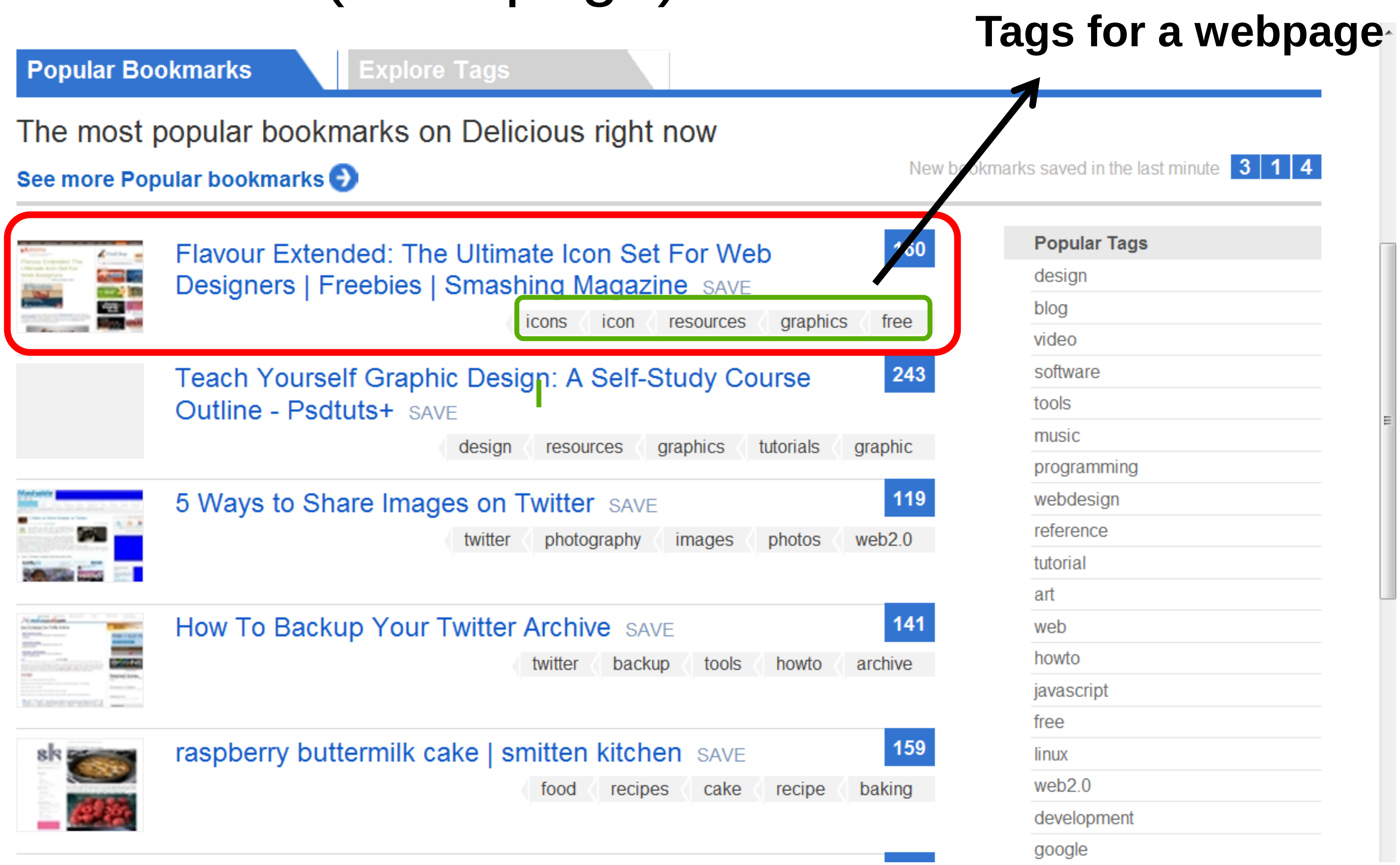
PERSONALIZED TAG RECOMMENDATION

Ziyu Guan†, Xiaofei He†, Jiajun Bu†,
Qiaozhu Mei‡, Chun Chen†, Can Wang†

† Zhejiang University, China
‡Univ. of Illinois/Univ. of Michigan

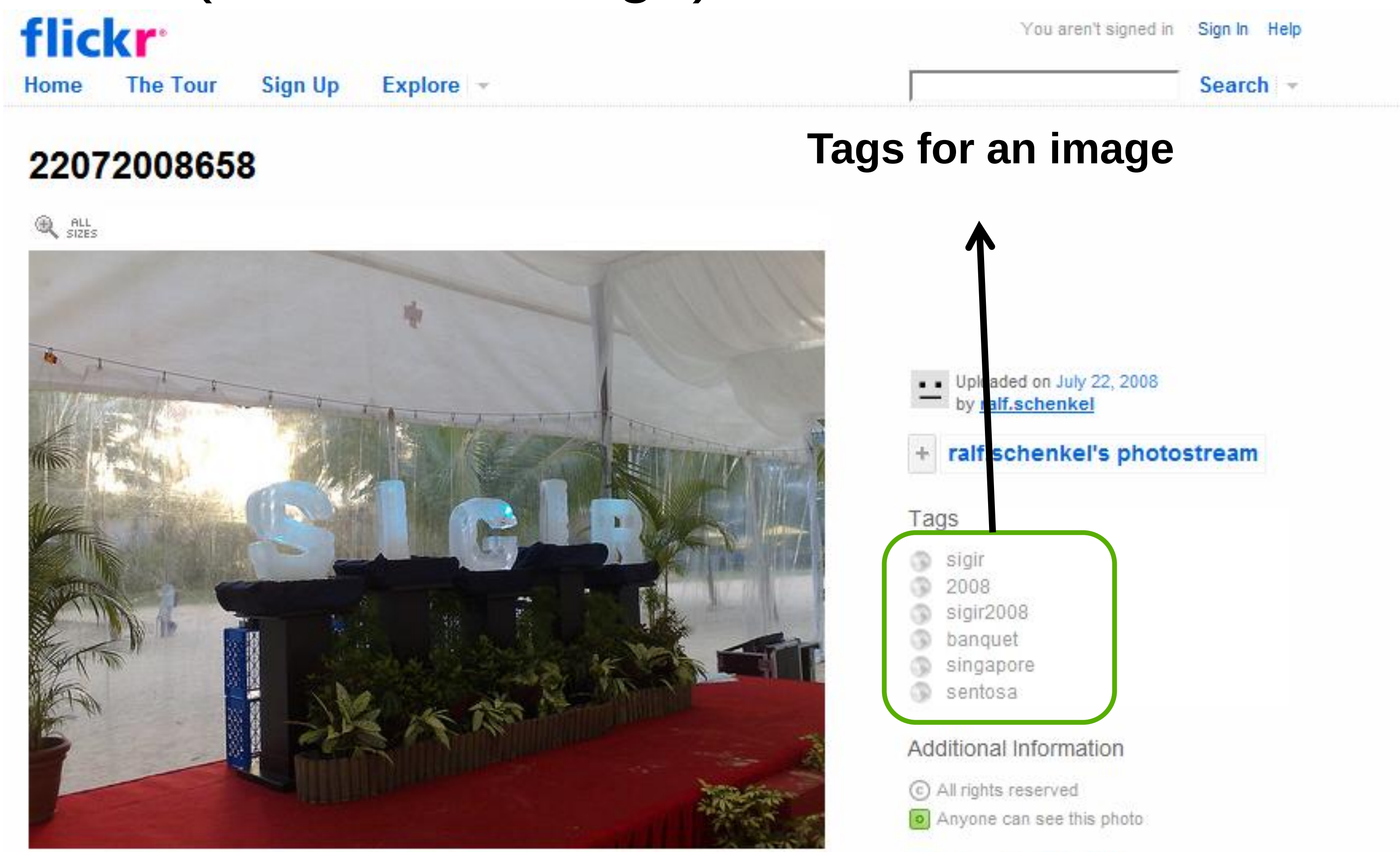
Booming of Social Tagging Applications

