDP	0.885	0.861	0.857	1.131	1.094	1.091
MCCRF	**0.871**	**0.845**	**0.837**	**1.115**	**1.078**	**1.067**

- The proposed MCCRF performs the best
  - Effectiveness of relational feature dependency
  - Effiective fusion of multiple features

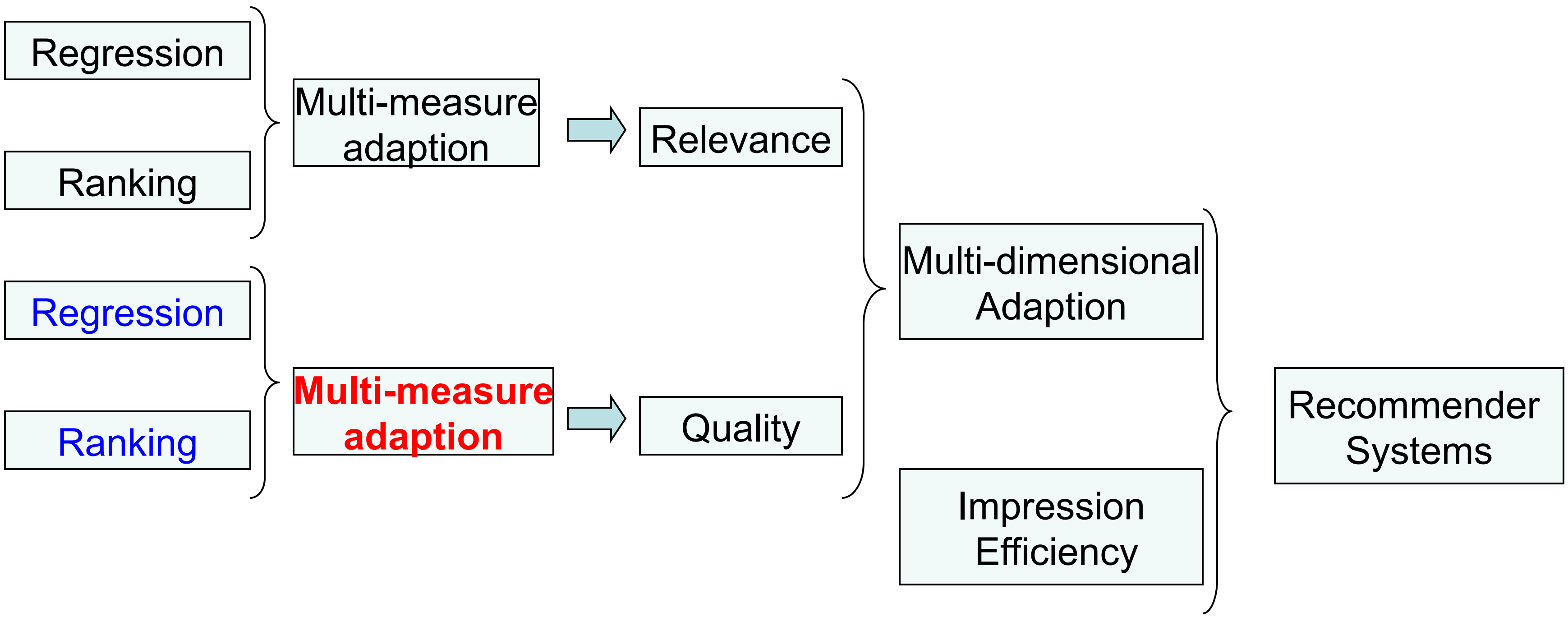
Summary of Part 1

- We propose a novel model MCCRF as a framework for relational recommendation
- We propose an MCMC-based method for training and inference
- Experimental verification on the effectiveness of the proposed approach on MovieLens and Epinions

Outline

- Background of Recommender Systems
- Motivation of the Thesis
- Part 1: Relational Fusion of Multiple Features
- Part 2: Effective Fusion of Regression and Ranking
- Part 3: Effective Fusion of Quality and Relevance
- Part 4: Impression Efficiency Optimization
- Conclusion

Part 2: Effective Fusion of Regression and Ranking



- Limitation of previous work
  - Over bias in single measure
  - They cannot adapt to the other measure. Information is not fully utilized for data sparse problem

Regression v.s. Ranking

(a)

	i_1	i_2	i_3	i_4	i_5
u_j	4.0			3.0	
Ground Truth:		3.0	4.0		1.0
Algorithm1:		3.6	3.5		1.0
Algorithm2:		4.2	4.5		1.0

(b)

	i_1	i_2	i_3	i_4	i_5
u_j	4.0			3.0	
u_1	3.0		2.0	4.0	
u_2	5.0		4.0	1.0	

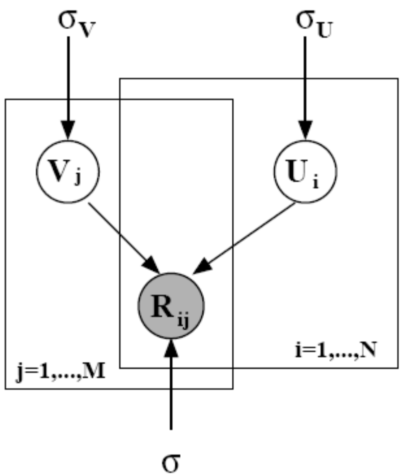
- Regression: modeling and predicting the ratings
  - Output, y_{j2}=3.6, y_{j3}=3.5, y_{j5}=1.0 …
- Ranking: modeling and predicting the ranking orders
  - Output, $y_{j3}>y_{j2}>y_{j5}$…
- Comparisons
  - Advantage of regression
    - (1) intuitive (2) simple complexity
  - Advantage of ranking
    - (1) richer information (2) direct for applications

Our Solution: Combining Regression and Ranking in Collaborative Filtering

- The work is the first attempt to investigate the combination of regression and ranking in collaborative filtering community
- As the first ever solution, we propose combination methods in both model-based and memory-based algorithms

Model-based Methods Selection

- Probabilistic graph of the models



$$p(V|\sigma_V^2) = \prod_{j=1}^{M} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}) \quad p(U|\sigma_U^2) = \prod_{i=1}^{N} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I})$$

- Regression method
  - Probabilistic Matrix Factorization (PMF) [Salakhutdinov 2007]

$$\arg\min_{U,V} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij}(R_{ij} - g(U_i^T V_j))^2 + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2)$$

- Ranking method
  - List-wise Matrix Factorization (LMF) [Shi 2010]

$$\arg\min_{U,V} \sum_{i=1}^{N} \sum_{j=1}^{M} P_{l_i}(R_{ij}) \log(\frac{P_{l_i}(R_{ij})}{(g(U_i^T V_j))}) + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2)$$

$$P_{l_i}(R_{ij}) = \frac{\exp(R_{ij})}{\sum_{k=1}^{K} \exp(R_{ik})}$$

Model-based Combination

- Objective function

$$\min_{U,V} \alpha_1 Loss_{Reg}(U,V) + \alpha_2 Loss_{Rank}(U,V) + Regularization(U,V)$$

$$Loss_{reg}(U,V) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - g(U_i^T V_j))^2$$

$$Loss_{rank}(U,V) = \sum_{i=1}^{N}\sum_{j=1}^{M} P_{l_i}(R_{ij}) \log(\frac{P_{l_i}(R_{ij})}{(g(U_i^T V_j))})$$

- Gradient descent optimization

$$\begin{aligned}
\frac{\partial \zeta}{\partial V_j} = & \quad \alpha_1 \sum_{i=1}^{N} -I_{ij}(R_{ij} - g(U_i^T V_j))g'(U_i^T V_j)U_i \\
+ & \quad \alpha_2 \sum_{i=1}^{N} I_{ij} \frac{\exp(g(U_i^T V_j))}{\sum_{k=1}^{M} I_{ik}\exp(g(U_i^T V_j))} g'(U_i^T V_j)U_i \\
- & \quad \alpha_2 \sum_{i=1}^{N} I_{ij} \frac{\exp(R_{ij})}{\sum_{k=1}^{M} I_{ik}\exp(R_{ik})} g'(U_i^T V_j)U_i \\
+ & \quad \lambda V_j
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \zeta}{\partial U_i} = & \quad \alpha_1 \sum_{j=1}^{M} -I_{ij}(R_{ij} - g(U_i^T V_j))g'(U_i^T V_j)V_j \\
+ & \quad \alpha_2 \sum_{j=1}^{M} I_{ij} \frac{\exp(g(U_i^T V_j))}{\sum_{k=1}^{M} I_{ik}\exp(g(U_i^T V_j))} g'(U_i^T V_j)V_j \\
- & \quad \alpha_2 \sum_{j=1}^{M} I_{ij} \frac{\exp(R_{ij})}{\sum_{k=1}^{M} I_{ik}\exp(R_{ik})} g'(U_i^T V_j)V_j \\
+ & \quad \lambda U_i
\end{aligned}$$

Memory-based Methods Selection

- Regression Method
  - User-based PCC [Breese 1998]

Pearson Correlation Coefficient (PCC) similarity

$$Sim(a,u) = \frac{\sum_{i \in I(a)\cap I(u)} (r_{a,i} - \overline{r}_a)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I(a)\cap I(u)} (r_{a,i} - \overline{r}_a)^2} \sqrt{\sum_{i \in I(a)\cap I(u)} (r_{u,i} - \overline{r}_u)^2}}$$

Rating Prediction

$$f(u,i) = \overline{u} + \frac{\sum_{u_a \in S(u)} Sim(u_a, u)(r_{u_a,i} - \overline{u}_a)}{\sum_{u_a \in S(u)} Sim(u_a, u)}$$

- Ranking Method
  - EigenRank [Liu SIGIR 2008]

Kendall Rank Correlation Coefficient (KRCC) similarity

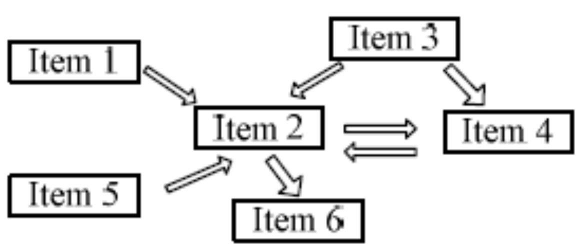
$$s_{u,v} = 1 - \frac{4 \times \sum_{i,j \in I_u \cap I_v} I^{-}\big((r_{u,i} - r_{u,j})(r_{v,i} - r_{v,j})\big)}{|I_u \cap I_v| \cdot (|I_u \cap I_v| - 1)}$$

Ranking Prediction

$$\Psi(i,j) = \frac{\sum_{v \in N_u^{i,j}} s_{u,v} \cdot (r_{v,i} - r_{v,j})}{\sum_{v \in N_u^{i,j}} s_{u,v}}$$

$$p(j|i) = \frac{e^{\Psi(j,i)}}{\sum_{j \in I} e^{\Psi(j,i)}}$$

$$y_i = \pi_i, \pi = \pi * P,$$



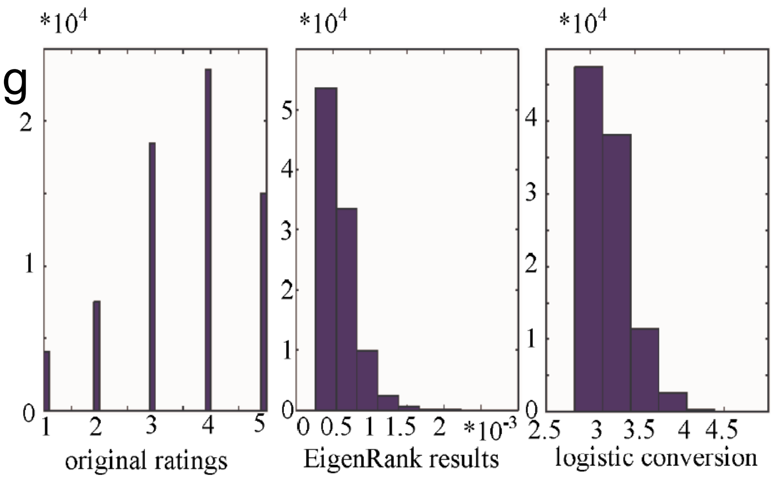
Memory-based Combination

- No objective function
  - To combine the results from regression and ranking algorithms
- Challenge
  - The output values are incompatible
- Naive combination

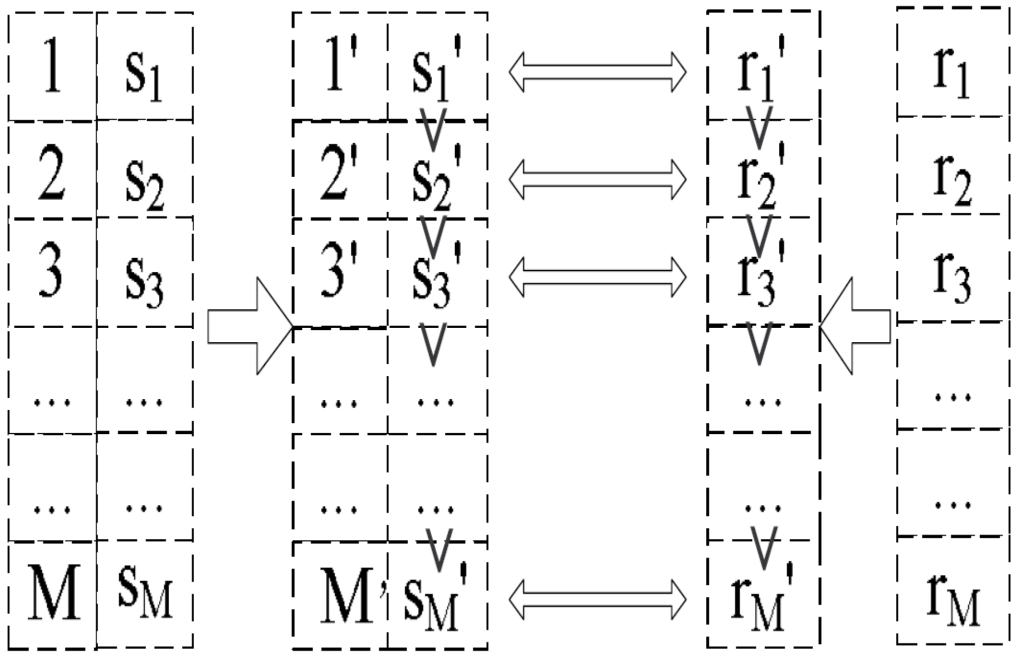
$$F(x) = 1/(1+\exp(-x))$$

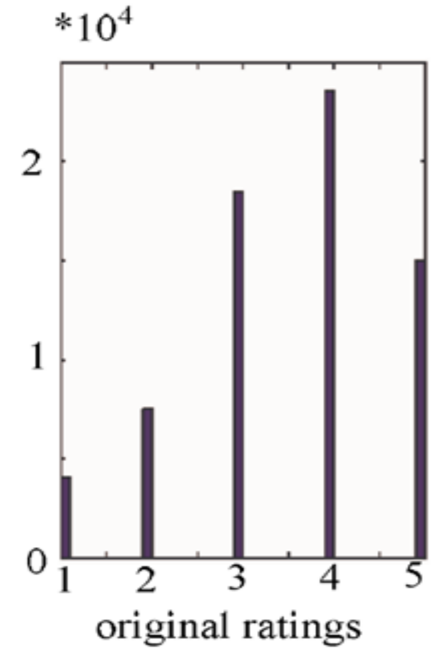
$$Rate_{combination} = \alpha * Rate_{rating} + (1-\alpha) * Rate_{regression}$$

Conflict: The results of the ranking model do not follow the Gauss distribution as the real data.



