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)

Tags for a webpage:
Booming of Social Tagging Applications

- Flickr (Photos & Image)

Tags for an image:
- sigir
- 2008
- sigir2008
- banquet
- singapore
- sentosa
Booming of Social Tagging Applications

• CiteULike (Research publications)
Booming of Social Tagging Applications

- Last.fm (Music)

Tags for an artist
Compared to Web Search

- Much more informative "queries"
- Far less informative "documents"
- No search engine to obtain initial relevant tag set.

Diagram:
- Query
- Search Engine
- Documents
- Tags
- Data
Tag Recommendation in Real Applications

- **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”
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)\}\n
• Tagging data involves three interrelated types of objects: users, docs and tags

Aggregated across user dimension:

- \( (u_1, t_j, d_k) \)
- \( (u_2, t_j, d_k) \)
- \( (u_3, t_j, d_k) \)
- \( (t_j, d_k) - 3 \)
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 \( f : V \rightarrow \mathbb{R} \) 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):

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

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 $\mathcal{D}$
  - $H_{D,T}$ – bipartite graph describing annotation relationships between $\mathcal{D}$ and $\mathcal{T}$
  - $H_{U,T}$ – bipartite graph describing users’ historical usage of tags

- **Exploit**
  - the affinity relationships between documents.
  - the annotation relationships between documents and tags.

We use a user's tag usage history to represent a user (profile of tag preferences).
Problem Formulation (GRoMO)

• Notations
  – $W$ – adjacency matrix of $G_D$
  – $R$ – adjacency matrix of $H_{D,T}$
  – $y_d, y_t$ – query vectors of documents and user-preferred tags
  – $f, g$ – ranking vectors of documents and tags

• Problem: given $W$, $R$, $y_d$ and $y_t$, to learn $f$ and $g$
Optimization Framework of GRoMO

\[ Q(f, g) = \frac{1}{2\mu} \sum_{i,j=1}^{\mathcal{D}} W_{ij} \left( \frac{1}{\sqrt{D_{ii}}} f_i - \frac{1}{\sqrt{D_{jj}}} f_j \right)^2 + \eta \sum_{i=1}^{\mathcal{D}} \sum_{j=1}^{\mathcal{T}} R_{ij} \left( \frac{1}{\sqrt{D_{ii}^d}} f_i - \frac{1}{\sqrt{D_{jj}^t}} g_j \right)^2 + \alpha \sum_{i=1}^{\mathcal{D}} (f_i - y_{di})^2 + \beta \sum_{i=1}^{\mathcal{T}} (g_i - y_{ti})^2, \quad (1) \]

\[ \langle f, g \rangle = \arg \min_{f, g} Q(f, g). \]
Matrix-vector Form

- Define

\[ S_W = D^{(-1/2)} WD^{(-1/2)}, \quad S_R = D_d^{(-1/2)} R D_t^{(-1/2)}. \]

- The cost function can be written as

\[
Q(f, g) = \mu f^T (I - S_W)f + \eta (f^T f + g^T g - 2f^T S_R g) \\
+ \alpha (f - y_d)^T (f - y_d) + \beta (g - y_t)^T (g - y_t).
\]

- Closed-form solution:

\[
f^* = \left[(1 - \beta)I - \mu S_W - \frac{\eta^2}{\beta + \eta} S_R S_R^T\right]^{-1} \times \left(\alpha y_d + \frac{\beta \eta}{\beta + \eta} S_R y_t\right)
\]

\[
g^* = \frac{\eta}{\beta + \eta} S_R^T f^* + \frac{\beta}{\beta + \eta} y_t.
\]
Iterative Solution of GRoMO

• Set $f(0) = y_d$, $g(0) = y_t$. In the $t$-th iteration, first use $f(t)$ to compute $g(t+1)$:

$$g(t + 1) = \frac{\eta}{\beta + \eta} S_R^T f(t) + \frac{\beta}{\beta + \eta} y_t,$$