- Del.icio.us (Web page)



Booming of Social Tagging Applications

- Flickr (Photos & Image)



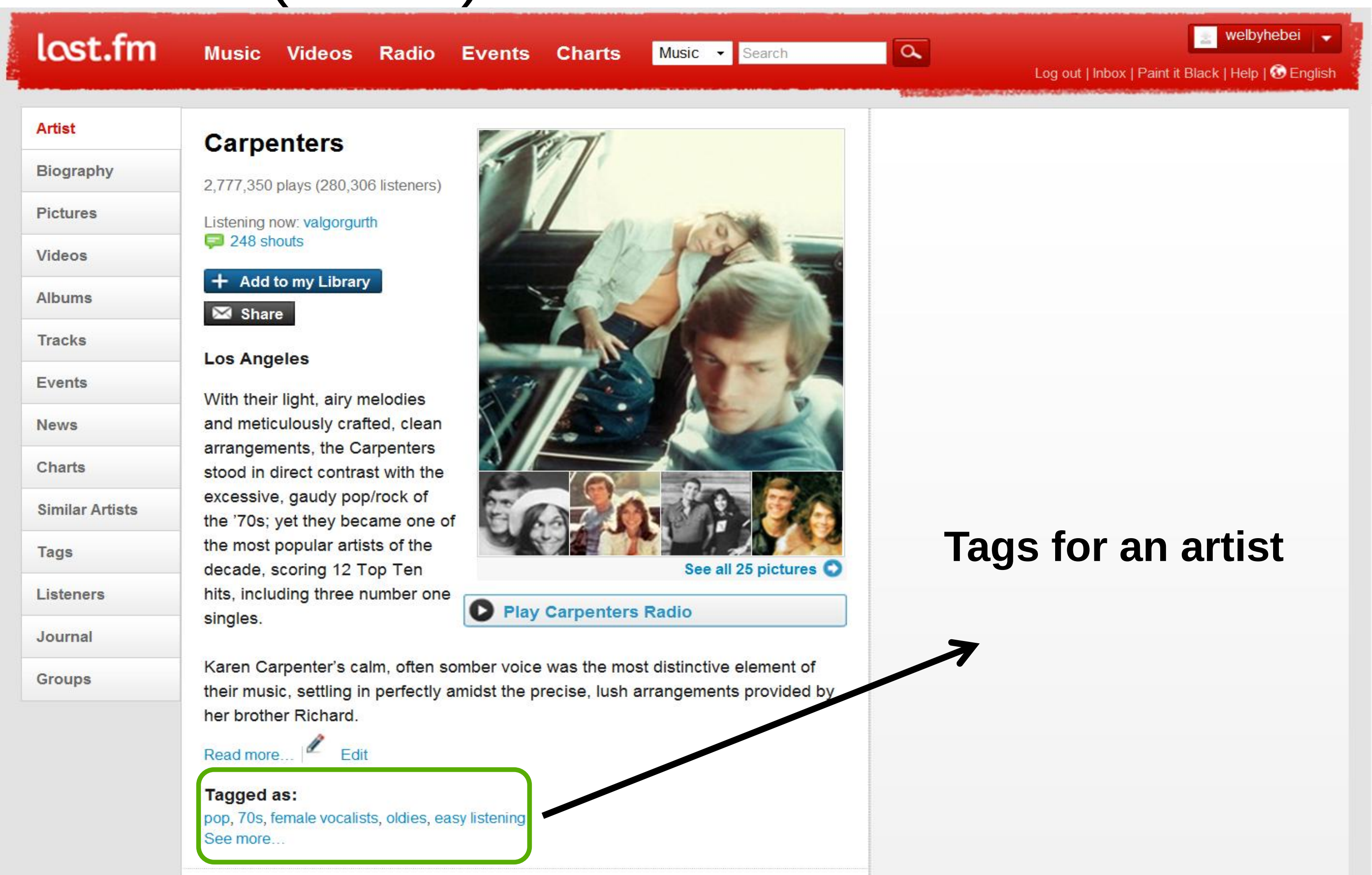
Booming of Social Tagging Applications

- CiteULike (Research publications)

Two tags for a paper

Booming of Social Tagging Applications

- Last.fm (Music)