Sampling Trick





$P(R) \sim N(R|\mu, \sigma^2)$

$\mu = 3.88 \qquad \sigma^2 = 1.26$

EigenRank results

Sampling results

$$Rate_{combination} = \alpha * Rate_{rating} + (1 - \alpha) * Rate_{regression}$$

Experimental Setup

- Datasets
  - MovieLens
    - 1,682 items, 943 users
    - 100,000 ratings
  - Netflix
    - 17,770 items, 480,000users
    - 100,000,000 ratings

Statistics	MovieLens	Netflix
Avg. Num. of Ratings/User	106.04	209.25
Avg. Num. of Ratings/Item	59.45	5654.50
Min. Num. of Ratings/User	20	1
Min. Num. of Ratings/Item	1	3
Max. Num. of Ratings/User	737	17653
Max. Num. of Ratings/Item	583	232944
Density of User/Item Matrix	6.3%	1.18%

- Metrics
  - Regression Measure
    - MAE, RMSE
  - Ranking Measure
    - Normalized Discount Cumulated Gain (NDCG)
- Setup
  - 2/3 users training, 1/3 users testing
  - Given 5, Given 10, Given 15
- Regression-prior and Ranking-prior
  - RegPModel, RegPMemory
  - RankPModel, RankPMemory

Performance of *Model-based* Combination

MovieLens Dataset

Methods	Given5				
	MAE	RMSE	NDCG1	NDCG3	NDCG5
PMF	0.870	1.153	0.640	0.663	0.673
LMF	0.967	1.350	**0.692**	0.687	0.701
RegPModel	**0.845**	**1.081**	0.672	0.676	0.692
RankPModel	0.863	1.132	0.681	**0.689**	**0.702**
Methods	Given10				
	MAE	RMSE	NDCG1	NDCG3	NDCG5
PMF	0.825	1.067	0.691	0.700	0.719
LMF	0.953	1.301	0.679	0.708	0.732
RegPModel	**0.809**	**1.003**	0.703	0.708	0.727
RankPModel	0.813	1.028	**0.705**	**0.719**	**0.736**
Methods	Given15				
	MAE	RMSE	NDCG1	NDCG3	NDCG5
PMF	0.780	0.97	0.693	0.730	0.753
LMF	0.946	1.290	0.734	0.748	0.763
RegPModel	**0.781**	**0.960**	0.742	0.750	**0.768**
RankPModel	0.786	0.964	**0.745**	**0.752**	0.765

Netflix Dataset

Methods	Given5				
	MAE	RMSE	NDCG1	NDCG3	NDCG5
PMF	0.820	1.069	0.694	0.691	0.691
LMF	0.979	1.404	0.713	0.702	0.715
RegPModel	**0.819**	**1.062**	0.697	0.685	0.696
RankPModel	0.828	1.063	**0.721**	**0.713**	**0.723**
Methods	Given10				
	MAE	RMSE	NDCG1	NDCG3	NDCG5
PMF	0.819	1.061	0.709	0.720	0.733
LMF	0.944	1.444	0.712	**0.721**	0.731
RegPModel	**0.781**	**0.979**	0.724	0.716	0.727
RankPModel	0.801	1.013	**0.732**	0.719	**0.735**
Methods	Given15				
	MAE	RMSE	NDCG1	NDCG3	NDCG5
PMF	0.769	0.947	0.749	0.742	0.763
LMF	0.918	1.292	0.722	0.743	0.764
RegPModel	**0.757**	**0.922**	0.747	0.750	0.722
RankPModel	0.762	0.931	**0.750**	**0.755**	**0.775**

- The combination methods outperform single-measure-adapted methods in all the metrics
- Regression-prior model has also an improvement in ranking-based measure
- Ranking-prior model has also an improvement in regression-based measure

Performance of *Memory-based* Combination

MovieLens Dataset

Methods	MAE	RMSE	NDCG1	NDCG3	NDCG5
	Given5				
PCC	0.877	1.257	0.668	0.671	0.690
EigenRank	0.878	1.287	0.684	0.709	0.719
RegPMemory	**0.817**	**1.099**	**0.696**	0.698	0.711
RankPMemory	0.848	1.194	0.693	**0.711**	**0.721**
	Given10				
PCC	0.806	1.067	0.690	0.713	0.734
EigenRank	0.876	1.288	0.692	0.718	0.737
RegPMemory	**0.789**	**1.028**	0.699	0.720	0.742
RankPMemory	0.836	1.163	**0.700**	**0.725**	**0.745**
	Given15				
PCC	0.780	0.999	0.710	0.732	0.752
EigenRank	0.876	1.285	0.722	0.741	0.758
RegPMemory	**0.774**	**0.987**	0.720	0.743	0.763
RankPMemory	0.803	1.066	**0.726**	**0.748**	**0.767**

Netflix Dataset

Methods	MAE	RMSE	NDCG1	NDCG3	NDCG5
	Given5				
PCC	0.760	0.936	0.753	0.767	0.781
EigenRank	1.033	1.687	**0.792**	0.775	0.793
RegPMemory	**0.753**	**0.918**	0.760	0.770	0.783
RankPMemory	1.005	1.600	0.790	**0.776**	**0.794**
	Given10				
PCC	0.776	0.982	0.727	0.738	0.755
EigenRank	1.028	1.645	0.745	0.750	0.758
RegPMemory	**0.767**	**0.954**	0.730	0.746	0.759
RankPMemory	0.882	1.235	**0.755**	**0.755**	**0.767**
	Given15				
PCC	0.838	1.161	0.707	0.710	0.712
EigenRank	1.046	1.685	0.744	0.732	0.740
RegPMemory	**0.819**	**1.079**	0.726	0.729	0.739
RankPMemory	1.020	1.605	**0.747**	**0.735**	**0.744**

- The combination methods outperform single-measure-adapted methods in all the metrics
- Regression-prior model has also an improvement in ranking-based measure
- Ranking-prior model has also an improvement in regression-based measure

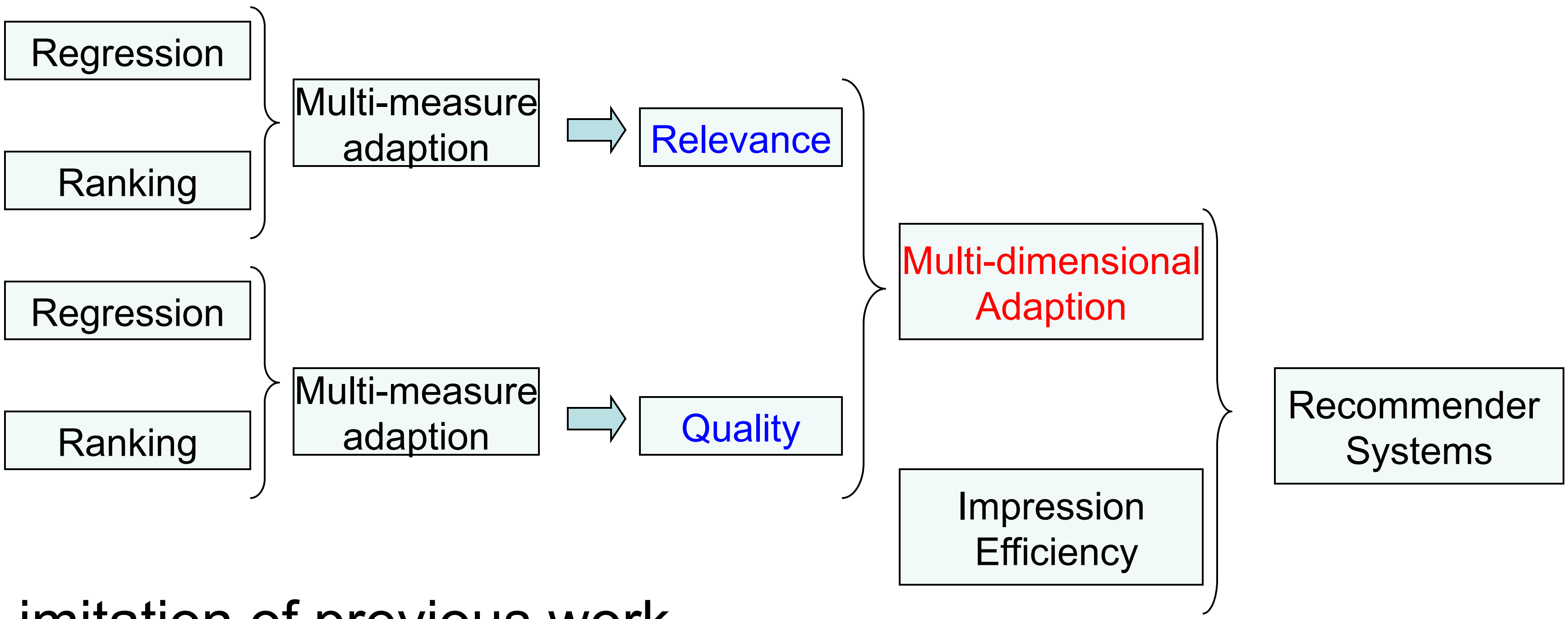
Summary of Part 2

- We conduct the first attempt to investigate the fusion of regression and ranking to solve the limitation of single-measure collaborative filtering algorithms
- We propose combination methods from both model-based and memory-based aspects
- Experimental result demonstrated that the combination will enhance performances in both metrics

Outline

- Background of Recommender Systems
- Motivation of the Thesis
- Part 1: Relational Fusion of Multiple Features
- Part 2: Effective Fusion of Regression and Ranking
- Part 3: Effective Fusion of Quality and Relevance
- Part 4: Impression Efficiency Optimization
- Conclusion

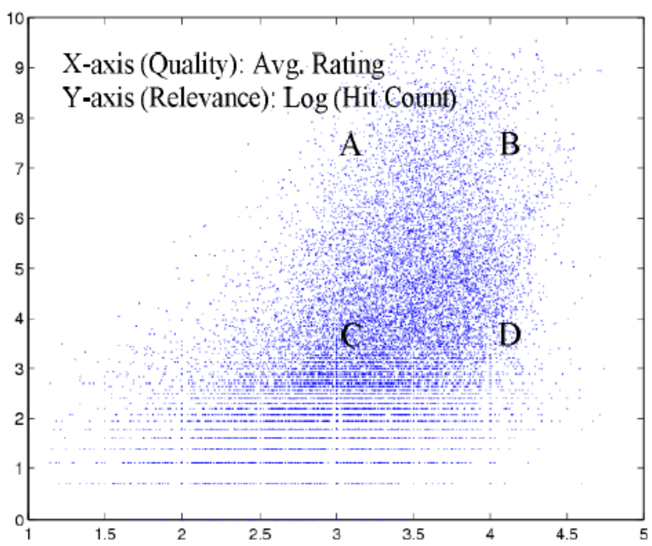
Part 3: Effective Fusion of Quality and Relevance

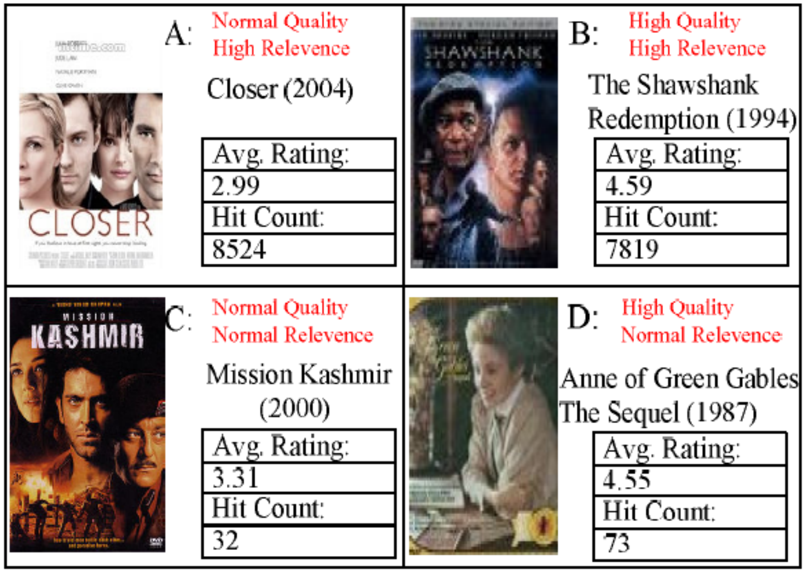


- Limitation of previous work
  - Single-dimensional algorithm cannot adapt to multi-dimensional performance
  - Incomplete recommendation

