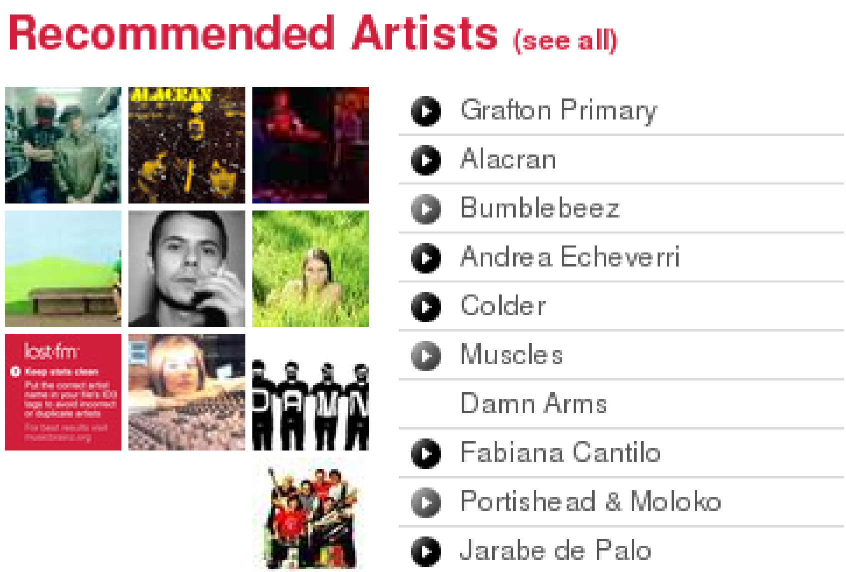
Recommending Interest Groups to Social Media Users by Incorporating Heterogeneous Resources

Wei Zeng and Li Chen
(lichen@comp.hkbu.edu.hk)
Department of Computer Science
Hong Kong Baptist University

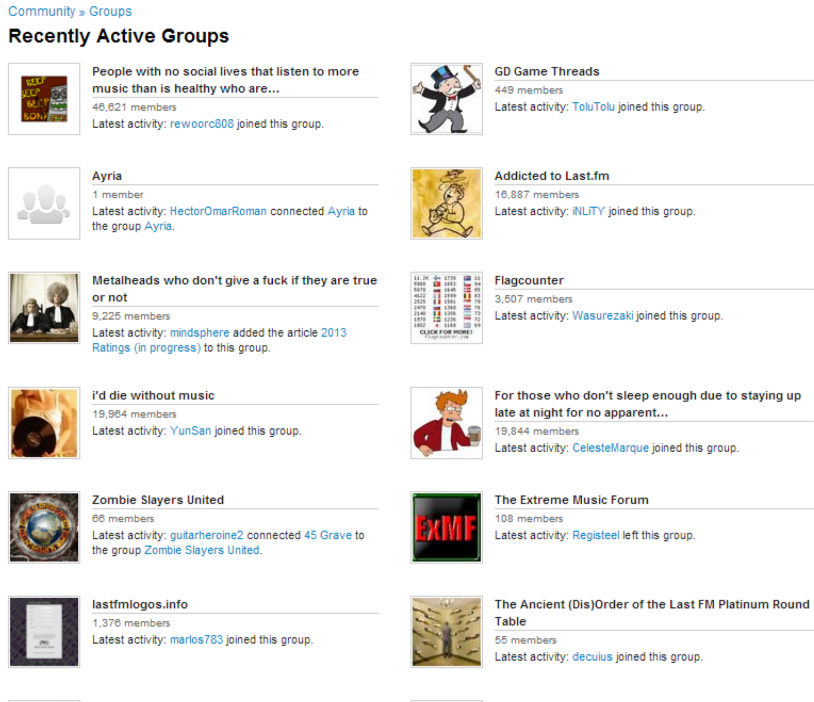
Motivation

Traditional recommendations



Our focus: recommending interest groups

Groups are based on a common interest, an artist or genre, or anything really!