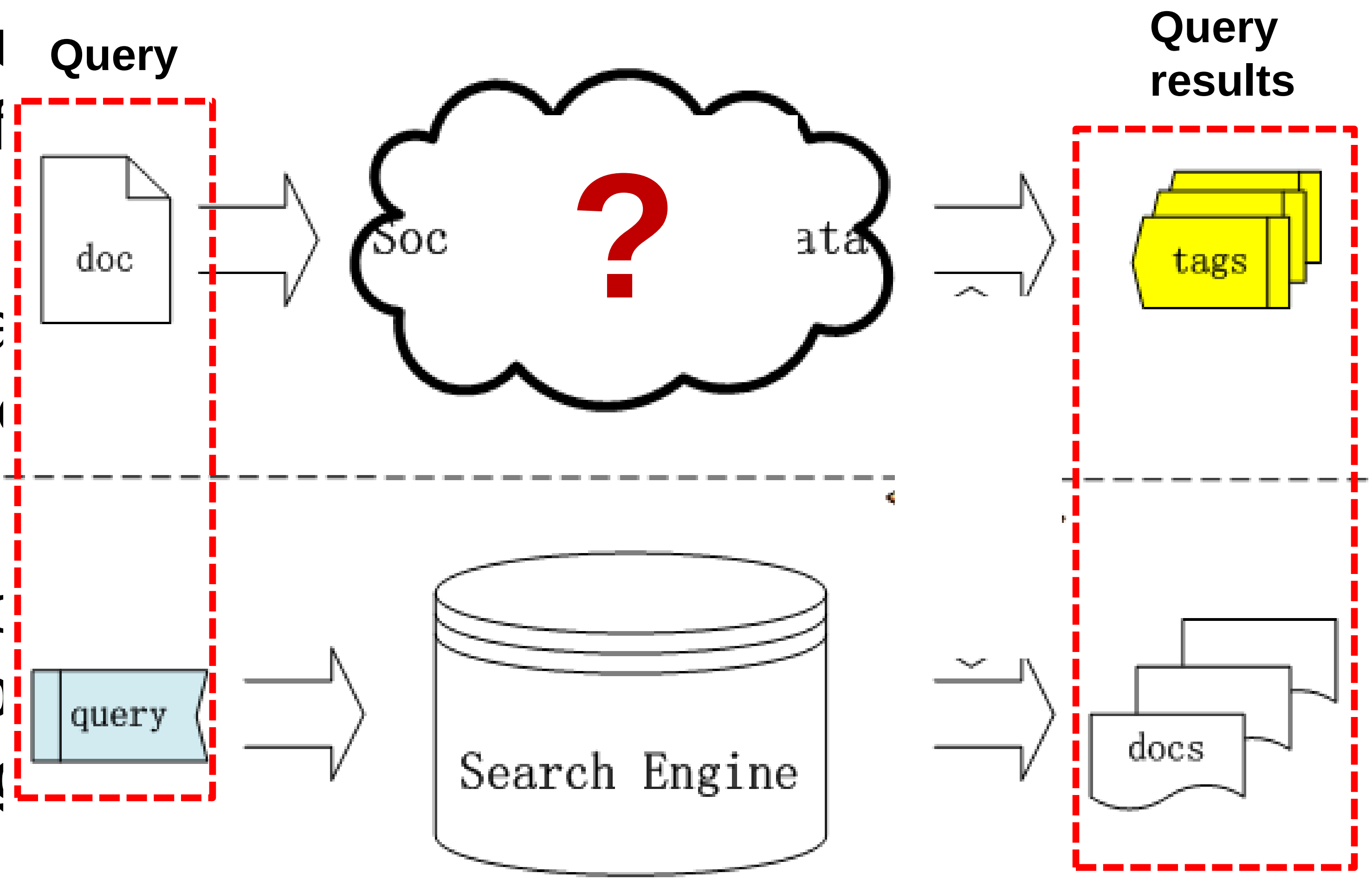


Tags for an artist

Large Scale Information Management

Compared to Web Search

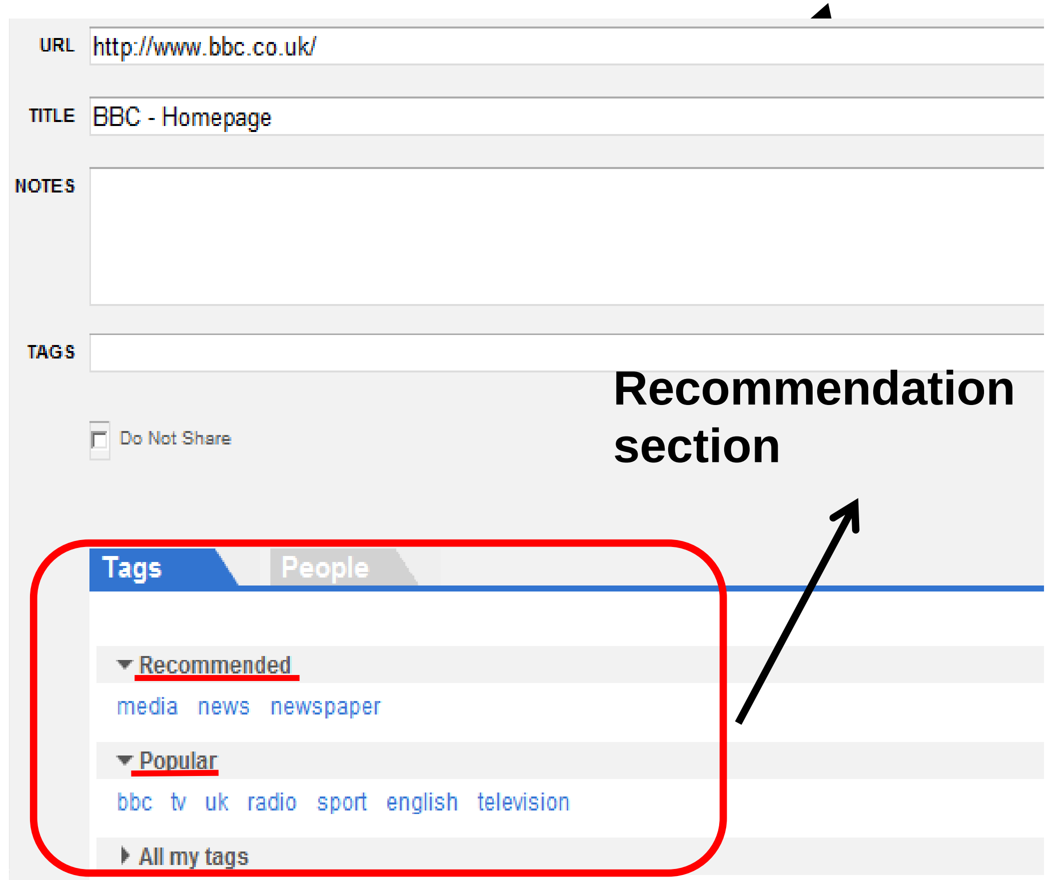
- Mu
  inf
- Fa
  “d
- No
  ob
  ta



Tag Recommendation in Real Applications

Artist to annotate

- Last.fm (Music)
  - Suggested tags: tags for the artist, selected by other users
  - Your tags: historical tags of the user
- Delicious (Web page)
  - "Popular" tags: tags selected by other users
  - "Recommended" tags: "Popular" U "Your tags"