Quality v.s. Relevance

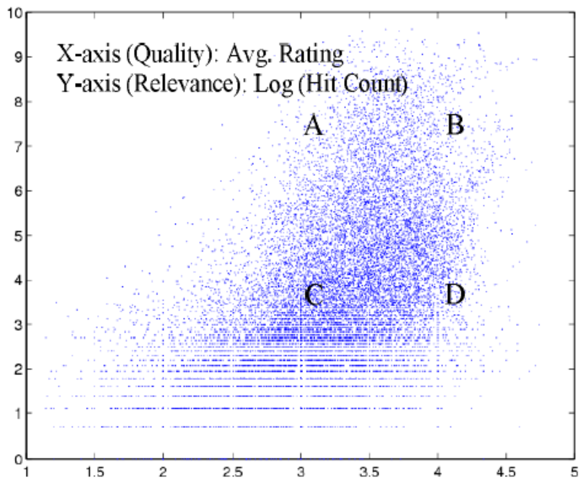
- Quality: Whether recommended items can be rated with high scores
- Relevance: How many recommended items will be visited by the user
- Examples

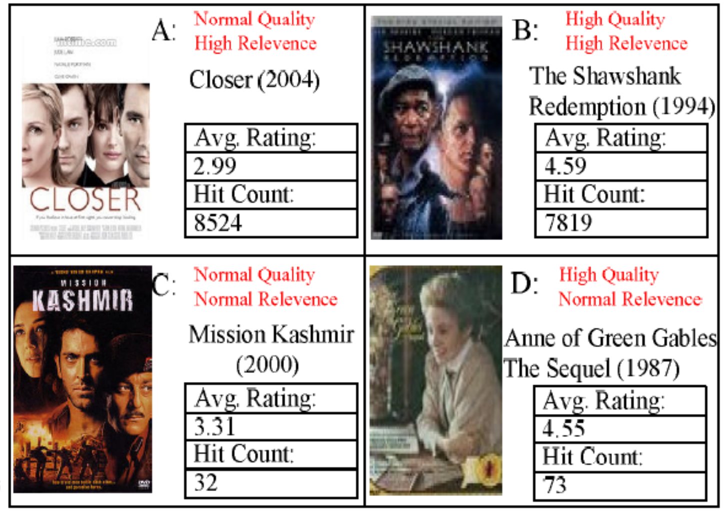




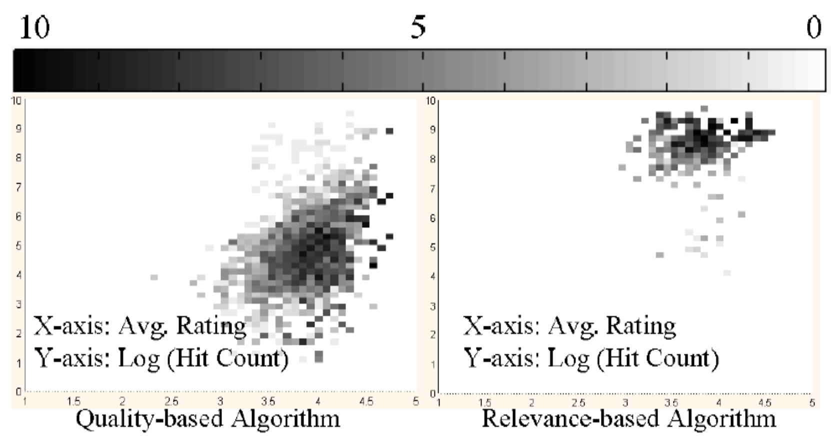
  - A user
    - may give a high rating to a classical movie for its good quality
    - but he/she might be more likely to watch a recent one that is more relevant and interesting to their lives, though the latter might be worse in quality

Incompleteness Limitation (Qualitative Analysis)





- Quality-based methods ignore Relevance
  - Type A is missing
  - Users may not show interests to visit some of the recommended items
- Relevance-based methods ignore quality
  - Type D is missing
  - Users will suffer from normal-quality recommended results



Incompleteness Limitation (Quantitative Analysis)

Table. Performance on quality-based NDCG

Methods	Given5			Given10			Given15		
	NDCG1	NDCG3	NDCG5	NDCG1	NDCG3	NDCG5	NDCG1	NDCG3	NDCG5
PMF	0.635	0.612	0.623	0.644	0.646	0.654	0.696	0.689	0.698
EigenRank	**0.698**	**0.685**	**0.679**	**0.699**	**0.696**	**0.698**	**0.713**	**0.707**	**0.719**
Assoc	0.529	0.542	0.560	0.597	0.593	0.595	0.615	0.610	0.627
Freq	0.642	0.600	0.596	0.636	0.607	0.610	0.638	0.618	0.632

Table. Performance on relevance-based NDCG

Methods	Given5			Given10			Given15		
	NDCG1	NDCG3	NDCG5	NDCG1	NDCG3	NDCG5	NDCG1	NDCG3	NDCG5
PMF	0.333	0.325	0.309	0.241	0.227	0.212	0.198	0.194	0.186
EigenRank	0.326	0.306	0.304	0.279	0.282	0.285	0.274	0.276	0.275
Assoc	0.518	0.484	0.467	0.466	**0.459**	**0.449**	**0.455**	**0.426**	**0.430**
Freq	**0.539**	**0.489**	**0.477**	**0.478**	0.429	0.412	0.428	0.377	0.364

- Quality-based algorithms
  - PMF [Salakhutdinov 2007], EigenRank [Liu 2008]
- Relevance-based algorithms
  - Assoc [Deshpande 2004], Freq [Sueiras 2007]

Our Solution: Combining Quality-based and Relevance-based Algorithms

- Fusion of quality-based and relevance-based algorithms
- Continuous-time MArkov Process (CMAP)
  - Integration-unnatural limitation
  - Quantity-missing limitation

Integrated Objective

- Normalized Discount Cumulated Gain (NDCG)
  - Both quality-based NDCG and relevance-based NDCG are accepted as practical measures in previous work [Guan 2009] [Liu 2008]
- Quality-based NDCG

$$NDCG_{P-quality} = \frac{1}{U}\sum_{u}^{U} Z_u \sum_{p=1}^{P} \frac{2^{r_{u,p}} - 1}{\log(1+p)}$$

- Relevance-based NDCG

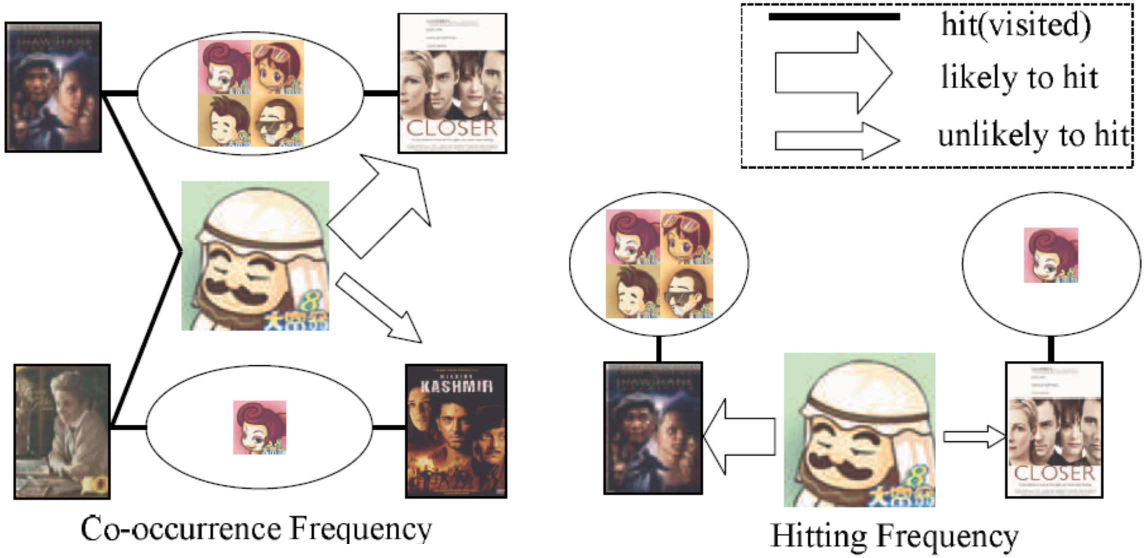
$$NDCG_{P-relevance} = \frac{1}{U}\sum_{u}^{U} Z_u \sum_{p=1}^{P} \frac{2^{h_{u,p}} - 1}{\log(1+p)}$$

- Integrated NDCG

$$NDCG_{I-linear} = \lambda * NDCG_Q + (1-\lambda) * NDCG_R$$

Fundamental Methods Selection

- Competitive quality-based method
  - EigenRank (random walk theory)
- Competitive relevance-based methods
  - Association-based methods [Deshpande 2004] (relational feature)
  - Frequency-based methods [Sueiras 2007] (local feature)



Combination Methods

- Linear Combination

$$S_{LinearComb} = w_1 F(EigenRank) + w_2 F(Assoc) + w_3 F(Hit - freq)$$

$$F(x) = 1/(1 + \exp(-x))$$

–Disadvantages: Incompatible values

- Rank Combination

$$BC_{RankComb} = w_1 BC_{EigenRank} + w_2 * BC_{Assoc} + w_3 * BC_{Hit-freq}$$

$$BC_{item} = 1/position(item)$$

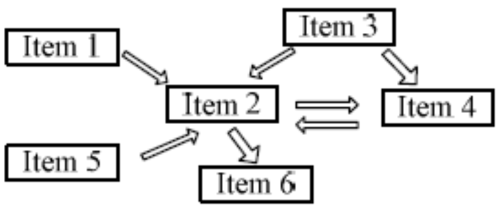
–Disadvantages: Missing quantity information

- Continuous-time MArkov Process (CMAP)

– Combination with an intuitive interpretation without missing quantity information

Continuous-time MArkov Process (CMAP)

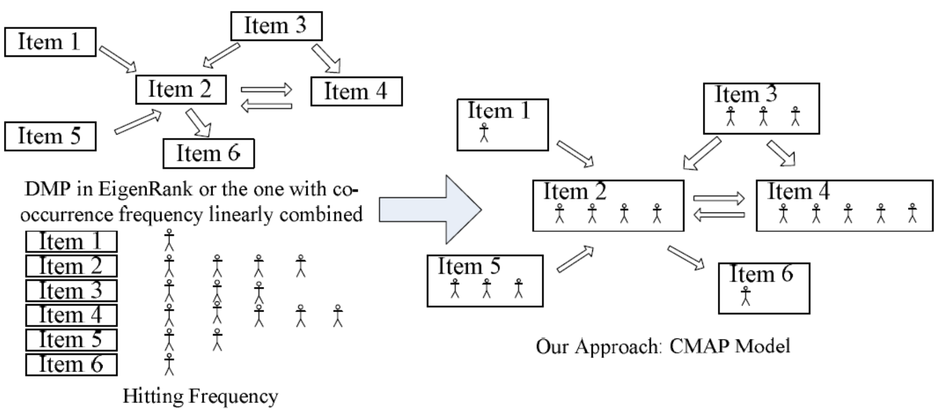
- Association feature (relational) combination



$$p'(j|i) = \frac{\xi(j,i)}{\sum_{j\in S}\xi(j,i)} \qquad \xi(i,j) = \frac{Freq(ij)}{Freq(i)*Freq(j)^{\beta}},$$

$$P_{new} = P*\alpha + P'*(1-\alpha).$$

- Frequency feature (local) combination



1) Customers' arrival follows the time-homogenous Poisson Process.

$$\begin{cases} P[N(t+\Delta t) - N(t) = 1] = \lambda\Delta t + o(\Delta t); \\ P[N(t+\Delta t) - N(t) > 1] = o(\Delta t). \end{cases}$$

2) Service time follows exponential distribution with the same service rate *u*.

