Thematically Focused Search in Web 2.0 Folksonomies

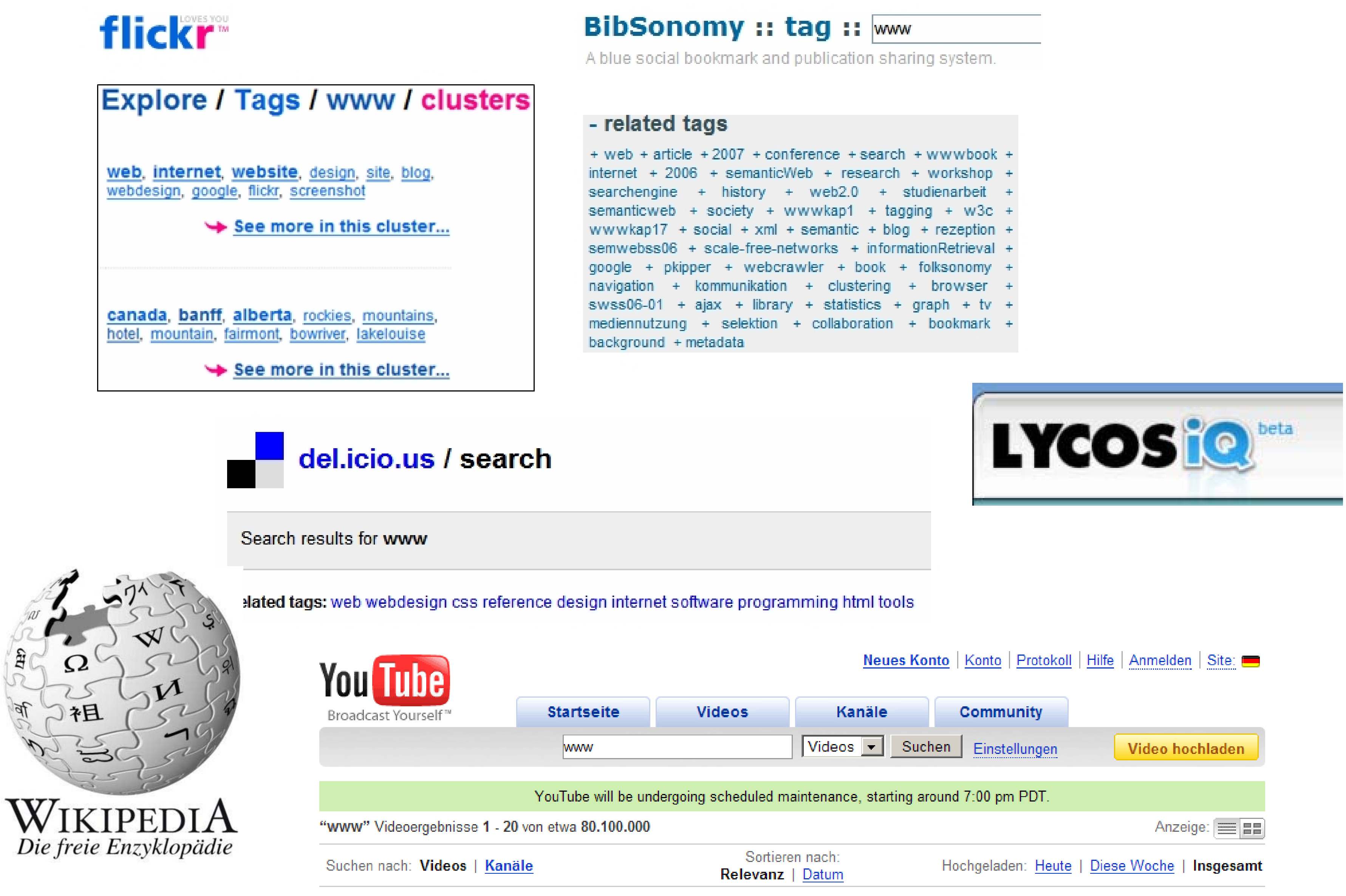
Sergej Sizov, Olaf Görlitz
University of Koblenz-Landau
Germany

UNIVERSITÄT KOBLENZ · LANDAU

Vannevar Bush's Memex (1945) & Mass Intelligence (2008)

can this huge mass of users give us new insights that would not be possible by considering individual contributions ?!

The mass makes the difference?