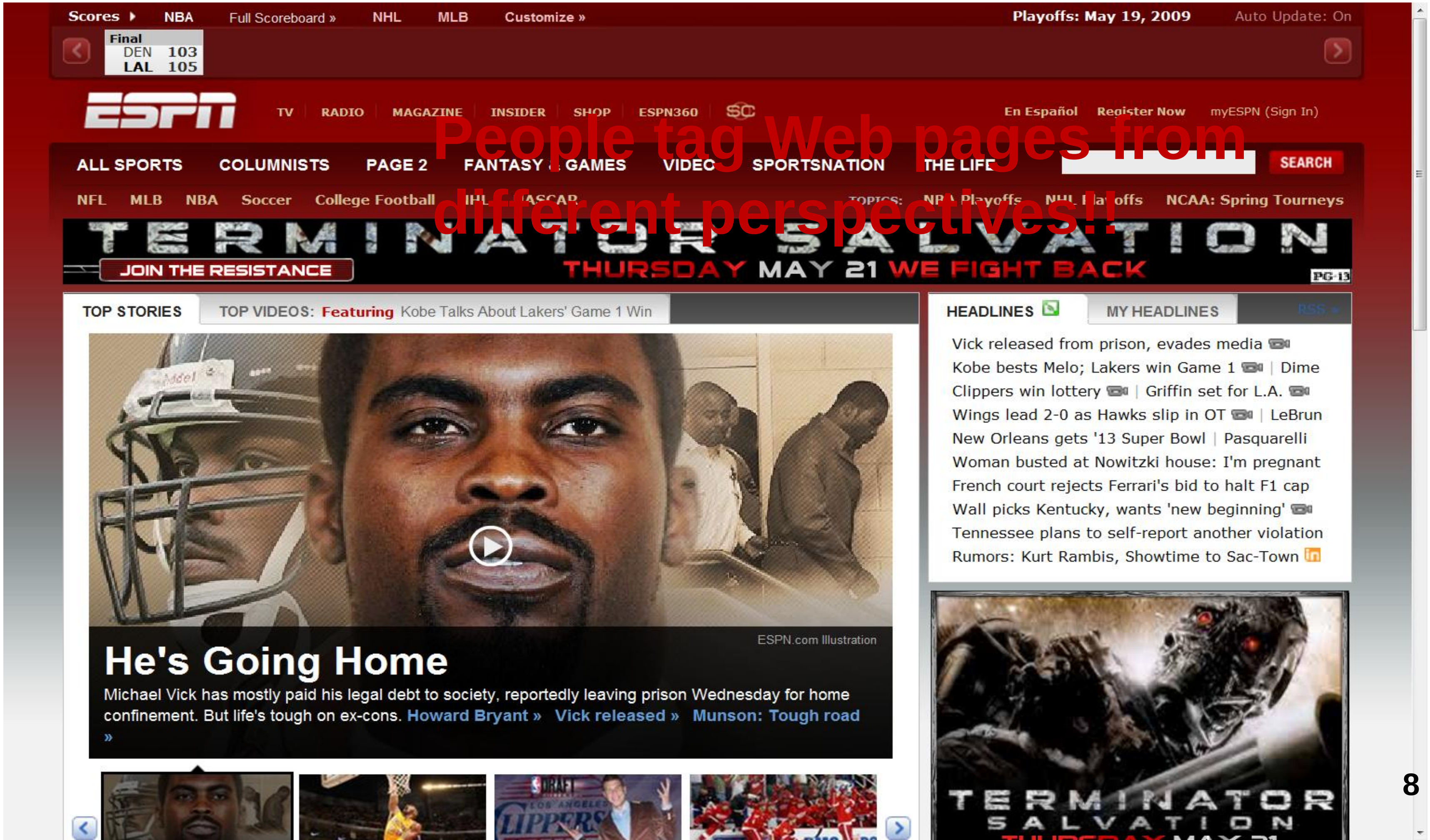


Recommendation section

Recommendation section

Personalization is Important

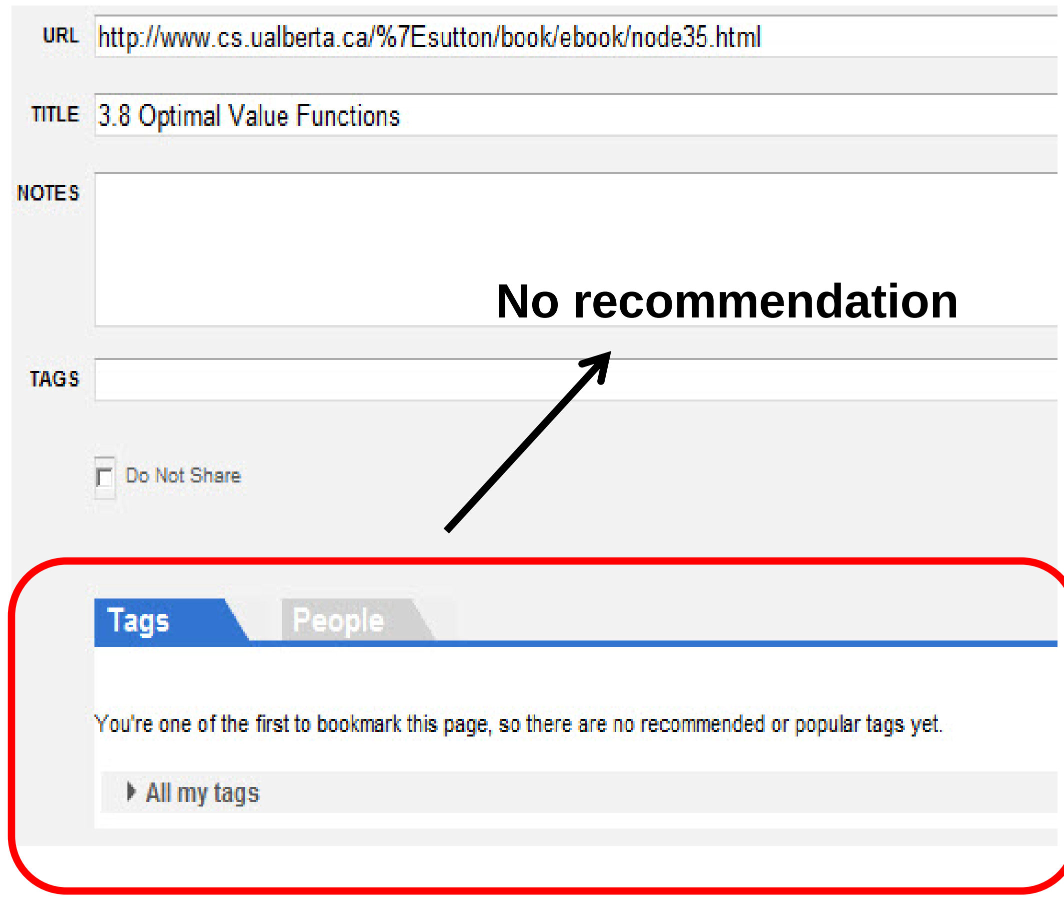
- Different users' bookmarks for the home page of ESPN: http://www.espn.com



People tag Web pages from different perspectives!!