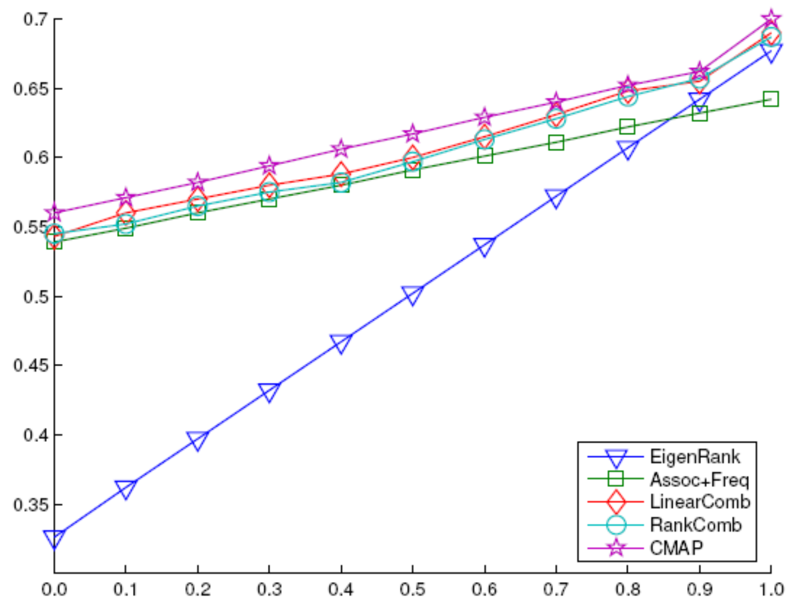
3) Waiting time of a customer on the condition that there is a queue:

$$P(T_g \le x|the\ queue\ exists) = 1 - exp^{-(u-\lambda)x}$$

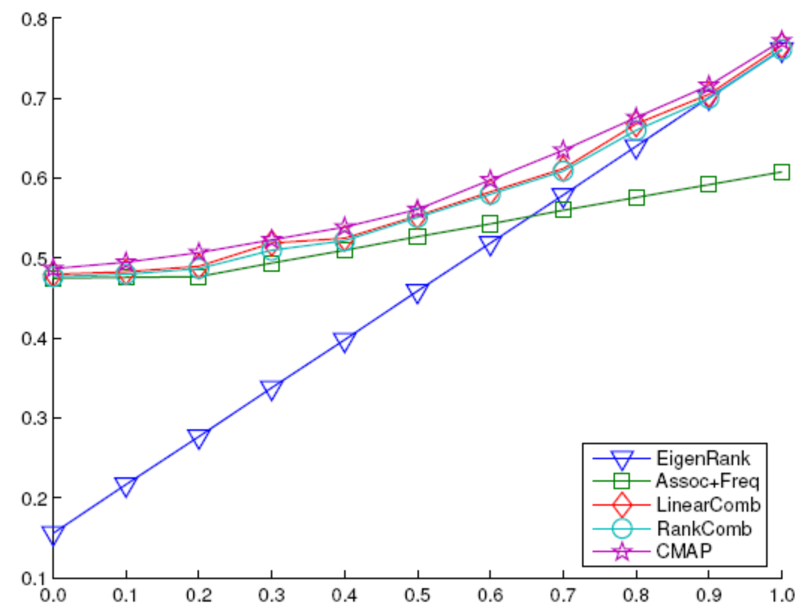
$$P[X_\tau = j|X_0 = i] = \frac{q_{ij}}{-q_{ii}} = P_{ij}*\alpha + P'_{ij}*(1-\alpha)$$

$$P(\tau > t|X_0 = i) = exp(-q_{ii}t)$$

Performance (Quantitative Analysis)



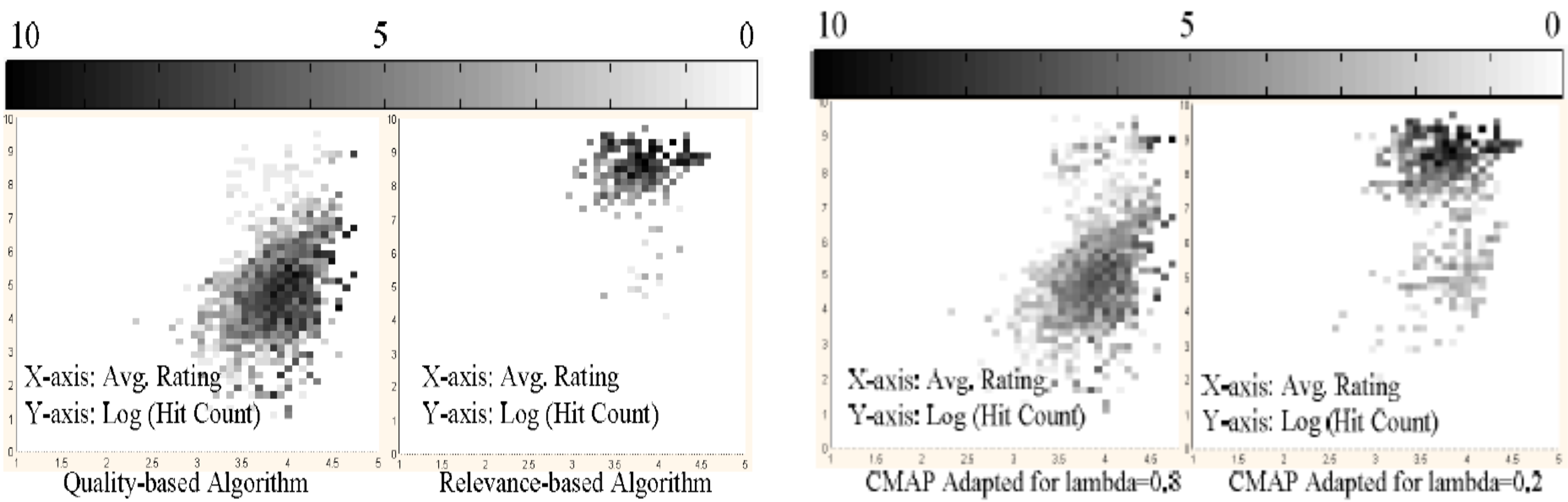
(a) MovieLens



(b) Netflix

- The three combination methods outperform the baselines in all the configurations
- The CMAP algorithm outperforms the other two fundamental combination methods

Performance (Qualitative Analysis)



Single-dimensional-adapted algorithms Multi-dimensional-adapted algorithms

- The incomplete problem can be well solved by the combination

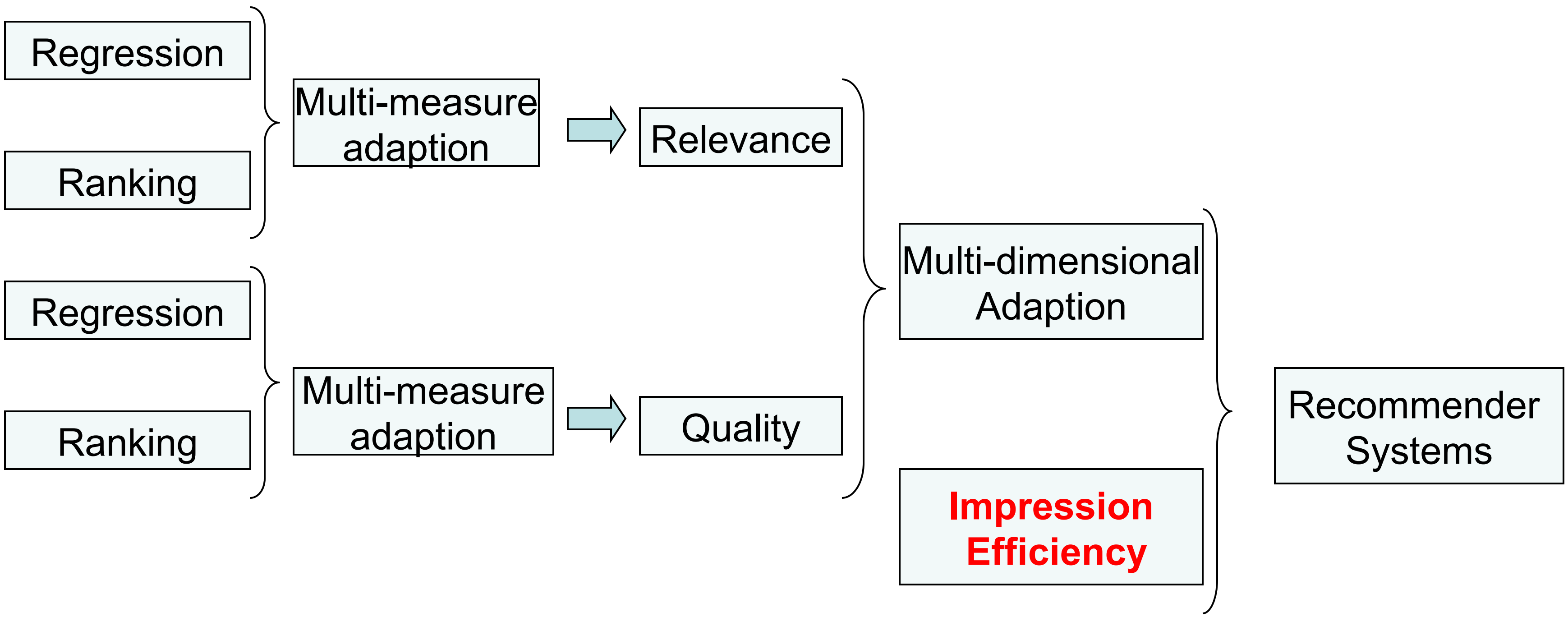
Summary of Part 3

- Incompleteness limitation identification of single-dimensional-adapted algorithms
- CMAP, as well as the other two novel approaches, in fusing quality-based and relevance-based algorithms
- Experimental verification on the effectiveness of CMAP

Outline

- Background of Recommender Systems
- Motivation of the Thesis
- Part 1: Relational Fusion of Multiple Features
- Part 2: Effective Fusion of Regression and Ranking
- Part 3: Effective Fusion of Quality and Relevance
- Part 4: Impression Efficiency Optimization
- Conclusion

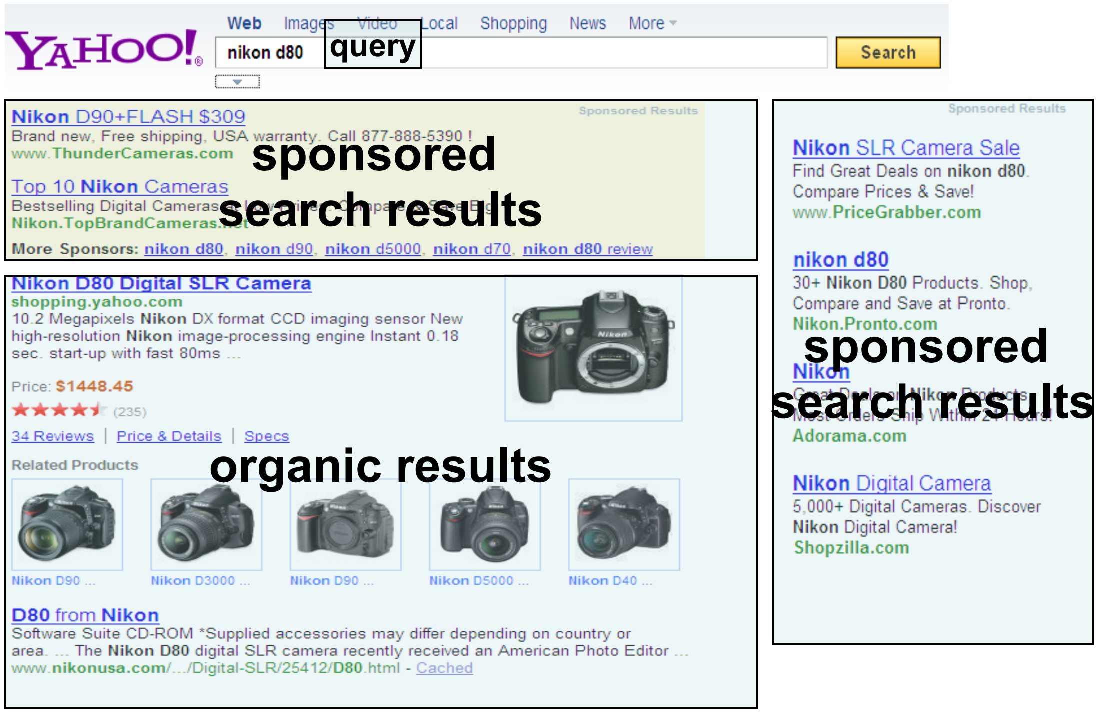
Part 4: Impression Efficiency Optimization



Impression Efficiency: How much revenue can be obtained by impressing a recommendation result?

- Limitation of previous work
  - The importance has been identified
  - But the issue has never been carefully studied

Commercial Intrusion from Over-quantity Recommendation in Sponsored Search

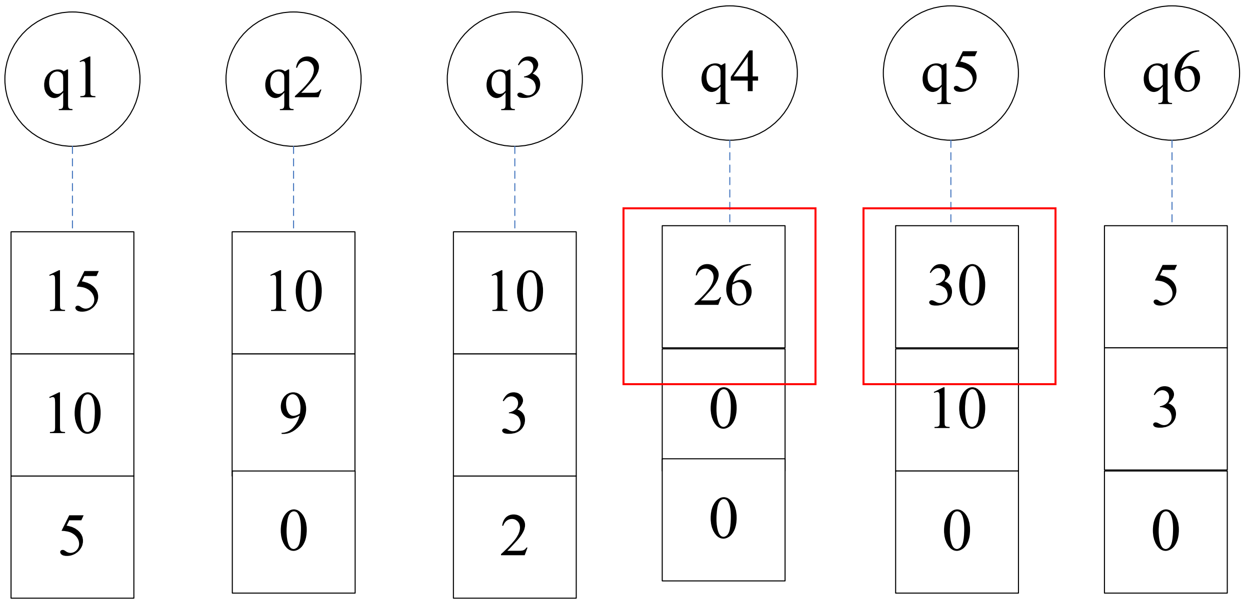


- Evidences of commercial intrusion
  - Users have shown bias against sponsored search results [Marable 2003]
  - More than 82% of users will see organic results first [Jansen 2006]
  - Organic results have obtained much higher click-through rate [Danescu-Niculescu-Mizil 2010]
  - Irrelevant ads will train the users to ignore ads [Buscher 2010]

Our Solution

- Formulate the task of optimizing impression efficiency
- We identify the unstable problem, which makes the static method not working well
- We propose a novel dynamic approach to solve the unstable problem

