Thematically Focused Search in Web 2.0 Folksonomies

Sergej Sizov, Olaf Görlitz
University of Koblenz-Landau
Germany
can this huge mass of users give us new insights that would not be possible by considering individual contributions ?!
The mass makes the difference?

Explore / Tags / www / clusters

web, internet, website, design, site, blog, webdesign, google, flickr, screenshot

See more in this cluster...

canada, banff, alberta, rockies, mountains, hotel, mountain, bowmore, lakehouse

See more in this cluster...

related tags: web, webdesign, css, reference, design, internet, software, programming, html, tools
Flickr example – what you get..
.. and what you want

Query: "Athos fire"

Flickr: queries with low recall

Searching for "bamberg aula" did not yield any results. Would you like to search for alternatives like "germany, deutschland, bavaria, bayern, or franken" instead?
Flickr: queries with low recall (2)
Outline

Problem formalization
Thematically focused search and ranking
Distributed setting – pro & contra
Evaluation
Formalizing the problem

Collaborative content sharing framework:

\[
\begin{align*}
\text{users } u & \in U & \text{tags } t & \in T & \text{resources } r & \in R \\
\end{align*}
\]

Information cloud:

\[
T := (Y^*, f), \quad f(t) : Y^* \subseteq Y, Y^* \rightarrow [0..1] 
\]

- user-centric:
  \[
  T_u := (Y_u, f), \quad Y_u \subseteq u \times T \times R, 
  \]

- resource-centric:
  \[
  T_r := (Y_r, f), \quad Y_u \subseteq U \times T \times r, 
  \]

- community-specific:
  \[
  T_{U^*} := (Y_{U^*}, f), \quad Y_{U^*} \subseteq U^* \times T \times R 
  \]

- collection-specific
  \[
  T_{R^*} := (Y_{R^*}, f), \quad Y_{R^*} \subseteq U \times T \times R^* 
  \]

- arbitrary
  \[
  T_{U^*R^*} := (Y_{U^*R^*}, f), \quad Y_{U^*R^*} \subseteq U^* \times T \times R^* 
  \]

.. e.g. obtained by traversing the hypergraph up to certain depth

Common recommender scenarios:

- Given a user, recommend photos which may be of interest.
- Given a user, recommend users they may like to contact.
- Given a user, recommend groups they may like to join.
The IR background – constructing feature vectors

**tag features:**

\[
\begin{align*}
\text{iif} \cdot \text{if}(p2p) &= (\log \frac{5}{2}, \log \frac{5}{5}) \cdot (6, 6)^T \\
\text{iif} \cdot \text{if}(talk) &= (\log \frac{5}{1}, \log \frac{5}{1}) \cdot (1, 1)^T \\
\text{iif} \cdot \text{if}(slide) &= (\log \frac{5}{1}, \log \frac{5}{1}) \cdot (1, 1)^T \\
\text{iif} \cdot \text{if}(pdf) &= (\log \frac{5}{1}, \log \frac{5}{1}) \cdot (1, 1)^T \\
\text{iif} \cdot \text{if}(routing) &= (\log \frac{5}{1}, \log \frac{5}{1}) \cdot (6, 6)^T
\end{align*}
\]

Further dimensions of interest:

- favorites
- group membership
- contact lists
- comments on other's resources

\[i_f(i) = (a_i, b_i)\]
\[\text{iif}(i) = \left(\log \frac{|J|}{|J^*|}, \log \frac{|K|}{|K^*|}\right),\]
\[j \in J^*/k \in K^*: (i, j, k) \in Y^*\]
\[\text{weight}(i) = \text{iif}(i) \cdot i_f(i)^T\]

.. defined analogously to tf·idf
Generalization: the tensor model

idea: using multi-dimensional arrays for representing relationships

matricize:

\[ X_{users} = \text{community cloud} \]
\[ X_{resources} = \text{cloud for resources} \]
\[ X_{tags} = \text{cloud for tags} \]

slide shows 3rd order tensor for common Web 2.0 dimensions, but can (and should) be extended by other relationships (favorites, comments, groups..)
Tensors: mode-n matrix multiplication, Tucker decomposition...

\[ A \times X \]

\[ Y = X \times_1 A \]
\[ Y_{(::k)} = X_{(::k)} \times A^T \]

\[ X \times X \]

\[ Y = X \times_3 B \]
\[ Y_{(i::)} = X_{(i::)} \times C^T \]

\[ X \times B \]

\[ Y = X \times_2 B \]
\[ Y_{(j::)} = X_{(j::)} \times B^T \]