Challenges

- Lots of web objects have few tags;
- Lots of users have few/no tags;
- Hard to combine “popular tags” and “your tags”;
- Collaborative, robust recommendation algorithms needed



Our Work v.s. Previous Work

- Most previous work focused on recommending tags for resources, ignoring the user factor.
  - Collaborative filtering based on documents (“users”) and tags (“movies”);
- Combine with user preference in an ad hoc way.

- We address personalized tag recommendation.
- An optimization framework with a unified objective function.

General idea of Our Approach

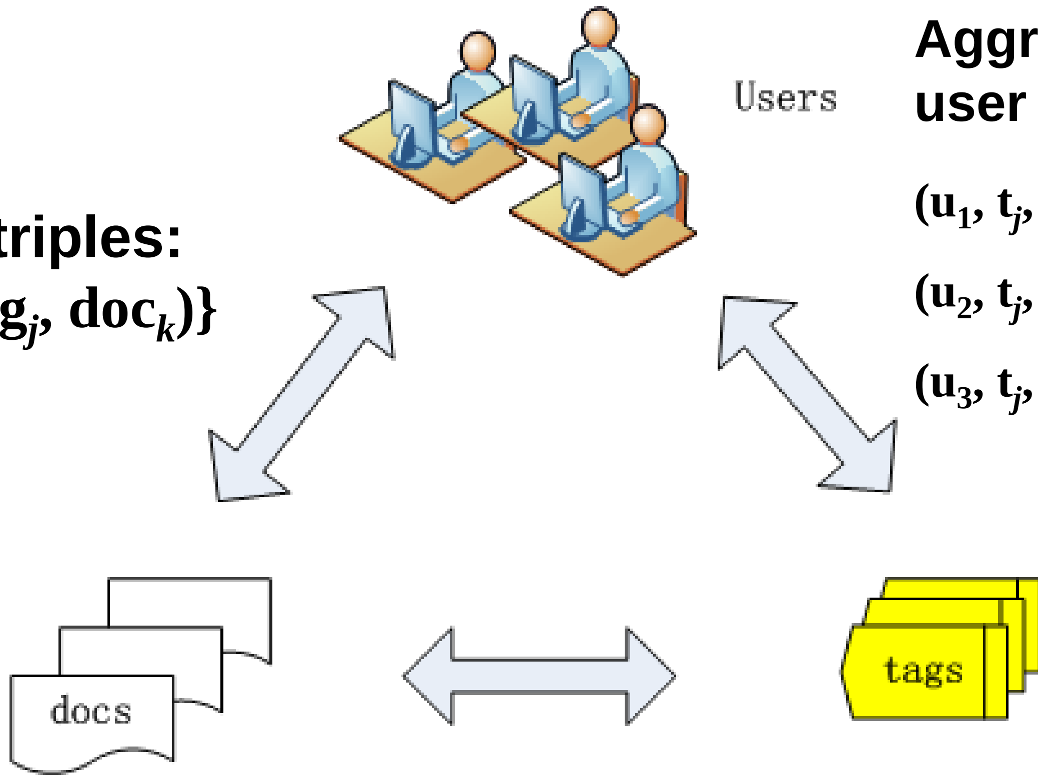
Graph-based Ranking of Multi-type interrelated Objects

Ranked list of tags

Representation of Tagging Data

A set of triples:

$\{(\text{user}_i, \text{tag}_j, \text{doc}_k)\}$



Aggregated across user dimension:

(u_1, t_j, d_k)

(u_2, t_j, d_k) (t_j, d_k) - 3

(u_3, t_j, d_k)

- Tagging data involves three interrelated types of objects: users, docs and tags

Highlights of Our Approach

- "Query" = document + user
- Exploit the affinity relation between documents; annotation relation between documents and tags; preference relation between users and tags.
- Model the problem as a graph-based ranking problem; developed a novel algorithm named Graph-based Ranking of Multi-type interrelated Objects (GRoMO)

Graph-based Regularization

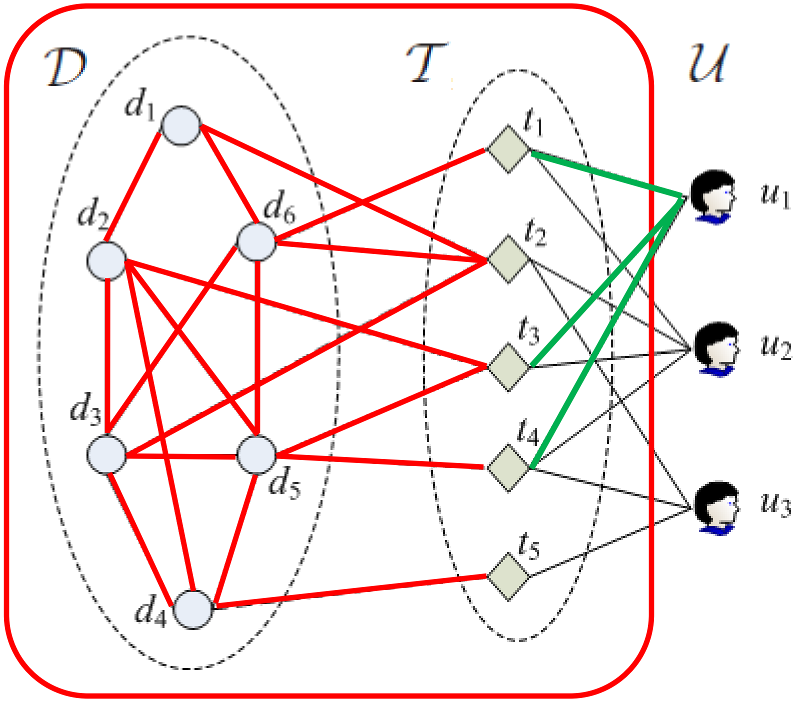
- Given data graph $G = (V, E, W)$, to learn a function $f : V \to \mathbb{R}$ from the data (e.g. for ranking or semi-supervised learning)
- A graph-based regularizer makes f smooth over the graph, i.e. similar data points should have similar function values (Zhou et al. NIPS04):

$$\frac{1}{2}\sum_{i,j} W_{ij}\left(\frac{1}{\sqrt{D_{ii}}}f_i - \frac{1}{\sqrt{D_{jj}}}f_j\right)^2 = \mathbf{f}^T\mathbf{L}\mathbf{f}$$

D: diagonal matrice, $D_{ii} = \sum_j W_{ij}$

Problem Formulation (Personalized Tag Recommendation)