A Background Example



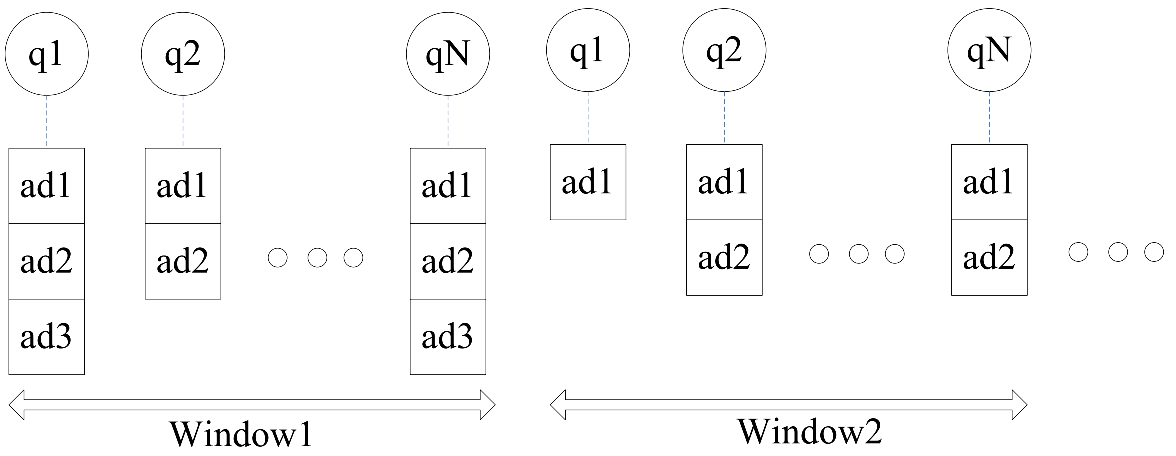
$N=6;\lambda=0.33$

$I(q_1) + I(q_2) + I(q_3) + I(q_4) + I(q_5) + I(q_6) <= N^*\lambda = 6^*0.33 = 2$

Best strategy:

$I(q_1) = 0;\ I(q_2) = 0;\ I(q_3) = 0;\ I(q_4) = 1;\ I(q_5) = 1;\ I(q_6) = 0$

Problem Formulation



- **Impression efficiency optimization:**

$$\max_{Imp(q)} \sum_{i=1}^{N} \sum_{j=1}^{I(q_i)} f(q_i, ad_j), \quad Sub.\ to \sum_{i=1}^{N} I(q_i) \le N * \lambda.$$

- **Evaluation metric**

$$error\ distance\ rate = \frac{\sum_{i=1}^{N} \sum_{j=1}^{I_{best}(q_i)} f(q_i, ad_j) - \sum_{i=1}^{N} \sum_{j=1}^{I(q_i)} f(q_i, ad_j)}{\sum_{i=1}^{N} \sum_{j=1}^{I_{best}(q_i)} f(q_i, ad_j)}$$

Experimental Setup

Sogou 搜狗

Pos.	Bid Term	Title	Body	Bid Price	Click
1	English eduction	English Education- Primary, Secondary (K-12) Education	US Diploma, Remote Online Learning	120	0
2	Learning English	Learn English Language	Online vocabulary booster, rewards 3rd grade classes use it free. Join	80	1
3	English learning	English School & Courses	Learn real English in the city of your dreams, contact us!	60	0

Table Summary of the Queries

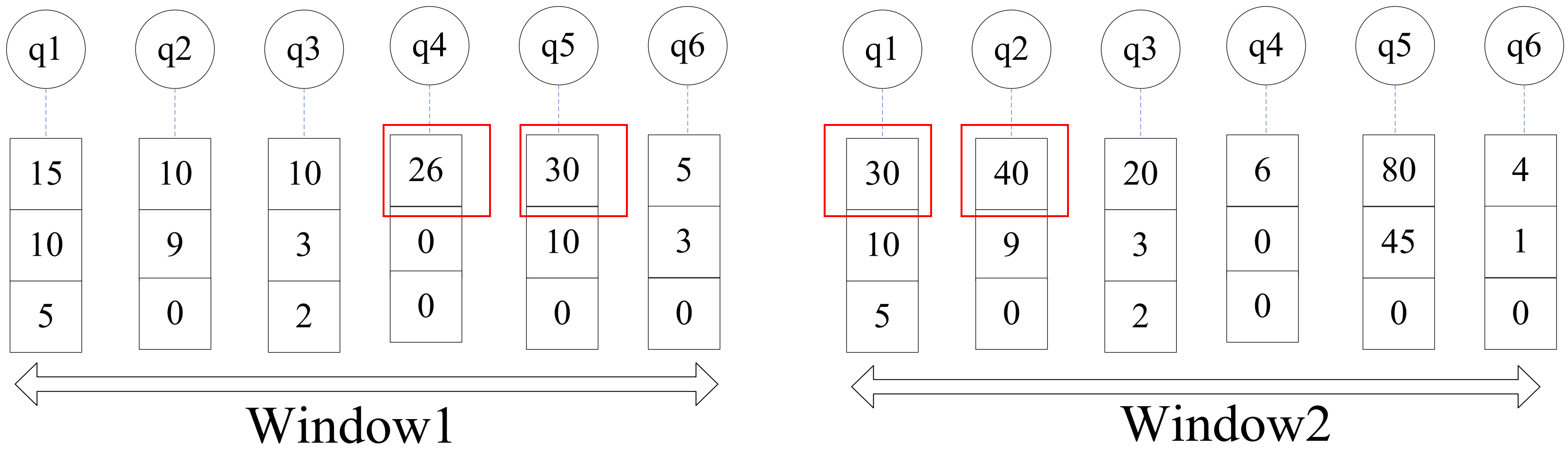
Query Freq.	# Unique Query	# Session	Avg CTR
1	1695146	1695146	0.0212
2	1058697	1342854	0.0153
3-4	772810	1275067	0.0141
5-8	262555	925065	0.0151
9-17	128304	880444	0.0162
18-32	47981	685582	0.0179
33-221480	48582	4427998	0.0201

Table Summary of the Ads

Ads Freq.	# Unique ad	# Impression	Avg CTR
1	26267	26267	0.0257
2	14689	29378	0.0222
3-4	17539	60146	0.0216
5-8	18058	113186	0.0208
9-17	18092	223102	0.0191
18-32	12786	306201	0.0190
33-82942	36365	17252570	0.0178

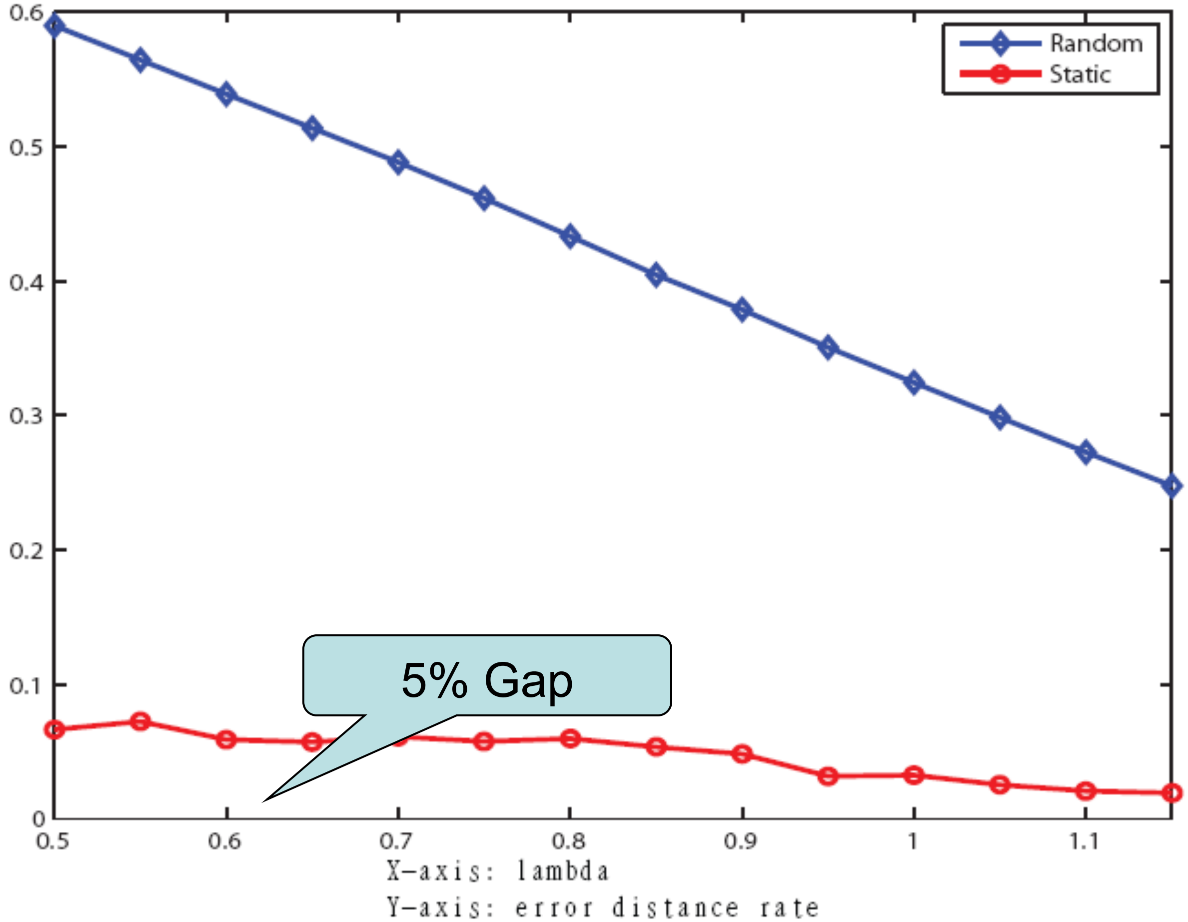
- The data is collected from Oct 1 2009 to December 1 2009
- CCM [Guo 2009] is employed as the revenue estimation function
- Assumption: for a query, the estimation revenue of the ad at the latter position is always smaller than that at the former position

The Static Method



- Case: N = 6, lambda = 0.33
- Static method process:
  1. Learn a threshold from previous window1 (threshold = 26)
  2. Select the elements that is larger than the threshold

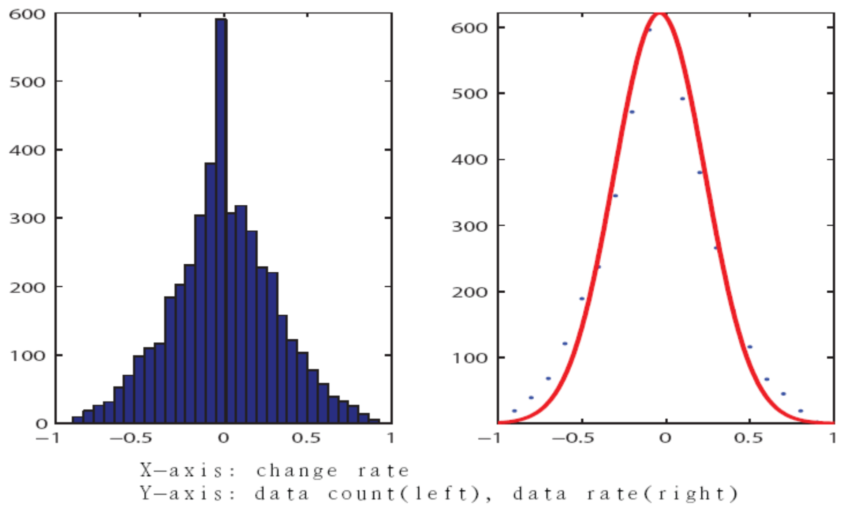
Performance of Static Method



- The static method outperforms random method significantly
- There is a 5% gap of the static method from the best strategy

The Unstable Problem

- The data is not stable and is changing over time
- Verification through statistics in the real world dataset
  - The change of average revenue
  - The change of threshold
  - The change of query distribution
  - The change of click-through rate (CTR)