Challenge

- Data sparsity problem
  - Last.fm, 100,000 users, user-item pairs **29,908,020**, but user-group pairs **1,132,281**

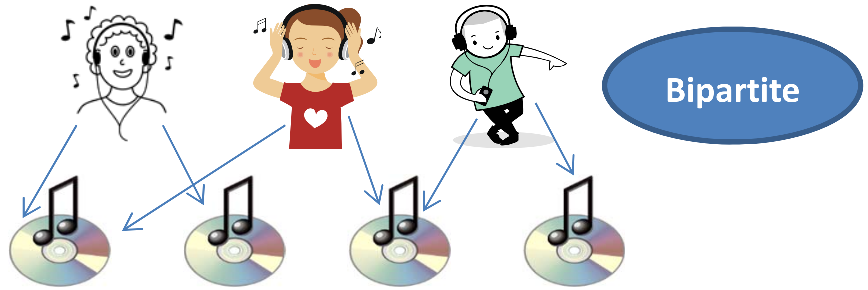
- Related work
  - **Community/affiliation recommendation** based on graph proximity model (Vasuke et al. RecSys'10), combinational collaborative filtering (Chen et al., KDD'08), or Latent Dirichlet Allocation (Chen et al., WWW'09)

- ***But***, few have fully incorporated other available resources to further increase the recommendation accuracy

Our methodology

- To fuse two auxiliary resources
  - User-user friendship

  - User-item preferences



- Research questions
  - *How to fuse, due to the different properties?*
  - *Which resource takes more effective effect?*
  - *What about their combined effect?*

Algorithm

- Fusion framework: **Matrix Factorization**
  - *Advantages*: scalability, efficiency, potential accuracy
- To fuse friendship
  - Regularization model
  - *Advantage*: for minimizing the gap between two entities
- To fuse user-item preferences
  - Factorization model
  - *Advantage*: for effectively factorizing user-item relations into two components

To fuse friendship

Basic matrix factorization of user-group matrix

equals 1 if the user *u* joined group *g*

$$\min_{u*,g*} \sum_{u,g} c^*_{ug}(\underline{p^*_{ug}} - x_u^T z_g)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_g \| z_g \|^2)$$

$$+\underline{\lambda_f}(\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \underline{\widehat{sim}(u,f)} x_f \|^2)$$

Regularization of user-user friendship

coefficient for the friendship regularization

normalized similarity degree between the user *u* and her/his friend *f*, based on common items, common groups, or common friends

To fuse user-item preferences

Basic matrix factorization of user-group matrix

$$\alpha \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_g \| z_g \|^2) +$$

$$(1-\alpha) \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_i \| y_i \|^2)$$

Factorization of user-item matrix

equals 1 if the user *u* clicked item *i* (implicit feedback)

used to adjust the relative weights of user-item matrix

Cont.

- Alternatively, for the comparison purpose

$$\min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \parallel x_u \parallel^2 + \sum_g \parallel z_g \parallel^2)$$

$$\boxed{+\lambda_f(\parallel x_u - \frac{1}{N(u)} \sum_{n \in N(u)} \omega^*_{un} * x_n \parallel^2)}$$

user *u's* neighbors who have common items with *u*

weight of similarity between two users *u* and *n,* based on their common items

Regularization of user-item relation

To fuse them together

Basic matrix factorization of user-group matrix

$$\alpha \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_g \| z_g \|^2) +$$

$$\lambda_f (\| x_u - \frac{1}{| F(u) |} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \|^2) +$$

$$(1-\alpha) \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_i \| y_i \|^2)$$

Factorization of user-item matrix

Regularization of user-user friendship

Experiment

	Element	Size	Element	Size
Last.fm	#user	100,000	#user-item pair	29,908,020
	#item	22,443	#friendship pair	583,621
	#group	25,397	#user-group pair	1,132,281
Douban	#user	71,034	#user-item pair	12,292,429
	#item	25,258	#friendship pair	273,832
	#group	2,973	#user-group pair	373,239