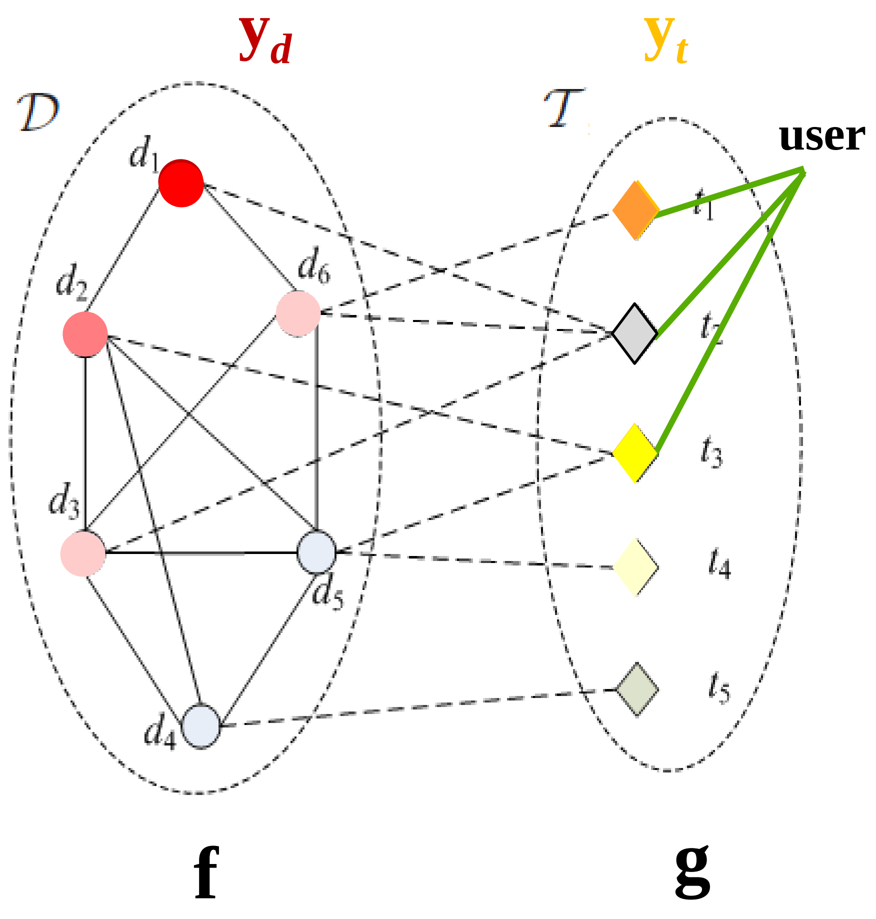
- Notations
  - $\mathcal{D}$ – the set of documents
  - $\mathcal{T}$ – the set of tags
  - $\mathcal{U}$ – the set of users
  - G_D – affinity graph of $\mathcal{D}$
  - $H_{D,\mathrm{T}}$ – bipartite graph describing annotation relationships between $\mathcal{D}$ and $\mathcal{T}$
  - $H_{\mathrm{U,T}}$ – bipartite graph describing users' historical usage of tags



- Exploit the affinity information relationships between documents and tags
- Input: query 'user' + document
  Output: ranking of tags
  - We use a user's tag usage history to represent a user (profile of tag preferences)

Problem Formulation (GRoMO)

- Notations
  - $\mathbf{W}$ – adjacency matrix of G_D
  - $\mathbf{R}$ – adjacency matrix of $H_{D,\mathrm{T}}$
  - $\mathbf{y}_d$, $\mathbf{y}_t$ – query vectors of documents and user-preferred tags
  - $\mathbf{f}$, $\mathbf{g}$ – ranking vectors of documents and tags
- Problem: given $\mathbf{W}$, $\mathbf{R}$, $\mathbf{y}_d$ and $\mathbf{y}_t$, to learn $\mathbf{f}$ and $\mathbf{g}$



Optimization Framework of GRoMO

$$Q(\mathbf{f},\mathbf{g}) = \frac{1}{2}\mu \sum_{i,j=1}^{|\mathcal{D}|} W_{ij}\left(\frac{1}{\sqrt{D_{ii}}}f_i - \frac{1}{\sqrt{D_{jj}}}f_j\right)^2$$

$$+\eta \sum_{i=1}^{|\mathcal{D}|}\sum_{j=1}^{|\mathcal{T}|} R_{ij}\left(\frac{1}{\sqrt{D^d_{ii}}}f_i - \frac{1}{\sqrt{D^t_{jj}}}g_j\right)^2$$

$$+\alpha \sum_{i=1}^{|\mathcal{D}|}(f_i - y_{di})^2 + \beta \sum_{i=1}^{|\mathcal{T}|}(g_i - y_{ti})^2, \qquad (1)$$

Similar documents have similar scores

If a tag is often used for a document, they have similar scores

Keep fidelity to the targeted document (s)

Keep fidelity to user-preferred tags

$$\langle \mathbf{f},\mathbf{g}\rangle = \arg\min_{\mathbf{f},\mathbf{g}} Q(\mathbf{f},\mathbf{g}).$$

Matrix-vector Form

- Define

$$\mathbf{S}_W = \mathbf{D}^{(-1/2)}\mathbf{W}\mathbf{D}^{(-1/2)}, \qquad \mathbf{S}_R = \mathbf{D}_d^{(-1/2)}\mathbf{R}\mathbf{D}_t^{(-1/2)}.$$

- The cost function can be written as

$$Q(\mathbf{f},\mathbf{g}) = \mu\mathbf{f}^T(\mathbf{I}-\mathbf{S}_W)\mathbf{f} + \eta(\mathbf{f}^T\mathbf{f} + \mathbf{g}^T\mathbf{g} - 2\mathbf{f}^T\mathbf{S}_R\mathbf{g})$$
$$+\alpha(\mathbf{f}-\mathbf{y}_d)^T(\mathbf{f}-\mathbf{y}_d) + \beta(\mathbf{g}-\mathbf{y}_t)^T(\mathbf{g}-\mathbf{y}_t).$$

- Closed-form solution:

$$\mathbf{f}^* = \left[(1-\beta)\mathbf{I} - \mu\mathbf{S}_W - \frac{\eta^2}{\beta+\eta}\mathbf{S}_R\mathbf{S}_R^T\right]^{-1} \times \left(\alpha\mathbf{y}_d + \frac{\beta\eta}{\beta+\eta}\mathbf{S}_R\mathbf{y}_t\right)$$

$$\mathbf{g}^* = \frac{\eta}{\beta+\eta}\mathbf{S}_R^T\mathbf{f}^* + \frac{\beta}{\beta+\eta}\mathbf{y}_t.$$