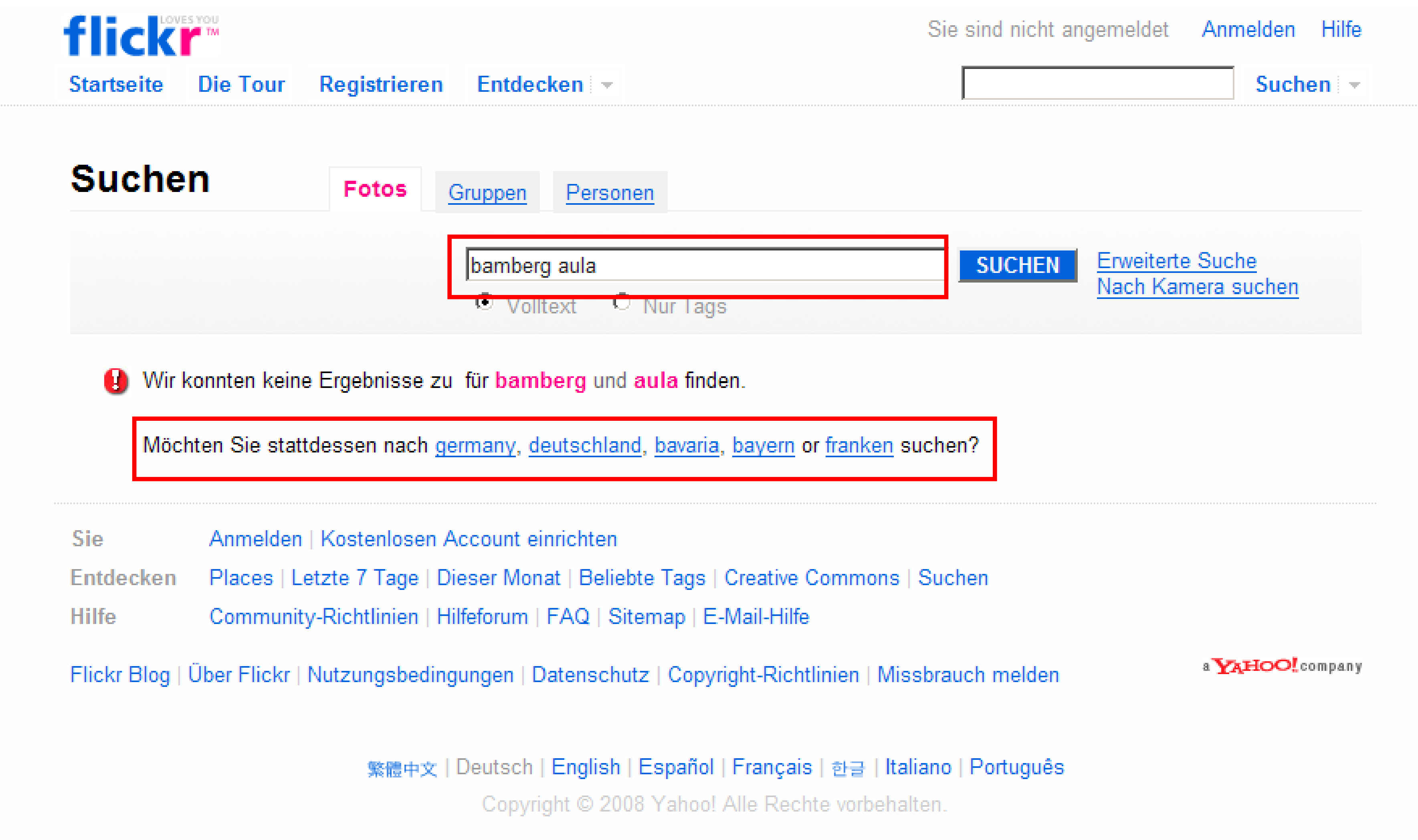


Flickr example – what you get ..

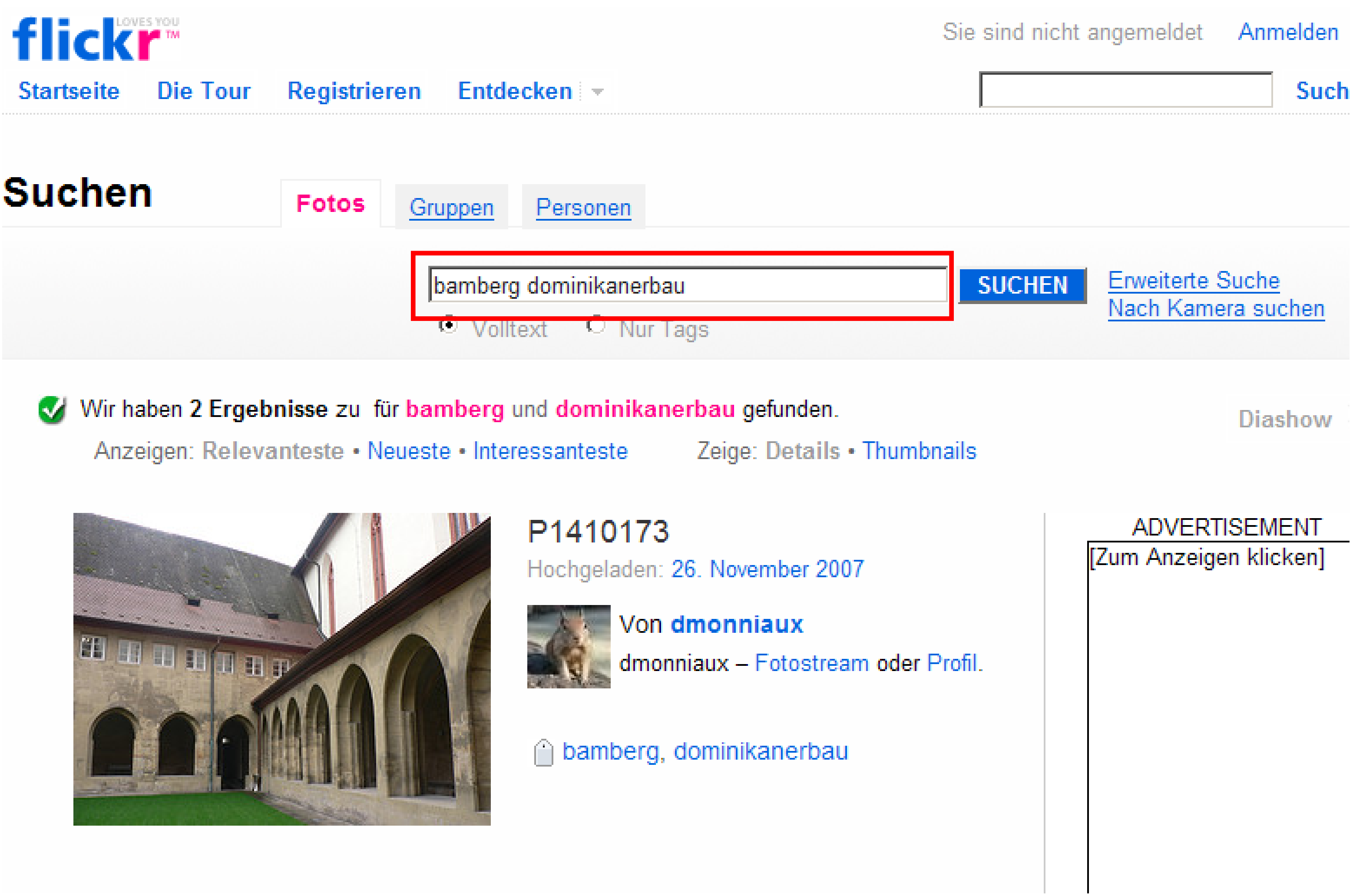
Query: "Athos fire"

У Светој српској царској лаври Хиландару на Светој Гори Атонској, у ноћи између 3. и 4. марта 2004. године, избио је пожар великих размера.