Evaluation: leave-one-out, **hit-ratio metric**

Results – fusing user-item preferences

	Last.fm		*Douban*	
Method	**Hits@5**	**Hits@10**	**Hits@5**	**Hits@10**
Group.MF (baseline)	0.0530	0.0875	0.1995	0.2933
Fusing user-item preferences (via Factorization)				
Group.MF.I.F@train.20	0.0573	0.0899	0.2030	0.2950
Group.MF.I.F@train.40	0.0678	0.1026	0.2102	0.3013
Group.MF.I.F@train.60	0.0714	0.1068	0.2113	0.3079
Group.MF.I.F@train.80	**0.0722**	**0.1070**	**0.2120**	**0.3095**
Fusing user-item preferences (via Regularization)				
Group.MF.I.R@train.20	0.0559	0.0885	0.2025	0.2932
Group.MF.I.F@train.40	0.0559	0.0885	0.2026	0.2936
Group.MF.I.R@train.60	0.0560	0.0886	0.2026	0.2936
Group.MF.I.R@train.80	0.0561	0.0887	0.2027	0.2937
Fusing friendship				
Group.MF.F.R	0.0566	0.0910	0.2072	0.2973
Group.MF.F.F	0.0553	0.0876	0.2038	0.2928
Group.MF.F.FCos	0.0549	0.0861	0.2075	0.2974
Group.MF.F.GCos	**0.0593**	**0.0923**	**0.2093**	**0.2999**
Group.MF.F.ICos	0.0569	0.0897	0.2062	0.2921

Result 1: the accuracy of factorization model (Group.MF.I.F) is improved with the increase of the density

Result 2: the accuracy of regularization model (Group.MF.I.R) is lower and does not obviously change when the data density varied

Results – fusing user-user friendship

	Last.fm		*Douban*	
Method	**Hits@5**	**Hits@10**	**Hits@5**	**Hits@10**
Group.MF (baseline)	0.0530	0.0875	0.1995	0.2933
Fusing user-item preferences (via Factorization)				
Group.MF.I.F@train.20	0.0573	0.0899	0.2030	0.2950
Group.MF.I.F@train.40	0.0678	0.1026	0.2102	0.3013
Group.MF.I.F@train.60	0.0714	0.1068	0.2113	0.3079
Group.MF.I.F@train.80	**0.0722**	**0.1070**	**0.2120**	**0.3095**
Fusing user-item preferences (via Regularization)				
Group.MF.I.R@train.20	0.0559	0.0885	0.2025	0.2932
Group.MF.I.F@train.40	0.0559	0.0885	0.2026	0.2936
Group.MF.I.R@train.60	0.0560	0.0886	0.2026	0.2936
Group.MF.I.R@train.80	0.0561	0.0887	0.2027	0.2937
Fusing friendship				
Group.MF.F.R	0.0566	0.0910	0.2072	0.2973
Group.MF.F.F	0.0553	0.0876	0.2038	0.2928
Group.MF.F.FCos	0.0549	0.0861	0.2075	0.2974
Group.MF.F.GCos	**0.0593**	**0.0923**	**0.2093**	**0.2999**
Group.MF.F.ICos	0.0569	0.0897	0.2062	0.2921

Result 3: the regularization model (Group.MF.F.R) outperforms the factorization model (Group.MF.F.F)