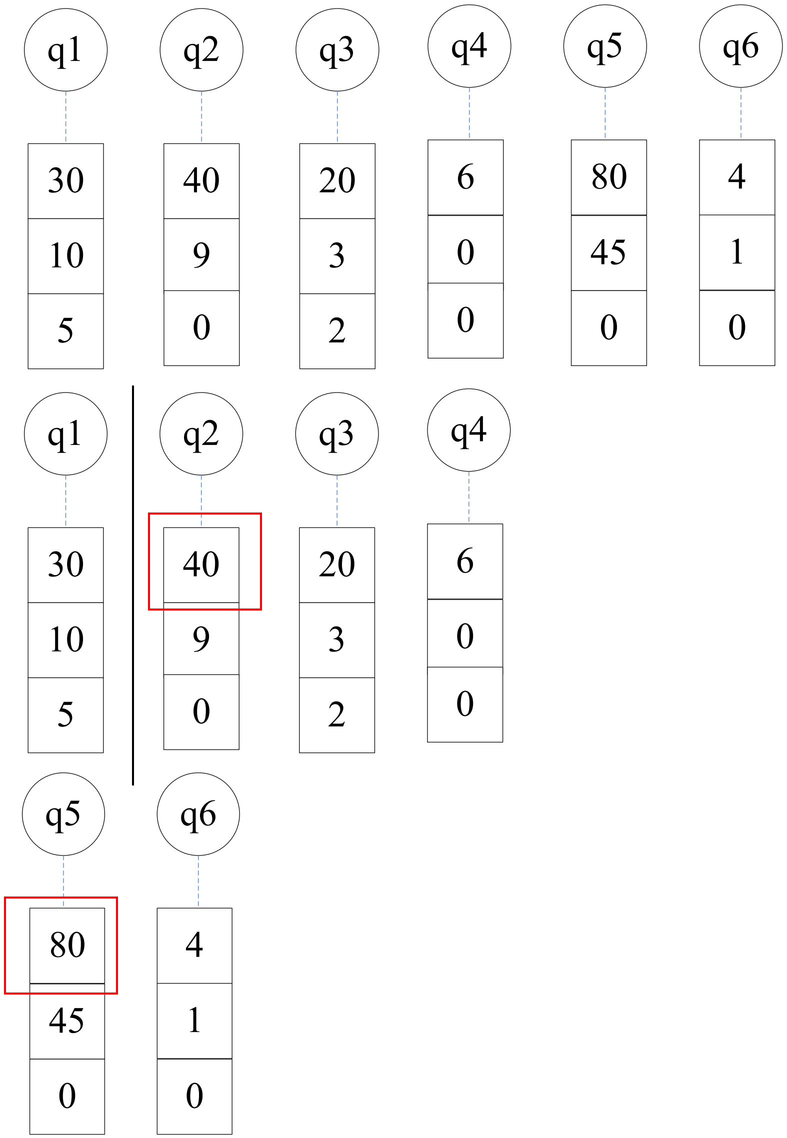


Change of CTR

•It follows a Gaussian distribution

•More than 20% cases, the data has changed for more than 20%

The Dynamic Method

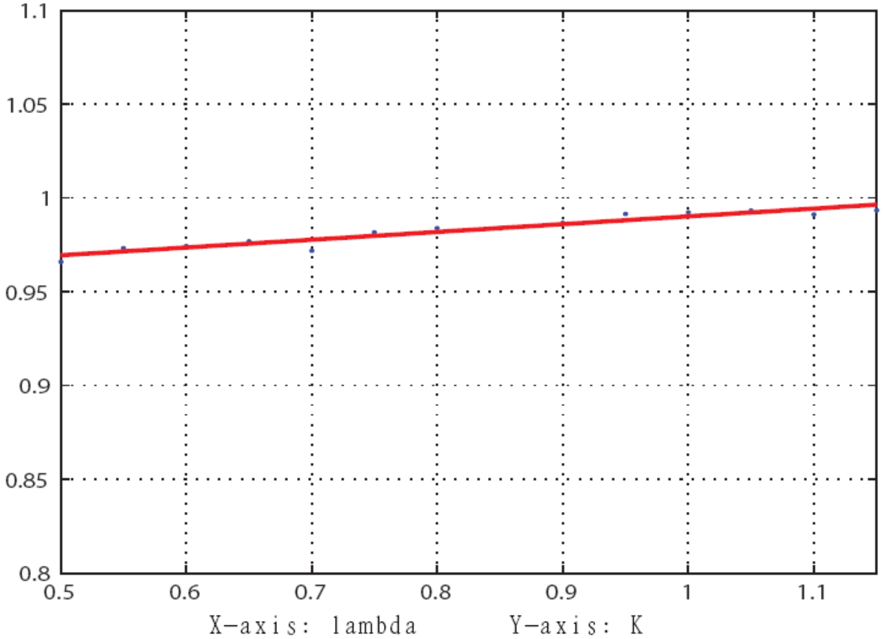


Case: N = 6, lambda = 0.333
Select 2 items from 6 queries

1) Generate a division from B(6,0.5)=4
2) Observe the first 4/e=1 group
3) Set the threshold = 30
4) Select the first element that is larger than 30
5) 40 would be selected
6) Reset the threshold to 40
7) Select the first element that is larger than 40
8) 80 would be selected
9) Output: 40+80 = 120

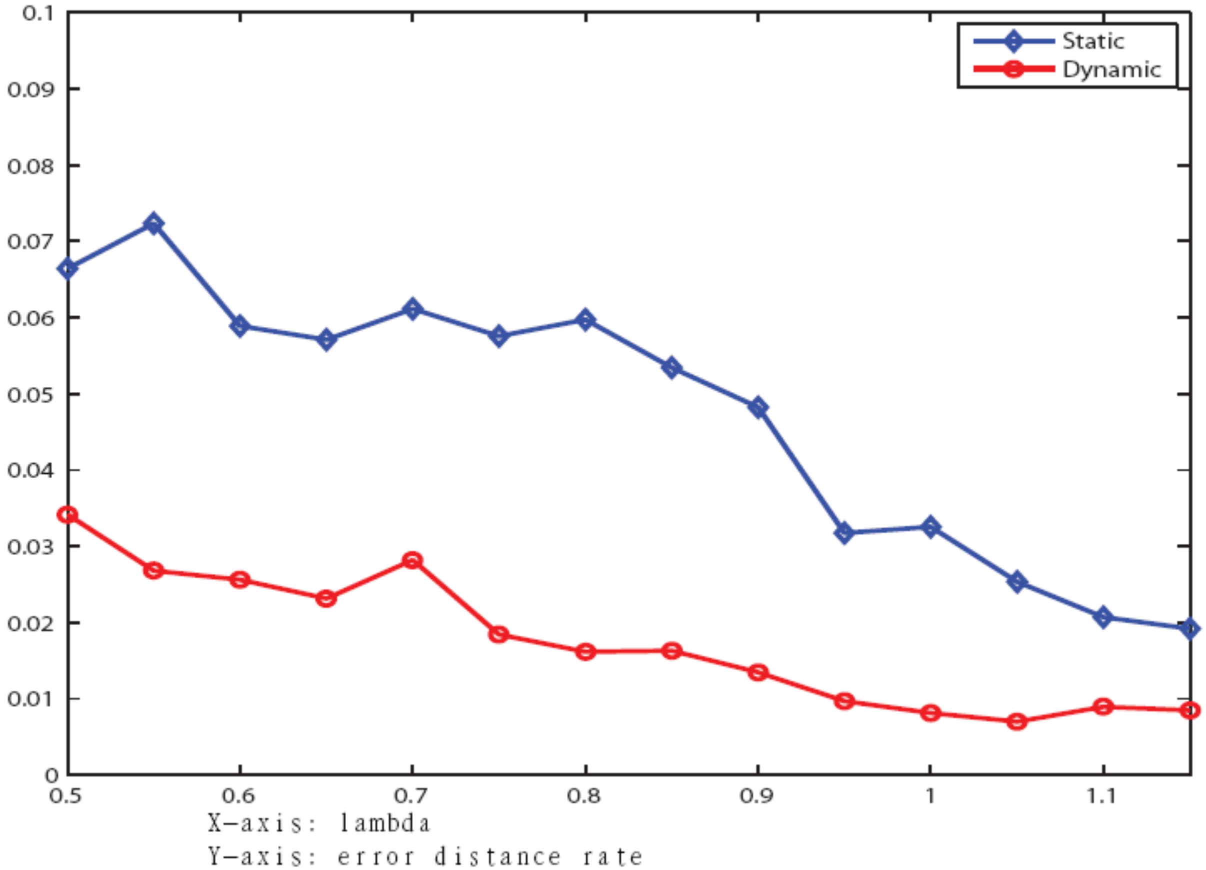
Empirical Analysis of the Dynamic Algorithm

- Competitive ratio:
  - the ratio between the algorithm's performance and the best performance



- The competitive rate in all configurations is above 0.97

Performance of The Dynamic Algorithm



•The dynamic algorithm outperforms the static algorithm significantly

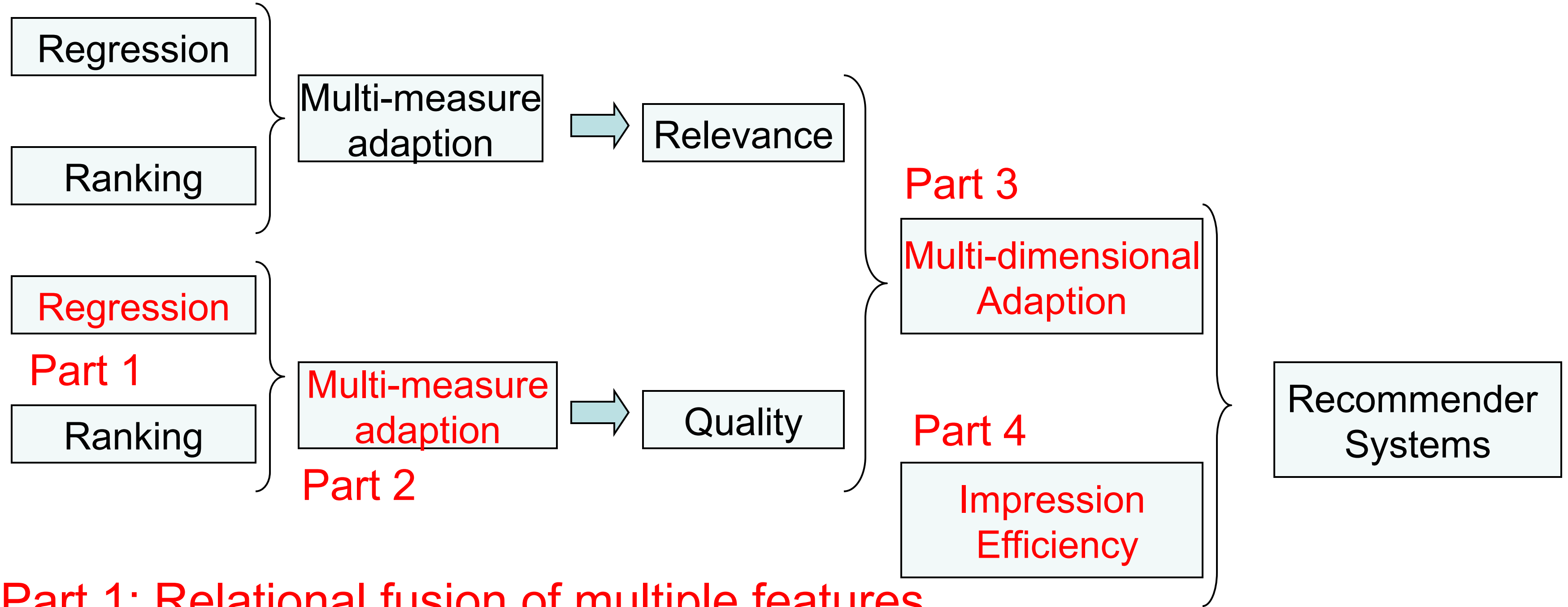
Summary of Part 4

- Impression efficiency optimization formulation
- Unstable limitation identification in static methods
- We propose a dynamic algorithm
- Significant improvement

Outline

- Background of Recommender Systems
- Motivation of the Thesis
- Part 1: Relational Fusion of Multiple Features
- Part 2: Effective Fusion of Regression and Ranking
- Part 3: Effective Fusion of Quality and Relevance
- Part 4: Impression Efficiency Optimization
- Conclusion

Conclusion



- Part 1: Relational fusion of multiple features
  - Relational dependency is ignored
  - Difficulty in learning feature weights
- Part 2: Fusion of regression and ranking for multi-measure adaption
  - Single-measure-adapted algorithms cannot adapt to multiple measures
- Part 3: Fusion of relevance and quality for multi-dimensional adaption
  - Single-dimensional algorithms cannot adapt to multiple dimensions
- Part 4: Impression efficiency optimization
  - The unstable problem in the static method

Q & A

- Thanks very much to my dearest supervisors, thesis committees and colleagues in the lab for your great help!

Appendix (1) Publications

- Ph.D study
  - Xin Xin, Michael R. Lyu, and Irwin King. *CMAP: Effective Fusion of Quality and Relevance for Multi-criteria Recommendation* (Full Paper). In Proceedings of ACM 4th International Conference on Web Search and Data Mining (WSDM 2011), Hong Kong, February 2011.
  - Xin Xin, Irwin King, Hongbo Deng, and Michael R. Lyu. *A Social Recommendation Framework Based on Multi-scale Continuous Conditional Random Fields* (Full and Oral Paper). In Proceedings of ACM 18th Conference on Information and Knowledge Management (CIKM 2009), Hong Kong, November 2009.
- Previous work
  - Xin Xin, Juanzi Li, Jie Tang, and Qiong Luo. Academic Conference Homepage Understanding Using Hierarchical Conditional Random Fields (Full and Oral Paper). In Proceedings of ACM17th Conference on Information and Knowledge Management (CIKM 2008), Napa Valley, CA, October 2008.
  - Xin Xin, Juanzi Li, and Jie Tang. Enhancing SemanticWeb by Semantic Annotation: Experiences in Building an Automatic Conference Calendar (Short Paper). In Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2007), Fremont, CA, November 2007.

Appendix (2) Unpublished Work

- Xin Xin, Haiqin Yang, Michael R. Lyu, and Irwin King. *Combining Regression and Ranking in Collaborative Filtering*. Submitted to CIKM 2011.

- Xin Xin, Wei Wang, Wei Yu, Jie Tang, Irwin King and Michael R. Lyu. *Learning to Impress in Sponsored Search*. Preparing to submit it to WWW 2012.

- Xin Xin, Michael R. Lyu, and Irwin King. *Relational Fusion-based Framework for Recommender Systems*. Preparing to submit it to TOIS.

- Wei Wang, Xin Xin, Irwin King, Jie Tang, and Michael R. Lyu. *Compete or Collaborate? Incorporating Relational Influence within Search Results into Click Models in Sponsored Search*. Submitted to CIKM 2011.

Appendix (3) MAE and RMSE Definition

- Mean Absolute Error (MAE)

$$MAE = \frac{\sum |R_{u,i} - \tilde{R}_{u,i}|}{N}$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\sum (R_{u,i} - \tilde{R}_{u,i})^2}{N}}$$

$\tilde{R}_{u,i}$ is the predicted ratings of item i by user u, $R_{u,i}$ is the ground truth, and N is the total number of testing predictions.

Appendix (4) NDCG Definition

- Normalized Discount Cumulated Gain (NDCG)

$$NDCG_{P-quality} = \frac{1}{U}\sum_{u}^{U} Z_u \sum_{p=1}^{P} \frac{2^{r_{u,p}} - 1}{\log(1+p)}$$

  - *U* is the number of test users.
  - *Z* is the normalization factor of a single user.
  - *P* is the position
  - $r_{u,p}$ is the rating of user u at position p.
- Example
  - Ideal rank: 3, 2, 1. Value1=Z=7/1+3/log(3)+1/log(4).
  - Current rank: 2, 3, 1. Value2=3/1+7/log(3)+1/log(4).
  - NDCG = Value2/Value1.