**general idea:** decompose tensor in order to identify significant "factors" along each dimension (multi-dimensional methods analogously to LSI, PCA)

**our current approach:** input-tensor R is decomposed into \( R=UxDxV \), \( V \) contains the orthogonal mapping of \( R \) into space of real tensors with target dimension (i.e. \( V \) is also a matrix). Restrict the resulting feature vectors to 5-10 most significant dimensions, analogously to LSI.
User-centered focusing

1. Compute characteristic feature vectors for resources, tags, contacts, or favorites of the given user
2. Construct appropriate decision model (centroid, naive bayes, SVM, etc.)
3. Explore the tagging cloud around the user, order matches according wrt estimated utility function (cosine similarity, classification confidence, etc.)
4. Return the top-k result set (e.g. top-10, top-20) to the user
Datasets

- Flickr dataset (2004-2005)
  - 319,686 users,
  - 1,607,879 tags,
  - 28,153,045 resources,
  - 112,900,000 tag assignm.

- Del.icio.us dataset (2003-2006)
  - 532,924 users,
  - 2,481,698 tags,
  - 17,262,480 resources,
  - 140,126,586 tag assignm.

Evaluation: apriori method
- remove a certain fraction of relationships (e.g. group participation, comments, ..) from the test cloud
- test the ability of the recommender to reconstruct missing relationships (i.e. to place them within top-k of the result set)
## Results: user-focused recommendations

### recommending favorites

<table>
<thead>
<tr>
<th>User representation</th>
<th>Training:10 prec@10</th>
<th>Training:10 prec@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.167</td>
<td>0.167</td>
</tr>
<tr>
<td>User items</td>
<td>0.259</td>
<td>0.268</td>
</tr>
<tr>
<td>Commented items</td>
<td>0.236</td>
<td>0.221</td>
</tr>
<tr>
<td>Favorites</td>
<td>0.872</td>
<td>0.727</td>
</tr>
<tr>
<td>Combined</td>
<td>0.854</td>
<td>0.713</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User representation</th>
<th>Training:20 prec@10</th>
<th>Training:20 prec@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.167</td>
<td>0.167</td>
</tr>
<tr>
<td>Commented items</td>
<td>0.255</td>
<td>0.248</td>
</tr>
<tr>
<td>Favorites</td>
<td>0.918</td>
<td>0.851</td>
</tr>
<tr>
<td>Combined</td>
<td>0.899</td>
<td>0.828</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User representation</th>
<th>Training:40 prec@10</th>
<th>Training:40 prec@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.167</td>
<td>0.167</td>
</tr>
<tr>
<td>Commented items</td>
<td>0.265</td>
<td>0.266</td>
</tr>
<tr>
<td>Favorites</td>
<td>0.933</td>
<td>0.903</td>
</tr>
<tr>
<td>Combined</td>
<td>0.914</td>
<td>0.876</td>
</tr>
</tbody>
</table>

### recommending contacts

<table>
<thead>
<tr>
<th>User representation</th>
<th>prec@5</th>
<th>prec@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.167</td>
<td>0.167</td>
</tr>
<tr>
<td>User Items</td>
<td>0.574</td>
<td>0.472</td>
</tr>
<tr>
<td>Commented Items</td>
<td>0.576</td>
<td>0.473</td>
</tr>
<tr>
<td>Favorites</td>
<td>0.535</td>
<td>0.455</td>
</tr>
<tr>
<td>Contacts (training 10)</td>
<td>0.604</td>
<td>0.497</td>
</tr>
<tr>
<td>Contacts (training 20)</td>
<td>0.611</td>
<td>0.498</td>
</tr>
</tbody>
</table>

tensor based recommendation: consistently better accuracy in preliminary experiments, now under evaluation
Decentralized setting: pro & contra argumentation

- multiple accounts for different resource types
- space limitations (e.g. max 200 photos in Flickr)
- censorship, rank manipulations
- single point of failure

Distributed Tagging System?
- tag any kind of personal data on the Desktop
- share and browse tagged data in a P2P network

Our implementation: Tagster
open source, available at http://isweb.uni-koblenz.de
Meta Methods: General Model

given: set of methods $V = \{v_1, ..., v_k\}$, confidence grades $res(v_i, d)$ for document $d$