• Then, use $g(t+1)$ and $f(t)$ to compute $f(t+1)$:

$$f(t + 1) = \frac{\mu}{1 - \beta} S_W f(t) + \frac{\eta}{1 - \beta} S_R g(t + 1) + \frac{\alpha}{1 - \beta} y_d.$$

• Another Iterative form involving $f$ only:

$$f(t + 1) = \frac{1}{1 - \beta} \left( \mu S_W + \frac{\eta^2}{\beta + \eta} S_R S_R^T \right) f(t) + \frac{\alpha}{1 - \beta} y_d + \frac{\beta \eta}{(1 - \beta)(\beta + \eta)} S_R y_t.$$
Graph Construction

• For $W$ we use cosine similarities between documents as edge weights.

• We set $W_{ij}$ (document affinity) as
  \[ W_{ij} = \begin{cases} 
  \text{cosine}(i, j) & \text{if } i \in \text{KNN}(j) \text{ or } j \in \text{KNN}(i) \\
  0 & \text{otherwise} 
  \end{cases} \]

• We set $R_{ij}$ ($B$ is the observed set of tagging data)
  \[ R_{ij} = | \{ u_k \mid u_k \in \mathcal{U} \text{ and } (u_k, d_i, t_j) \in B \} | \]
Setting Query Vectors

- Query vector $y_d$ 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 $y_t$ is set as

$$y_{ti} = \begin{cases} 
\frac{[\log(\text{frequency}_{u,t_i})+1]}{\sum_{t_j \in T_u} [\log(\text{frequency}_{u,t_j})+1]} & t_i \in T_u \\
0 & \text{otherwise} 
\end{cases}$$
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

<table>
<thead>
<tr>
<th>Training Data (%)</th>
<th>NDCG@1</th>
<th>NDCG@3</th>
<th>NDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GRoMO</td>
<td>PVS</td>
<td>GVS</td>
</tr>
<tr>
<td>50</td>
<td>0.5422***</td>
<td>0.4518</td>
<td>0.5136</td>
</tr>
<tr>
<td>60</td>
<td>0.5434***</td>
<td>0.4645</td>
<td>0.5202</td>
</tr>
<tr>
<td>70</td>
<td>0.5558***</td>
<td>0.4759</td>
<td>0.5301</td>
</tr>
<tr>
<td>80</td>
<td>0.5534</td>
<td>0.4873</td>
<td>0.5473</td>
</tr>
</tbody>
</table>

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

GRoMO > GVS > PVS

GRoMO works especially better when smaller training data is observed
Experimental Results – Parameter Setting

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

<table>
<thead>
<tr>
<th>UserID</th>
<th>Ground Truth</th>
<th>GReMO Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>8414</td>
<td>amazon, bargains, Coupons, deals, discount, howto, shopping</td>
<td>amazon, deals, bargains, shopping, discounts, coupons, bargain, s3, search, discount</td>
</tr>
<tr>
<td>37982</td>
<td>amazon, cheap, coupons, sales, shopping</td>
<td>amazon, deals, shopping, bargains, coupons, discounts, s3, search, cheap, discount</td>
</tr>
<tr>
<td>5472</td>
<td>blog, howto, shopping, tips, tools</td>
<td>shopping, tools, free, web, design, reference, software, howto, tips, amazon</td>
</tr>
</tbody>
</table>

- Three Users’ annotations in the last 10% testing data for the URL “http://www.brand-name-coupons.com/how-to-search-amazon-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

$$\frac{\partial Q}{\partial f} = [(1 - \beta)I - \mu S_W]f - \eta S_R g - \alpha y_d = 0.$$  

$$\frac{\partial Q}{\partial g} = (\beta + \eta)g - \eta S_R^T f - \beta y_t = 0.$$  

$$f^* = \left[(1 - \beta)I - \mu S_W - \frac{\eta^2}{\beta + \eta} S_R S_R^T \right]^{-1} \times \left(\alpha y_d + \frac{\beta \eta}{\beta + \eta} S_R y_t \right)$$

$$g^* = \frac{\eta}{\beta + \eta} S_R^T f^* + \frac{\beta}{\beta + \eta} y_t.$$