Appendix (5) PCC Definition

- Pearson Correlation Coefficient (PCC)

$$Sim(a,u) = \frac{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)^2} \sqrt{\sum_{i \in I(a) \cap I(u)} (r_{u,i} - \overline{r}_u)^2}}$$

  - *a* and *u* are two uses.
  - *I(a)* are the items user *a* has rated.
  - $r_{a,i}$ is the rating of item *i* by user *a*.
  - $\overline{r}_a$ is the average rating of user *a*.

Appendix (5) KRCC Definition

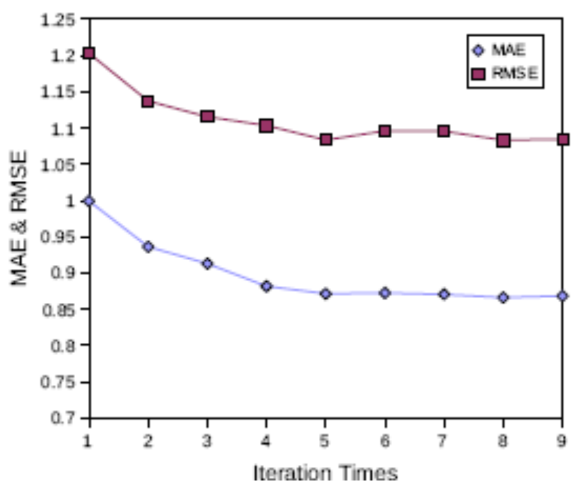
- Kendall Rank Correlation Coefficient (KRCC)

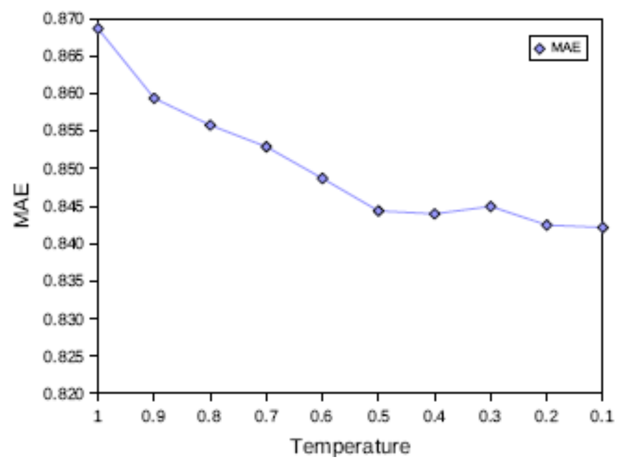
$$S_{u,v} = 1 - \frac{4 * \sum_{i,j \in I_u \cap I_v} I^{-}((r_{u,i} - r_{u,j})(r_{v,i} - r_{v,j}))}{|I_u \cap I_v|(|I_u \cap I_v| - 1)}$$

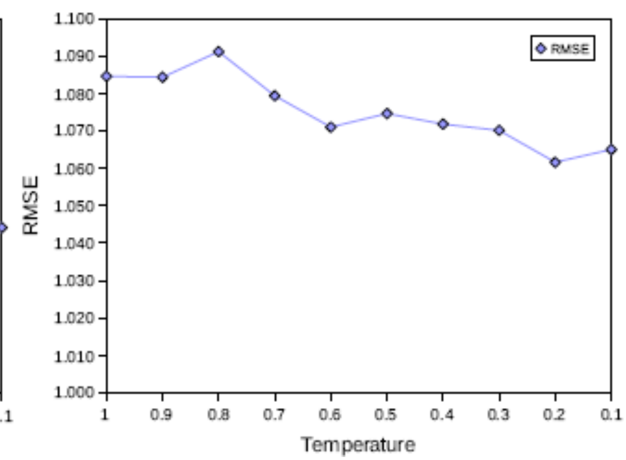
  - i_u is the item set for user *u*.
  - I^{-} is the indicating function.

Appendix (6) Complexity Analysis for MCCRF

- For each user-item pair, the calculation complexity is
  - *O(#feature * #neighbor * #iteration)*

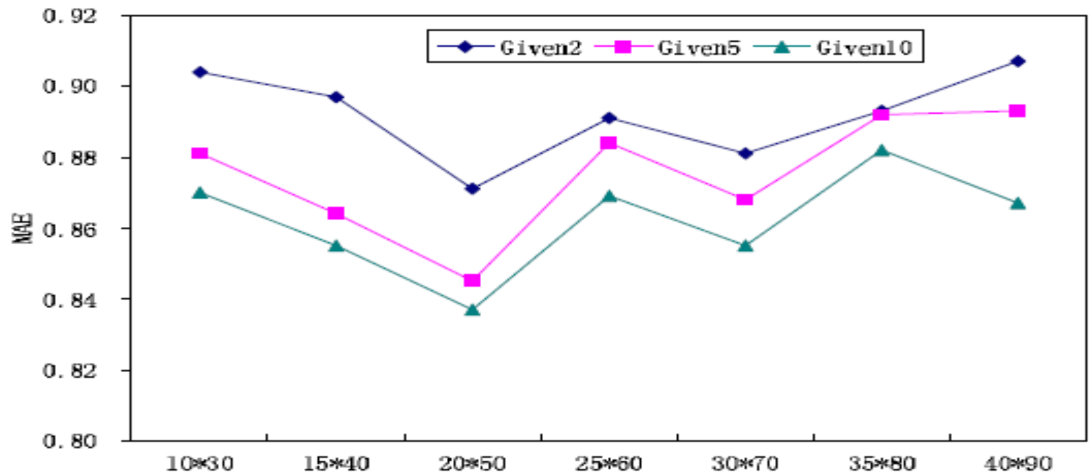


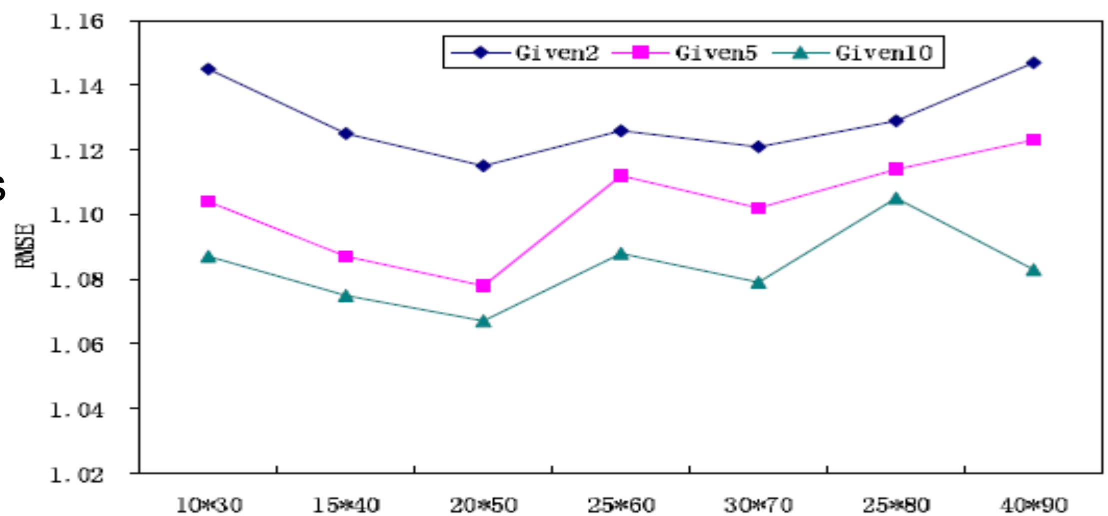




Appendix (7) Cluster Method in Large Datasets for MCCRF

- To run the whole data will take too much memory in large datasets like Epinions
- Xue et al 2005 propose to employ cluster methods to solve this problem
- The figures show the impact of cluster size in Epinions using K-means cluster methods





Appendix (8) Complexity Analysis for Model-based Combination of Regression and Ranking

- For each user-item pair, the calculation complexity is
  - *O(#observation * #latent feature)*

Appendix (9) Complexity Analysis for Memory-based Combination of Regression and Ranking

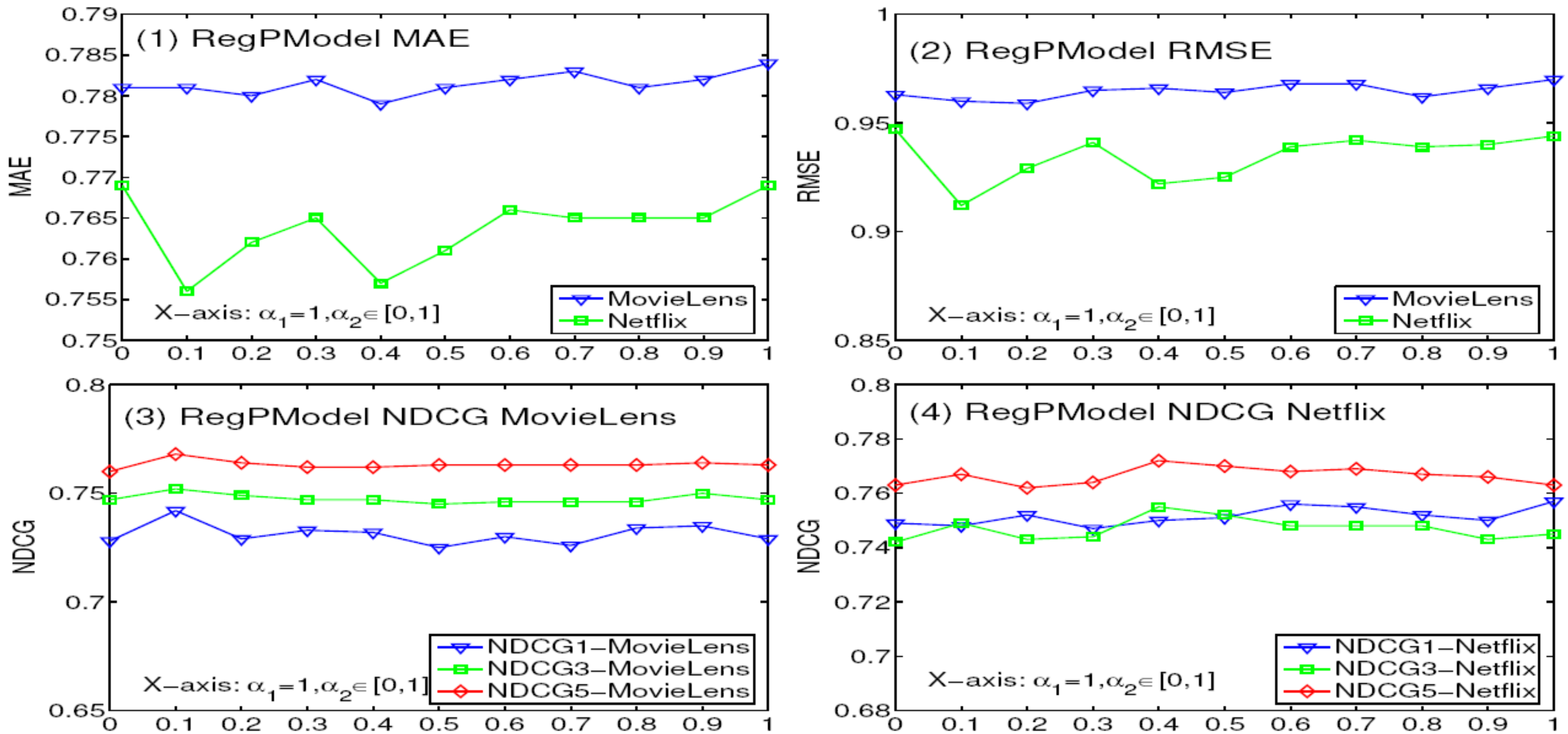
- Similarity calculation
  - PCC
    - *O(#user * #item * #common item)*
  - KRCC
    - *O(#user * #item * #item * #common item)*
- Rating calculation
  - *O(1)*
- Stationary distribution calculation
  - *O(#items * #iteration)*