Flickr: queries with low recall



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:

users $u \in U$ tags $t \in T$ resources $r \in R$

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

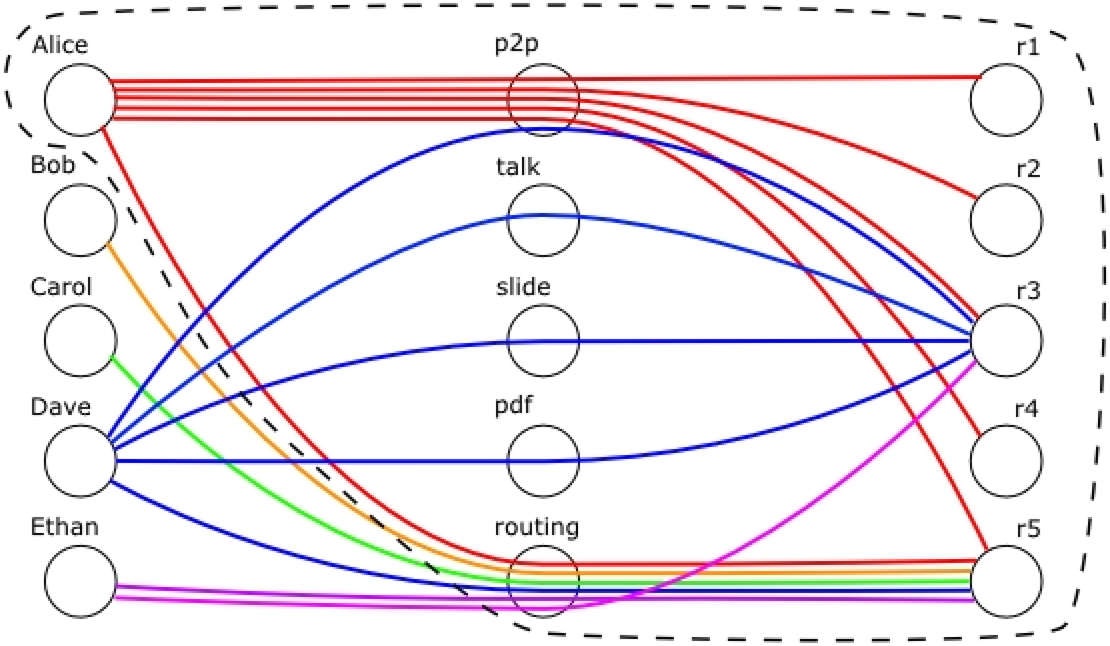
- user-centric: $T_u := (Y_u, f), Y_u \subseteq u \times T \times R,$
- resource-centric: $T_r := (Y_r, f), Y_u \subseteq U \times T \times r,$
- community-specific: $T_{U^*} := (Y_{U^*}, f), \quad Y_{U^*} \subseteq U^* \times T \times R$
- collection-specific $T_{R^*} := (Y_{R^*}, f), \quad Y_{R^*} \subseteq U \times T \times R^*$
- 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



$$
\begin{aligned}
if(i) &= (a_i, b_i) \\
iif(i) &= \left(\log\frac{|J|}{|J^*|}, \log\frac{|K|}{|K^*|}\right), \\
& \quad j \in J^* / k \in K^* : \\
& \quad (i,j,k) \in Y^* \\
weight(i) &= iif(i) \cdot if(i)^T
\end{aligned}
$$

.. defined analogously to tf·idf

tag features:

$$iif \cdot if(p2p) = (\log\frac{5}{2}, \log\frac{5}{5}) \cdot (6,6)^T$$

$$iif \cdot if(talk) = (\log\frac{5}{1}, \log\frac{5}{1}) \cdot (1,1)^T$$

$$iif \cdot if(slide) = (\log\frac{5}{1}, \log\frac{5}{1}) \cdot (1,1)^T$$

$$iif \cdot if(pdf) = (\log\frac{5}{1}, \log\frac{5}{1}) \cdot (1,1)^T$$

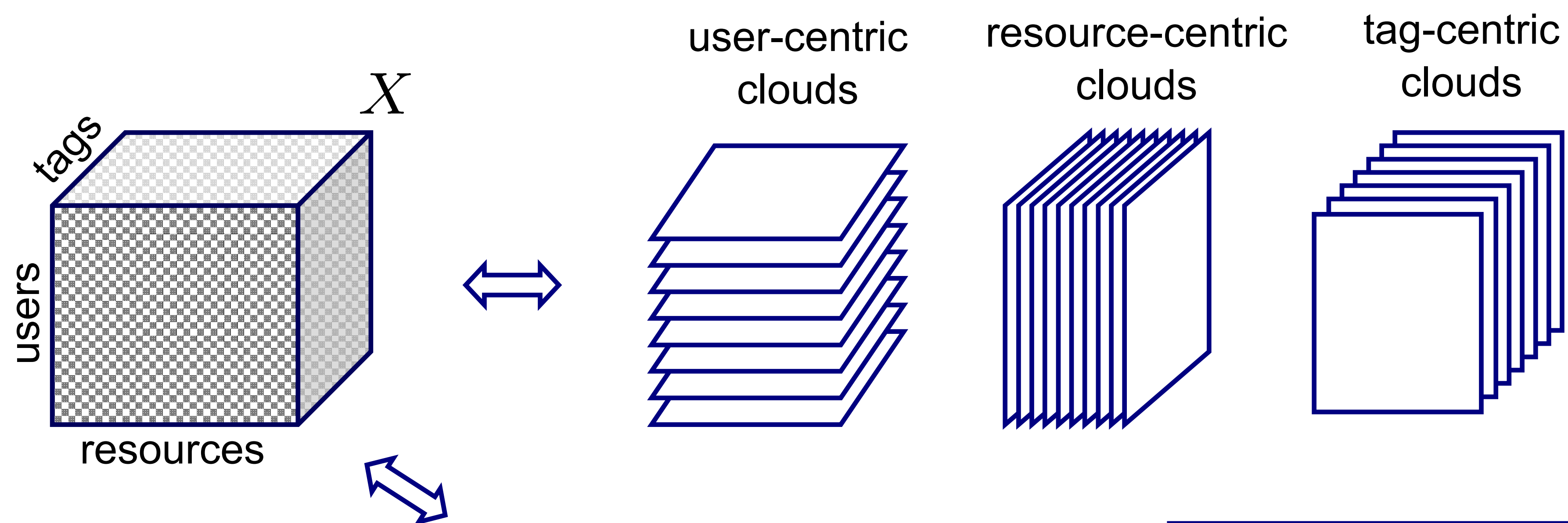
$$iif \cdot if(routing) = (\log\frac{5}{5}, \log\frac{5}{1}) \cdot (6,6)^T$$