Iterative Solution of GRoMO

- Set $\mathbf{f}(0) = \mathbf{y}_d$, $\mathbf{g}(0) = \mathbf{y}_t$. In the t-th iteration, first use $\mathbf{f}(t)$ to compute $\mathbf{g}(t+1)$:

$$\mathbf{g}(t+1) = \frac{\eta}{\beta+\eta}\mathbf{S}_R^T\mathbf{f}(t) + \frac{\beta}{\beta+\eta}\mathbf{y}_t,$$

- Then, use $\mathbf{g}(t+1)$ and $\mathbf{f}(t)$ to compute $\mathbf{f}(t+1)$:

$$\mathbf{f}(t+1) = \frac{\mu}{1-\beta}\mathbf{S}_W\mathbf{f}(t) + \frac{\eta}{1-\beta}\mathbf{S}_R\mathbf{g}(t+1) + \frac{\alpha}{1-\beta}\mathbf{y}_d.$$

- Another Iterative form involving $\mathbf{f}$ only:

$$\begin{aligned}\mathbf{f}(t+1) &= \frac{1}{1-\beta}\left(\mu\mathbf{S}_W + \frac{\eta^2}{\beta+\eta}\mathbf{S}_R\mathbf{S}_R^T\right)\mathbf{f}(t) \\ &\quad + \frac{\alpha}{1-\beta}\mathbf{y}_d + \frac{\beta\eta}{(1-\beta)(\beta+\eta)}\mathbf{S}_R\mathbf{y}_t.\end{aligned}$$

Graph Construction

- For **W** we use cosine similarities between documents as edge weights.
- We set W_{ij} (document affinity) as

$$W_{ij} = \begin{cases} \text{cosine}(i,j) & \text{if i} \in \text{KNN}(j) \text{ or } j \in \text{KNN}(i) \\ 0 & otherwise \end{cases}$$

- We set R_{ij} (B is the observed set of tagging data)

$$R_{ij} = \mid \{u_k \mid u_k \in \mathcal{U} \text{ and } (u_k, d_i, t_j) \in \mathcal{B}\} \mid$$

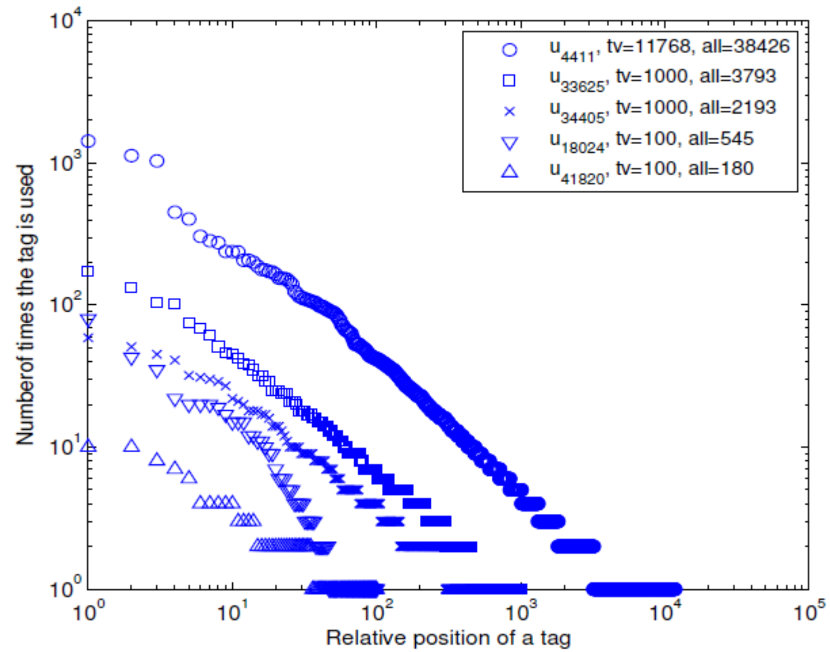
Setting Query Vectors

- Query vector $\mathbf{y}_d$ is set as follows

$$y_{di} = \begin{cases} 1 & d_i = d \\ 0 & \text{otherwise} \end{cases} .$$

- Tag frequency of a user tends to follow power law, hence $\mathbf{y}_t$ is set as

$$y_{ti} = \begin{cases} \frac{\lceil \log(frequency_{u,t_i})+1 \rceil}{\sum_{t_j \in \mathcal{T}_u} \lceil \log(frequency_{u,t_j})+1 \rceil} & t_i \in \mathcal{T}_u \\ 0 & \text{otherwise} \end{cases} ,$$



Frequency of tag v.s. relative position

Experiments

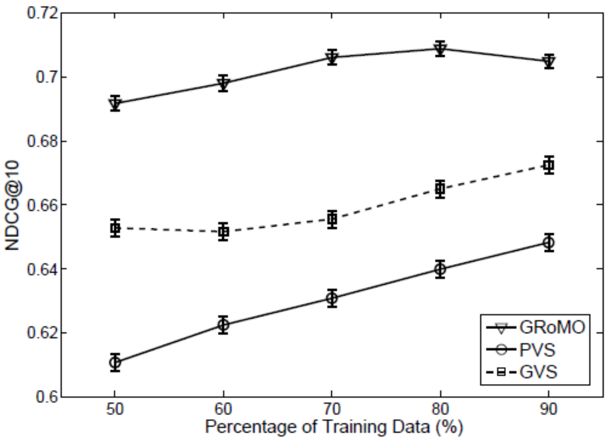
- Dataset
  - Our dataset contains 167,885 bookmarks.

  - Statistics: 300 users, 11,795 Web pages, 17,777 tags
- We use 10% bookmarks as test data
  - Web page + user as “queries”; tags as gold standard.
- Evaluate with NDCG, average precision, average recall.

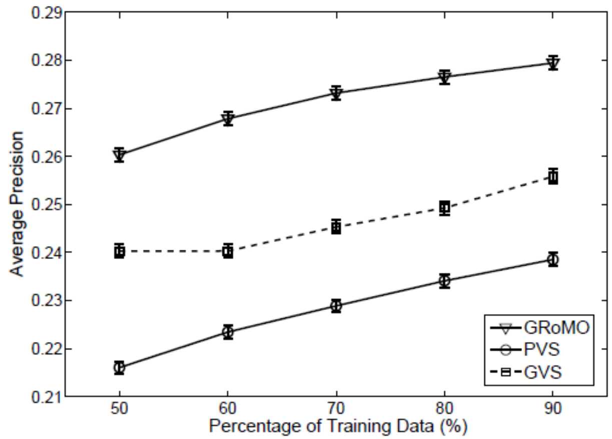
Baseline

- Global Vector Similarity (GVS): independent to the user, only dependent on the docs & tags.
  - Item-based collaborative filtering using documents and tags;
- Personal Vector Similarity (PVS): recommend the tags used by the user.
  - Using documents (and tags) tagged by the particular user