Meta result (restrictivity by thresholds $t_1$ and $t_2$, tuning by weights $w(v_i)$):

$$Meta(d) = \begin{cases} 
+1 & \text{if } \sum_i res_i(d) \cdot w(v_i) > t_1 \\
-1 & \text{if } \sum_i res_i(d) \cdot w(v_i) < t_2 \\
0 & \text{otherwise}
\end{cases}$$

Special cases:
- “Unanimous Decision”
- “Voting”
- “Weighted Average” (e.g., weighted by some quality estimator)
Collaborative organization of document collections

Given: set of methods $V = \{v_1, ..., v_L\}$, "unanimous decision"

$$X_i = \begin{cases} 
1 & \text{if } v_i \text{ assigns document correctly} \\
0 & \text{otherwise}
\end{cases}$$

$$P(X_1 = 1, ..., X_L = 1) = P(X_1 = 1) \cdot \prod_{i=1}^{L-1} \frac{P(X_i = 1)P(X_{i+1} = 1) + \text{cov}(X_i, X_{i+1})}{P(X_i = 1)}$$

$$\text{error(meta)} = P(X_1 = 0, ..., X_L = 0|X_1 = .. = X_L)$$

$$\text{loss(meta)} = 1 - P(X_1 = ... = X_L)$$
Decentralized Collaboration: Results

Accuracy: del.icio.us, Junk=1/2

Junk Reduction and Document Loss: del.icio.us, Junk=1/2
Distributed scenario: an example

iuf: \( \log(5/2) = 0.4 \)
Distributed scenario: example (2)

\[ \text{update required!} \]
The PINTS approach

- Index peers monitor feature vector accuracy for their tags
- Compare feature approximation with the tag's true $iuf$ value

**BOB's approximation:**

$$
\mathbf{v}_{BOB}(\theta) = \begin{cases} 
    tf(t_1) \cdot (a_{t_1} \cdot \theta + b_{t_1}) \\
    \vdots \\
    tf(t_m) \cdot (a_{t_m} \cdot \theta + b_{t_m}) \\
    \vdots \\
    tf(t_N) \cdot (a_{t_N} \cdot \theta + b_{t_N}) 
\end{cases} 
$$

**Index peer's view:**

$$
\mathbf{v}_{BOB,t_m}(\theta) = \begin{cases} 
    tf(t_1) \cdot (a_{t_1} \cdot \theta + b_{t_1}) \\
    \vdots \\
    tf(t_m) \cdot iuf_{true}^{t_m} \\
    \vdots \\
    tf(t_N) \cdot (a_{t_N} \cdot \theta + b_{t_N}) 
\end{cases} 
$$

- Index peer needs to know the other approximations
- Vector similarity must be above threshold $\delta$

$$
\text{sim}(\mathbf{v}^*, \mathbf{v}^*_{t_m}) > \delta \\
\text{sim}(\mathbf{v}^*, \mathbf{v}^*_{t_m}) = \frac{\mathbf{v}^* \cdot \mathbf{v}^*_{t_m}}{\|\mathbf{v}^*\| \|\mathbf{v}^*_{t_m}\|}
$$
The PINTS approach (2)

\[
\text{sim}(\mathbf{v}^*, \mathbf{v}^*_{t_m}) = \frac{\mathbf{v}^* \cdot \mathbf{v}^*_{t_m}}{||\mathbf{v}^*|| \cdot ||\mathbf{v}^*_{t_m}||}
\]

\[
\mathbf{v}^* \cdot \mathbf{v}^*_{t_m} = \sum_{t_i \in t_m} (\text{tf}(t_i)^2 \cdot (a_{t_i} \cdot \theta + b_{t_i})^2) + \text{tf}(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m}) \cdot iuf_{t_m}^{\text{true}}
\]

\[
||\mathbf{v}^*|| = \sqrt{\sum_{t_i \in t_m} (\text{tf}(t_i)^2 \cdot (a_{t_i} \cdot \theta + b_{t_i})^2) + \text{tf}(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m})^2}
\]

\[
||\mathbf{v}^*_{t_m}|| = \sqrt{\sum_{t_i \in t_m} (\text{tf}(t_i)^2 \cdot (a_{t_i} \cdot \theta + b_{t_i})^2) + \text{tf}(t_m)^2 \cdot (iuf_{t_m}^{\text{true}})^2}
\]