Further dimensions of interest:

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

Generalization: the tensor model

idea: using multi-dimensional arrays for representing relationships



matricize:

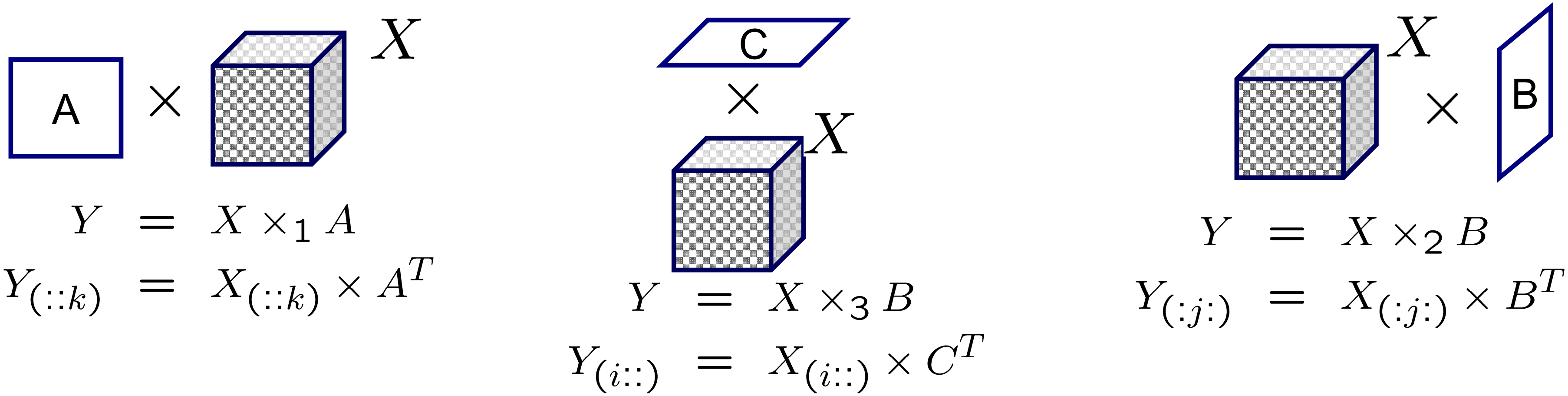
$X_{(users)} = \boxed{\text{community cloud}}$

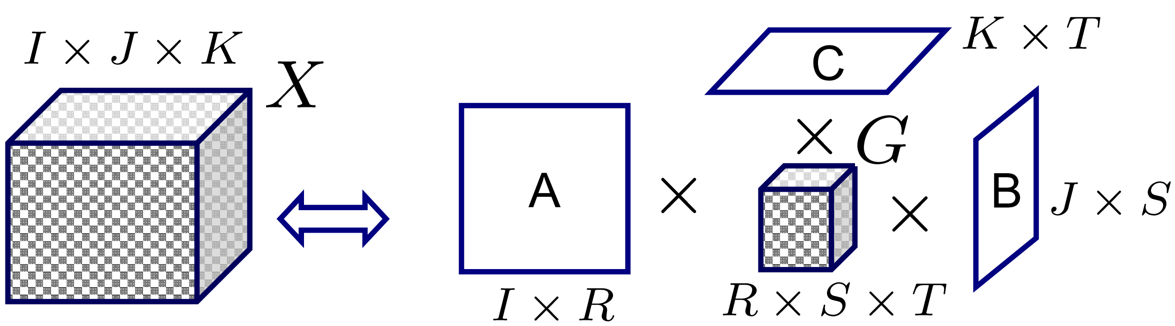
$X_{(resources)} = \boxed{\text{cloud for resources}}$

$X_{(tags)} = \boxed{\text{cloud for tags}}$

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,...





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 acording 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

User representation	Training:10 prec@10	Training:10 prec@20
Random	0.167	0.167
User items	0.259	0.268
Commented items	0.236	0.221
Favorites	0.872	0.727
Combined	0.854	0.713
	Training:20 prec@10	Training:20 prec@20
Random	0.167	0.167
Commented items	0.255	0.248
Favorites	0.918	0.851
Combined	0.899	0.828
	Training:40 prec@10	Training:40 prec@20
Random	0.167	0.167
Commented items	0.265	0.266
Favorites	0.933	0.903
Combined	0.914	0.876

recommending contacts

User representation	prec@5	prec@10
Random	0.167	0.167
User Items	0.574	0,472
Commented Items	0.576	0.473
Favorites	0.535	0.455
Contacts (training 10)	0.604	0.497
Contacts (training 20)	0.611	0.498

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 $V = \{v_1,\ldots,v_k\}$, 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