Appendix (10) Complexity Analysis for CMAP

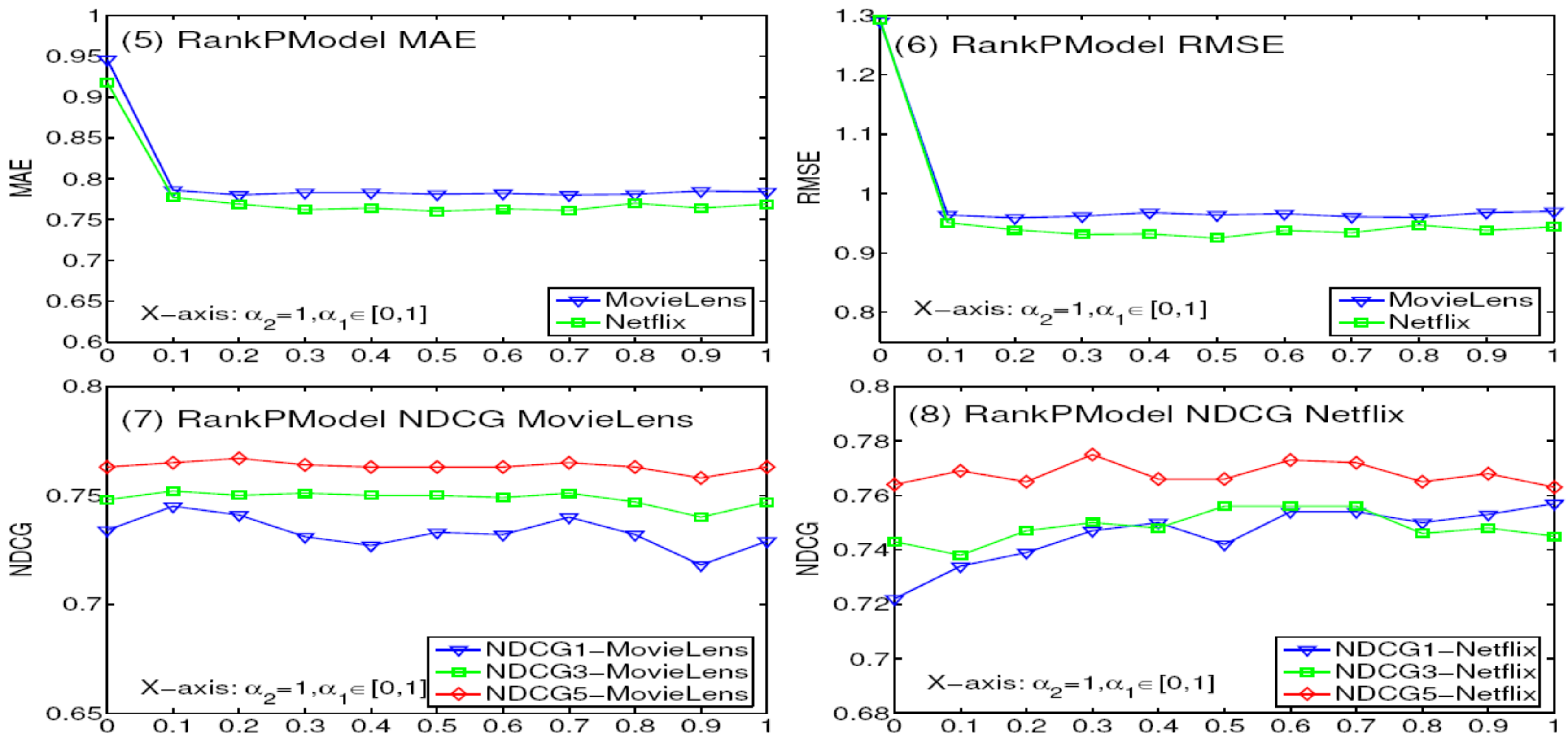
- Stationary distribution calculation.
  - *O(#item * #iteration)*

$$\begin{cases} \pi_i = \dfrac{\frac{\widetilde{\pi}_i}{-q_{ii}}}{\sum_{j=1}^{S} \frac{\widetilde{\pi}_j}{-q_{jj}}}; \\ \\ \widetilde{\pi}_j = \sum_{i \in S} \widetilde{\pi}_i \dfrac{q_{ij}}{-q_{ii}}. \end{cases}$$

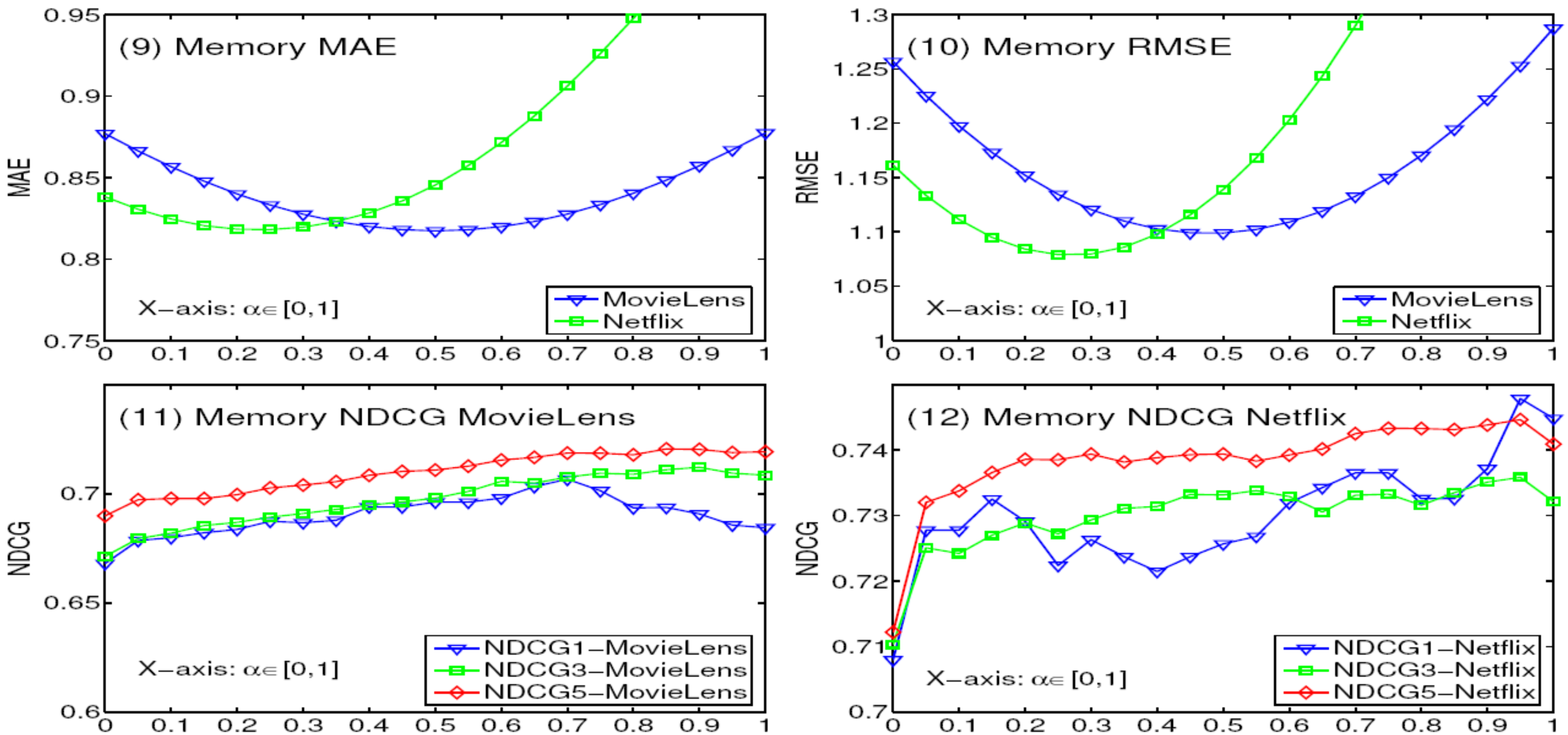
Appendix (11) Sensitivity Analysis for Model-based Regression-prior Combination



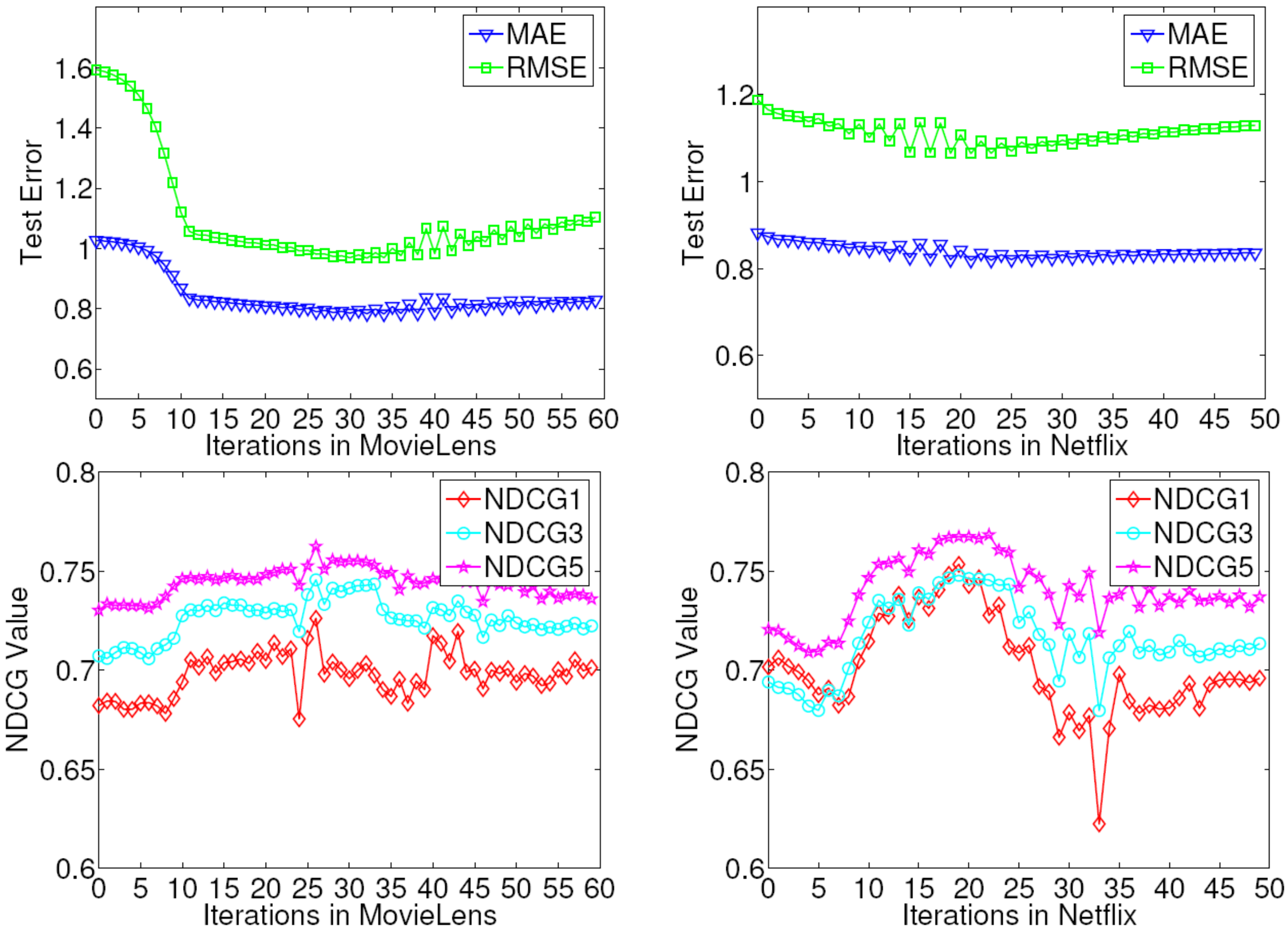
Appendix (12) Sensitivity Analysis for Model-based Ranking-prior Combination



Appendix (13) Sensitivity Analysis for Memory-based Combination

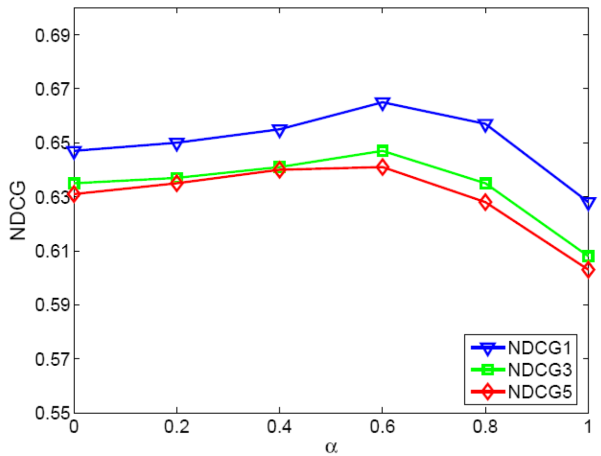


Appendix (14) Convergence in Model-based Combination

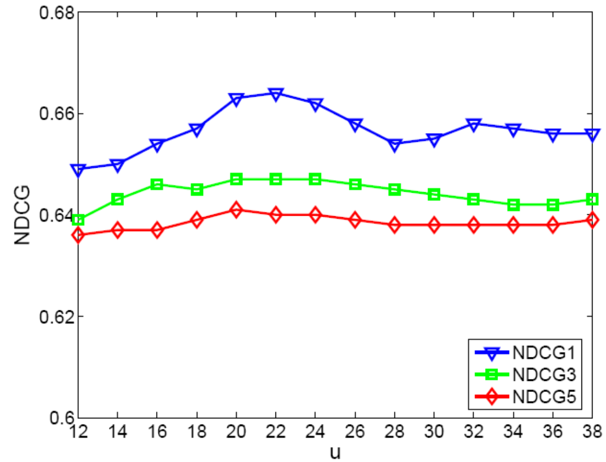


Appendix (15) Sensitivity Analysis for CMAP

– MovieLens

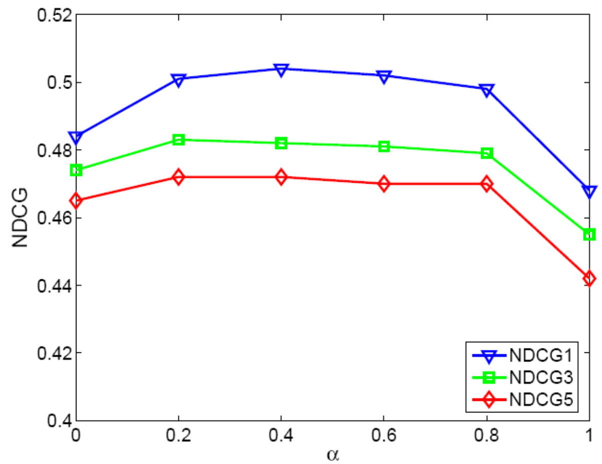


(a) Impact of α

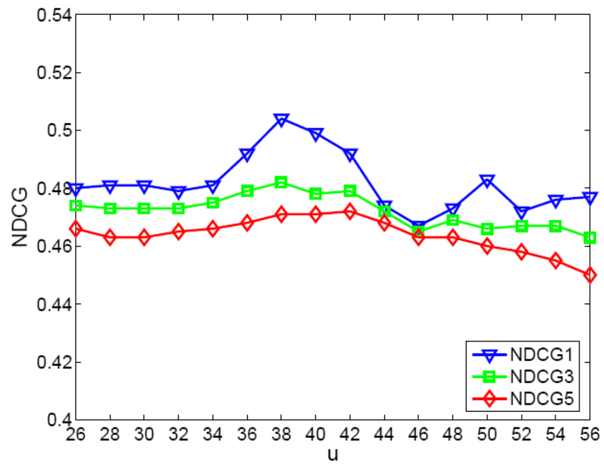


(b) Impact of u

– Netflix



(a) Impact of α



(b) Impact of u