\[
A_{t_m} = \sum_{t_i \in t_m} (\text{tf}(t_i) \cdot a_{t_i}^2) \\
B_{t_m} = \sum_{t_i \in t_m} (\text{tf}(t_i) \cdot a_{t_i} \cdot b_{t_i}) \\
C_{t_m} = \sum_{t_i \in t_m} (\text{tf}(t_i)^2 \cdot b_{t_i}^2)
\]

\[
\mathbf{v}^* \cdot \mathbf{v}^*_{t_m} = A_{t_m} \theta^2 + 2B_{t_m} \theta + C_{t_m} + \text{tf}(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m}) \cdot iuf_{t_m}^{\text{true}}
\]

\[
||\mathbf{v}^*|| = \sqrt{A_{t_m} \theta^2 + 2B_{t_m} \theta + C_{t_m} + \text{tf}(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m})^2}
\]

\[
||\mathbf{v}^*_{t_m}|| = \sqrt{A_{t_m} \theta^2 + 2B_{t_m} \theta + C_{t_m} + \text{tf}(t_m)^2 \cdot (iuf_{t_m}^{\text{true}})^2}
\]
PINTS: update strategy

\[ \text{sim}(\mathbf{v}^*, \mathbf{v}_{t_m}) = \frac{\mathbf{v}^* \cdot \mathbf{v}_{t_m}}{\|\mathbf{v}^*\| \|\mathbf{v}_{t_m}\|} \]

\[ \mathbf{v}^* \cdot \mathbf{v}_{t_m} = A_{t_m} \theta^2 + 2B_{t_m} \theta + C_{t_m} + tf(t_m) \cdot (a_{t_m} \cdot \theta + b_{t_m}) \cdot iuf_{t_m} \]

\[ \|\mathbf{v}^*\| = \sqrt{A_{t_m} \theta^2 + 2B_{t_m} \theta + C_{t_m} + tf(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m})^2} \]

\[ \|\mathbf{v}_{t_m}\| = \sqrt{A_{t_m} \theta^2 + 2B_{t_m} \theta + C_{t_m} + tf(t_m)^2 \cdot (iuf_{t_m})^2} \]
PINTS updates: Beispiel

\[
v^*_\text{BOB}(\theta) = \begin{pmatrix} 65 \cdot 0.4 \\ 114 \cdot 0.1 \\ 49 \cdot 0.3 \\ 68 \cdot 0.2 \end{pmatrix}, \quad A_{\text{bamberg}} = 0, \quad B_{\text{bamberg}} = 0, \quad C_{\text{bamberg}} = 531.1
\]

\[
v^*_\text{TOM}(\theta) = \begin{pmatrix} 37 \cdot 0.4 \\ 71 \cdot 0.3 \\ 54 \cdot 0.1 \\ 69 \cdot 0.5 \end{pmatrix}, \quad A_{\text{bamberg}} = 0, \quad B_{\text{bamberg}} = 0, \quad C_{\text{bamberg}} = 1673.1
\]

\[
sim(v^*_\text{BOB}, v^*_{\text{BCB}, \text{bamberg}}) = 0.958 \quad \text{update required}
\]
\[
sim(v^*_\text{TOM}, v^*_{\text{TOM}, \text{bamberg}}) = 0.989 \quad \text{no update required}
\]
PINTS Evaluation

Objectives
- check if (and how frequent) specified thresholds violated
- compare message complexity for various methods

Methodology
- use real world tagging traces (time-ordered tas assignm.)
  - flickr.com (~320k users, ~1.6m tags, ~28.2m resources)
  - del.icio.us (~533k users, ~2.3m tags, ~17.3m resources)
- replay tagging traces in P2P simulation
- measure inaccurate feature vectors, message complexity
- evaluate against interval-based update
PINTS Evaluation (2)

PINTS: higher accuracy than interval updates
PINTS Evaluation (2)

PINTS: high accuracy at low message complexity
Conclusions

Focused search & recommendation in Web 2.0 folksonomies:

- IR-like problem formalization
- Personal and social aspects/dimensions are important
- Multi-dimensional setting helps to improve accuracy
- Can be realized for centralized and decentralized architectures

Future work

- bridging the semantic gap between low-level and high-level features
- decentralized computations on large sparse matrices (e.g. P2P based PageRank or HITS estimation)
- evaluation methodology for Web 2.0 applications
- better understanding of Web 2.0 evolution patterns
thank you