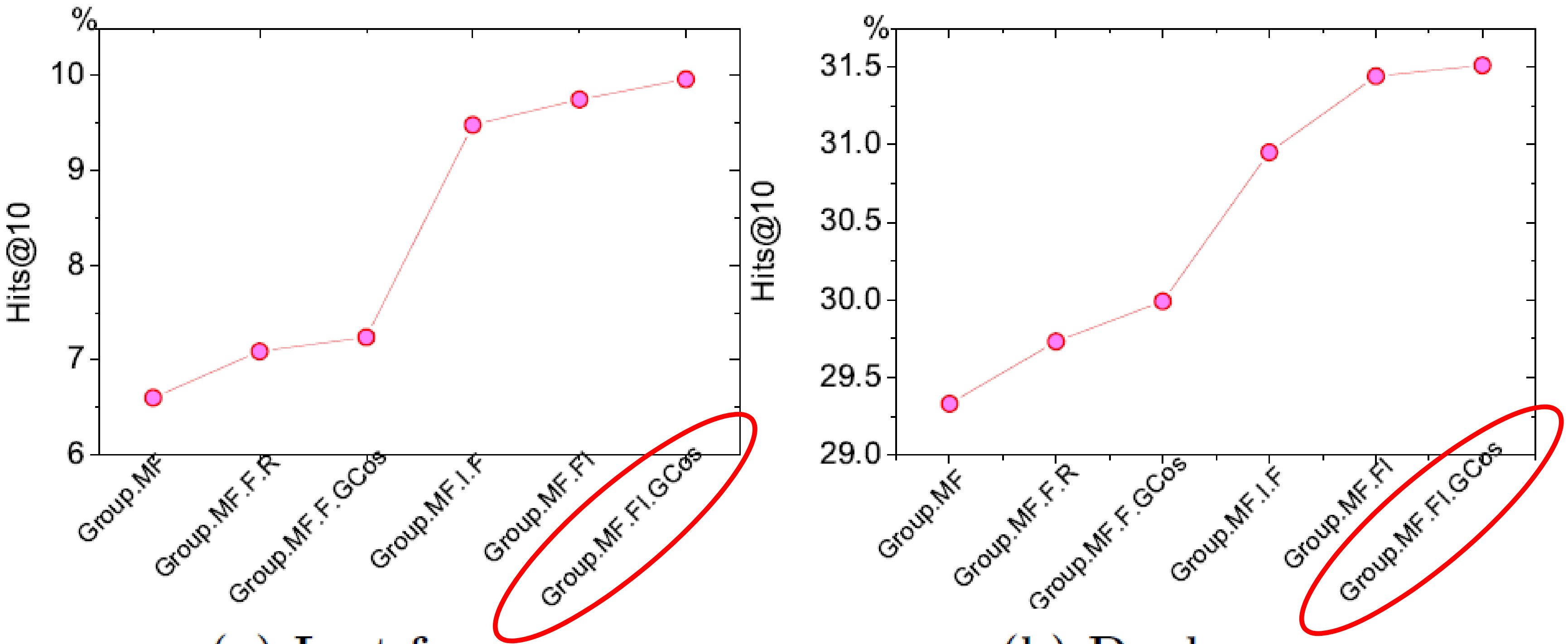
Result 4: the integration of group-based similarity measure (Group.MF.F.GCos) outperforms the others

Results – comparison of the two resources

	Last.fm		*Douban*	
Method	**Hits@5**	**Hits@10**	**Hits@5**	**Hits@10**
Group.MF (baseline)	0.0530	0.0875	0.1995	0.2933
Fusing user-item preferences (via Factorization)				
Group.MF.I.F@train.20	0.0573	0.0899	0.2030	0.2950
Group.MF.I.F@train.40	0.0678	0.1026	0.2102	0.3013
Group.MF.I.F@train.60	0.0714	0.1068	0.2113	0.3079
Group.MF.I.F@train.80	**0.0722**	**0.1070**	**0.2120**	**0.3095**
Fusing user-item preferences (via Regularization)				
Group.MF.I.R@train.20	0.0559	0.0885	0.2025	0.2932
Group.MF.I.F@train.40	0.0559	0.0885	0.2026	0.2936
Group.MF.I.R@train.60	0.0560	0.0886	0.2026	0.2936
Group.MF.I.R@train.80	0.0561	0.0887	0.2027	0.2937
Fusing friendship				
Group.MF.F.R	0.0566	0.0910	0.2072	0.2973
Group.MF.F.F	0.0553	0.0876	0.2038	0.2928
Group.MF.F.FCos	0.0549	0.0861	0.2075	0.2974
Group.MF.F.GCos	**0.0593**	**0.0923**	**0.2093**	**0.2999**
Group.MF.F.ICos	0.0569	0.0897	0.2062	0.2921

Result 5: the user-item preferences act more positive than the friendship in terms of enhancing group recommendation

Results – combined effect



(a) Last.fm (b) Douban

Result 6: combination of Group.MF.F.GCos and Group.MF.I.F@train.80 for fusing the two resources friendship and user-item preferences together achieves accuracy improvement

Conclusion

- Fused both **friendship** and **user-item preference** data to improve the accuracy of recommending interest groups
- Proved the outperforming suitability of **regularization model** for handling the one mode friendship data, and the **factorization model** for processing the user-item bipartite data

- <u>*Future work*</u>: more auxiliary resources, more algorithm comparisons