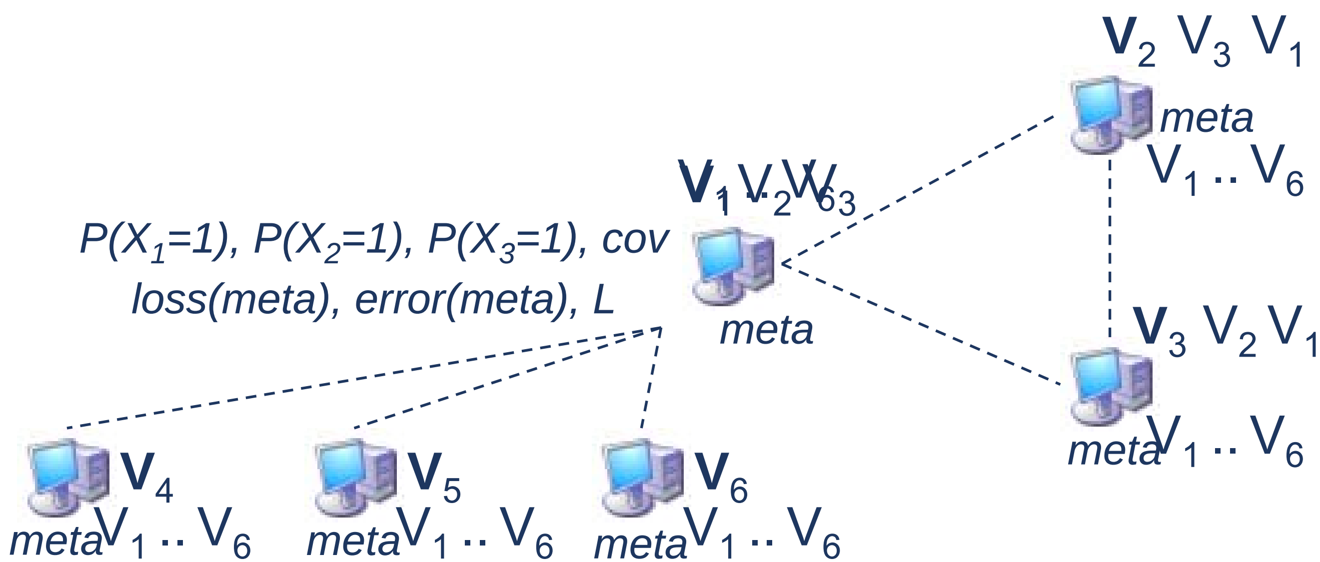
Given: set of methods $V=\{ v_1,\dots,v_L \}$, „unanimous decision"

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

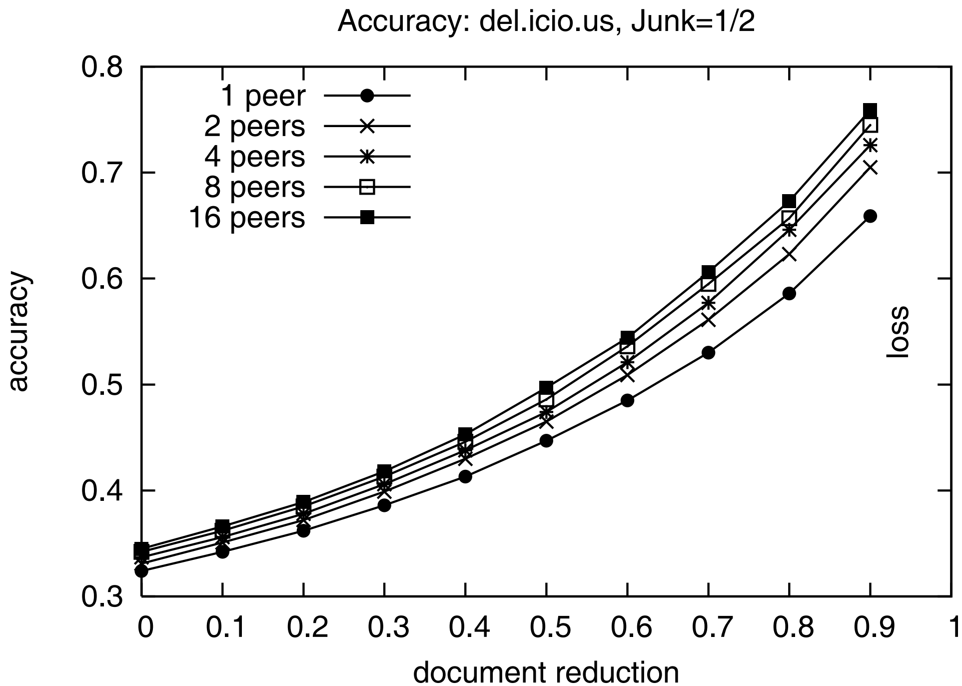
$$P(X_1 = 1, \dots, X_L = 1) \;=\; P(X_1 = 1) \cdot \prod_{i=1}^{L-1} \frac{P(X_i = 1)P(X_{i+1} = 1) + cov(X_i, X_{i+1})}{P(X_i = 1)}$$