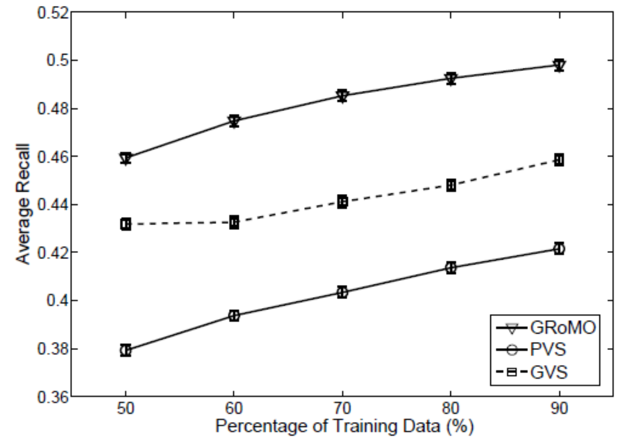
Experimental Results – Performance Comparison



(a) **NDCG@10**



(b) **Avg. Precision**



(c) **Avg. Recall**

GRoMO > GVS > PVS

Experimental Results – Performance Comparison

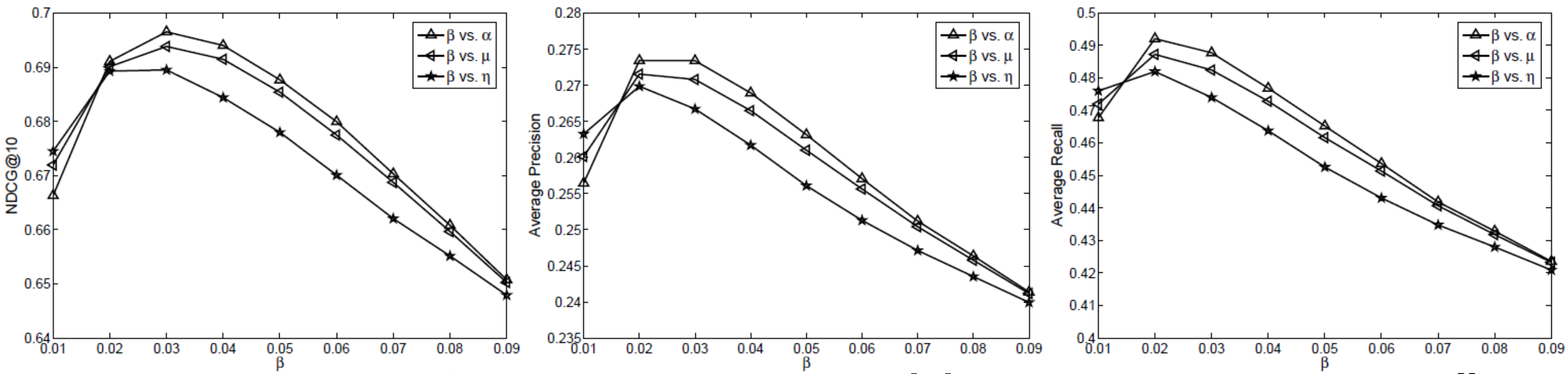
	NDCG@1			NDCG@3			NDCG@5		
Training Data (%)	GRoMO	PVS	GVS	GRoMO	PVS	GVS	GRoMO	PVS	GVS
50	**0.5422****	0.4518	0.5136	**0.5357****	0.4617	0.5209	**0.5863****	0.5127	0.5593
60	**0.5434****	0.4645	0.5202	**0.5381****	0.4726	0.5122	**0.5897****	0.5234	0.5542
70	**0.5558****	0.4759	0.5301	**0.5474****	0.4825	0.5179	**0.5982****	0.5314	0.5593
80	**0.5534**	0.4873	0.5478	**0.5483***	0.4907	0.5322	**0.5990****	0.5398	0.5706

NDCG@1, NDCG@3, and NDCG@5.

GRoMO > GVS > PVS

GRoMO works especially better when smaller training data is observed

Experimental Results – Parameter Setting



(a) **NDCG@10** (b) **Precision** (c) **Recall**

β versus each of the other parameters α, μ, η (fix the other two)

Observation: β need to be kept small
Optimal: μ = 0.3; η = 0.17; α = 0.5; β = 0.03

Experimental Results
Tag Recommendation Example

URL: http://www.brand-name-coupons.com/how-to-search-amazon-for-deals/		
UserID	Ground Truth	GRoMO Recommended
8414	amazon, bargains, Coupons, deals, discount, howto, shopping	**amazon**, **deals**, **bargains**, **shopping**, discounts, coupons, bargain, s3, search, **discount**
37982	amazon, cheap, coupons, sales, shopping	**amazon**, deals, **shopping**, bargains, **coupons**, discounts, s3, search, **cheap**, discount
5472	blog, howto, shopping, tips, tools	**shopping**, **tools**, free, web, design, reference, software, **howto**, **tips**, amazon

- **Three Users' annotations in the last 10% testing data for the URL "http://www.brand-name-coupons.com/how-to-searchamazon-for-deals.html".**
- **Tags with bold font indicate matches with the tags actually used by the user.**

Summary

- Personalized tag recommendation
- Graph-based ranking of multi-type interrelated objects
  - Doc-doc; doc-tag; and user-tag relations
- A solution by optimizing a unified objective function
- Future work
  - Explore doc-user, user-user relations
  - Parameter tuning
  - Efficient (e.g., distributed) solution for large scale data;

Thank You !

Derivation of Optimal Solution

- Differentiate Q with respect to **f** and **g**, we obtain

$$\frac{\partial Q}{\partial \mathbf{f}} = \left[(1-\beta)\mathbf{I} - \mu \mathbf{S}_W\right]\mathbf{f} - \eta \mathbf{S}_R \mathbf{g} - \alpha \mathbf{y}_d = 0.$$

$$\frac{\partial Q}{\partial \mathbf{g}} = (\beta + \eta)\mathbf{g} - \eta \mathbf{S}_R^T \mathbf{f} - \beta \mathbf{y}_t = 0.$$

$$\mathbf{f}^* = \left[(1-\beta)\mathbf{I} - \mu \mathbf{S}_W - \frac{\eta^2}{\beta+\eta}\mathbf{S}_R \mathbf{S}_R^T\right]^{-1} \times \left(\alpha \mathbf{y}_d + \frac{\beta\eta}{\beta+\eta}\mathbf{S}_R \mathbf{y}_t\right)$$

$$\mathbf{g}^* = \frac{\eta}{\beta+\eta}\mathbf{S}_R^T \mathbf{f}^* + \frac{\beta}{\beta+\eta}\mathbf{y}_t.$$