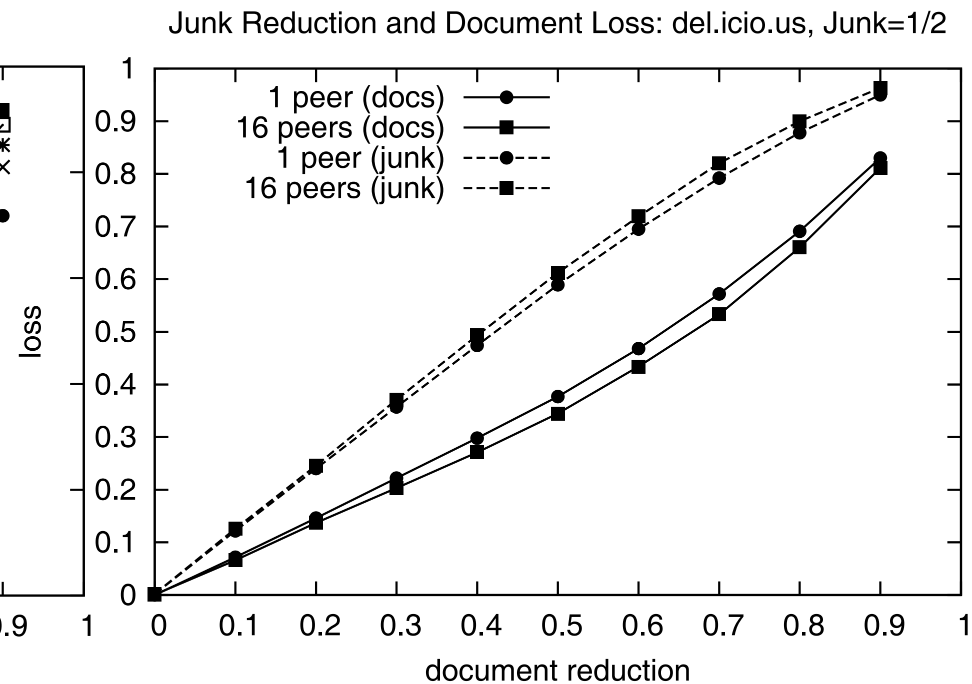
$$error(meta) \;=\; P(X_1 = 0, \dots, X_L = 0 | X_1 = .. = X_L)$$

$$loss(meta) \;=\; 1 - P(X_1 = \dots = X_L)$$

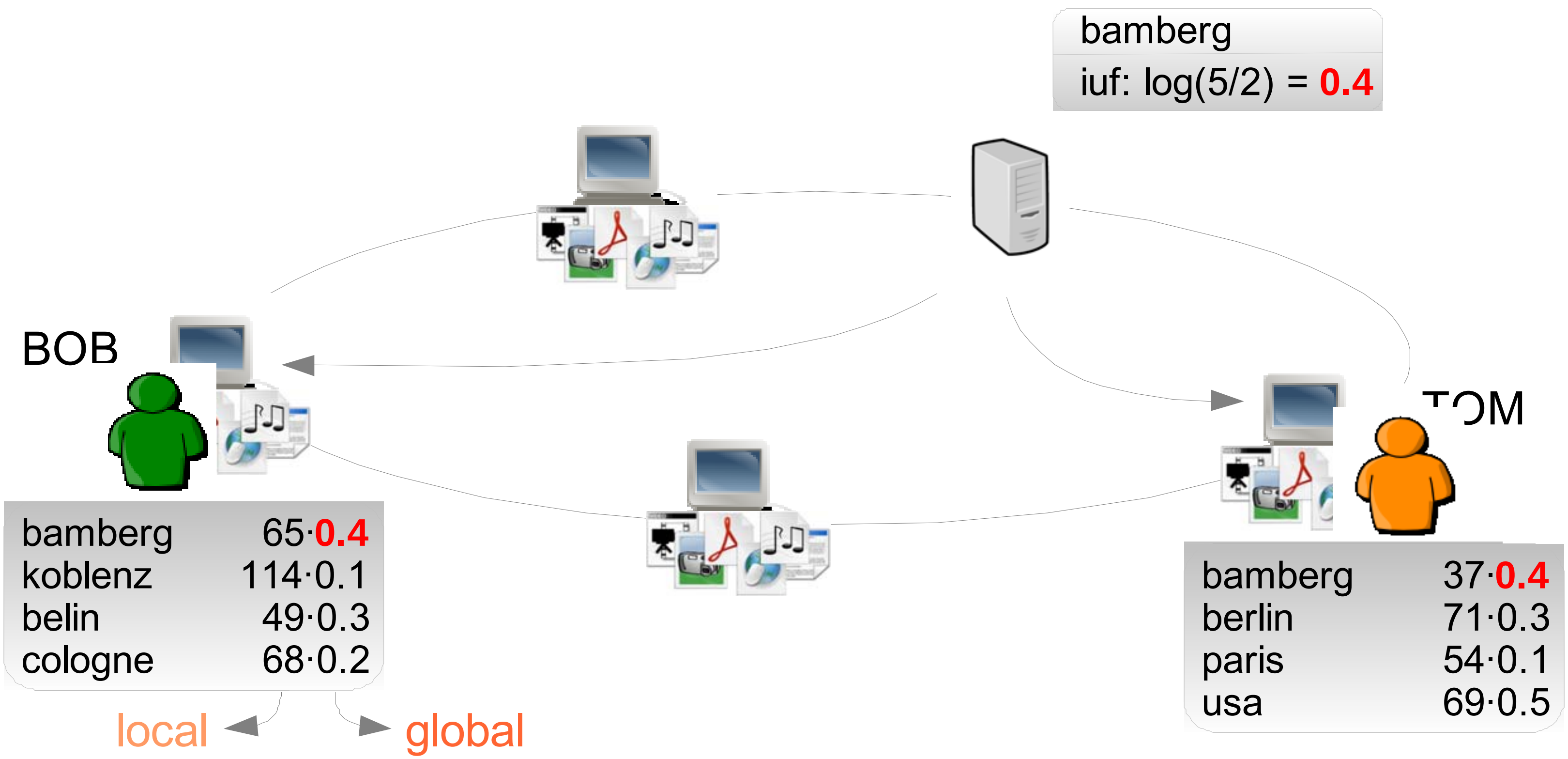


Decentralized Collaboration: Results

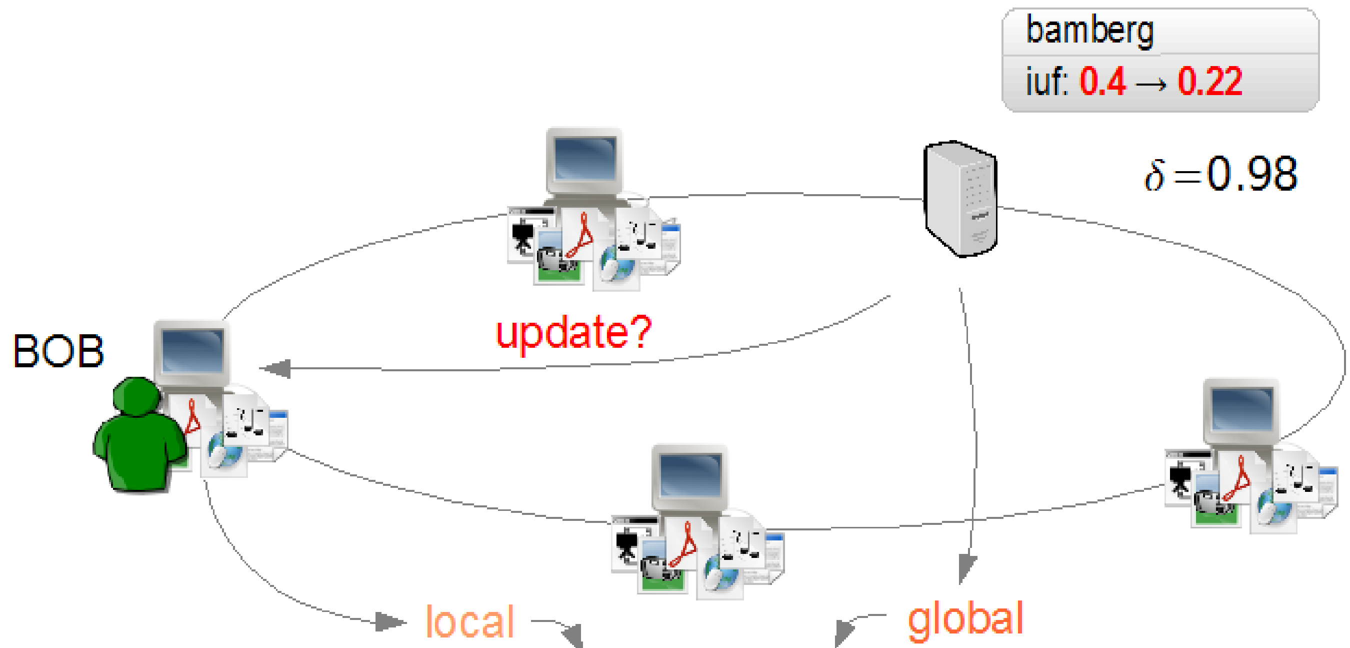




Distributed scenario: an example



Distributed scenario: example (2)



$$sim(v^{\star}_{BOB}, v_{BOB}) = sim\left(\begin{bmatrix} 65 \cdot 0.4 \\ 114 \cdot 0.1 \\ 49 \cdot 0.3 \\ 68 \cdot 0.2 \end{bmatrix} \begin{bmatrix} 65 \cdot 0.22 \\ 114 \cdot 0.1 \\ 49 \cdot 0.3 \\ 68 \cdot 0.2 \end{bmatrix} \right) = 0.958$$

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:

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

index peer's view:

$$v^{\circ}_{BOB,t_m}(\theta) = \begin{pmatrix} tf(t_1)\cdot(a_{t_1}\cdot\theta + b_{t_1}) \\ \vdots \\ \boxed{tf(t_m)\cdot iuf^{true}_{t_m}} \\ \vdots \\ tf(t_N)\cdot(a_{t_N}\cdot\theta + b_{t_N}) \end{pmatrix}$$

- Index peer needs to know the other approximations
- Vector similarity must be above threshold δ

$$sim(v^{\star}, v^{\circ}_{t_m}) > \delta \qquad sim(v^{\star}, v^{\circ}_{t_m}) = \frac{v^{\star}\cdot v^{\circ}_{t_m}}{\|v^{\star}\|\,\|v^{\circ}_{t_m}\|}$$

The PINTS approach (2)

$$sim(v^\star, v^\circ_{t_m}) = \frac{v^\star \cdot v^\circ_{t_m}}{\|v^\star\| \, \|v^\circ_{t_m}\|}$$

$$v^\star \cdot v^\circ_{t_m} = \sum_{t_i \neq t_m} \left(tf(t_i)^2 \cdot (a_{t_i} \cdot \theta + b_{t_i})^2\right) + tf(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m}) \cdot iuf^{true}_{t_m}$$

$$\|v^\star\| = \sqrt{\sum_{t_i \neq t_m} \left(tf(t_i)^2 \cdot (a_{t_i} \cdot \theta + b_{t_i})^2\right) + tf(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m})^2}$$

$$\|v^\circ_{t_m}\| = \sqrt{\sum_{t_i \neq t_m} \left(tf(t_i)^2 \cdot (a_{t_i} \cdot \theta + b_{t_i})^2\right) + tf(t_m)^2 \cdot (iuf^{true}_{t_m})^2}$$

$$A_{t_m} = \sum_{t_i \neq t_m} \left(tf(t_i)^2 \cdot a^2_{t_i}\right) \qquad B_{t_m} = \sum_{t_i \neq t_m} \left(tf(t_i)^2 \cdot a_{t_i} \cdot b_{t_i}\right) \qquad C_{t_m} = \sum_{t_i \neq t_m} \left(tf(t_i)^2 \cdot b^2_{t_i}\right)$$

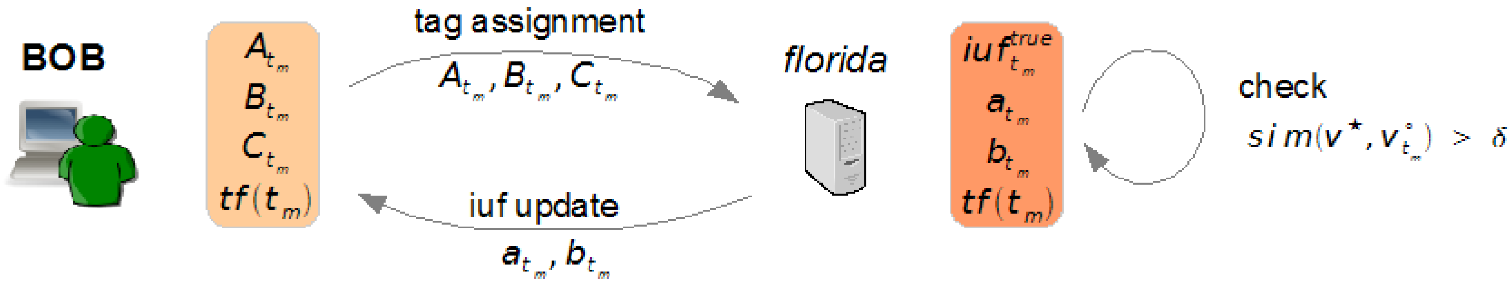
$$v^\star \cdot v^\circ_{t_m} = A_{t_m}\theta^2 + 2B_{t_m}\theta + C_{t_m} + tf(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m}) \cdot iuf^{true}_{t_m}$$

$$\|v^\star\| = \sqrt{A_{t_m}\theta^2 + 2B_{t_m}\theta + C_{t_m} + tf(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m})^2}$$

$$\|v^\circ_{t_m}\| = \sqrt{A_{t_m}\theta^2 + 2B_{t_m}\theta + C_{t_m} + tf(t_m)^2 \cdot (iuf^{true}_{t_m})^2}$$

PINTS: update strategy

$$sim(v^\star, v^\circ_{t_m}) = \frac{v^\star \cdot v^\circ_{t_m}}{\|v^\star\| \, \|v^\circ_{t_m}\|}$$

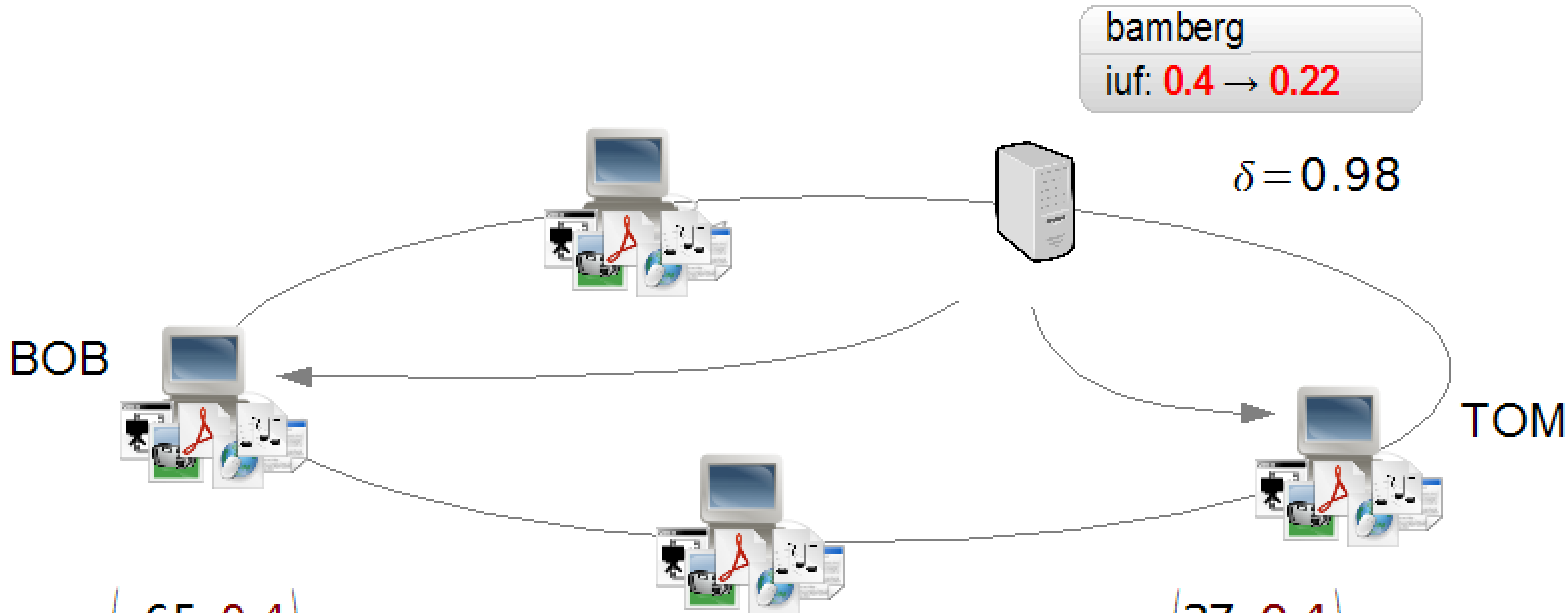


$$v^\star \cdot v^\circ_{t_m} = A_{t_m}\theta^2 + 2B_{t_m}\theta + C_{t_m} + tf(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m}) \cdot iuf^{true}_{t_m}$$

$$\|v^\star\| = \sqrt{A_{t_m}\theta^2 + 2B_{t_m}\theta + C_{t_m} + tf(t_m)^2 \cdot (a_{t_m} \cdot \theta + b_{t_m})^2}$$

$$\|v^\circ_{t_m}\| = \sqrt{A_{t_m}\theta^2 + 2B_{t_m}\theta + C_{t_m} + tf(t_m)^2 \cdot (iuf^{true}_{t_m})^2}$$

PINTS updates: Beispiel



$$v^{\star}_{BOB}(\theta) = \begin{pmatrix} 65 \cdot 0.4 \\ 114 \cdot 0.1 \\ 49 \cdot 0.3 \\ 68 \cdot 0.2 \end{pmatrix} \qquad \begin{aligned} A_{bamberg} &= 0 \\ B_{bamberg} &= 0 \\ C_{bamberg} &= 531.1 \end{aligned}$$

$$v^{\star}_{TOM}(\theta) = \begin{pmatrix} 37 \cdot 0.4 \\ 71 \cdot 0.3 \\ 54 \cdot 0.1 \\ 69 \cdot 0.5 \end{pmatrix} \qquad \begin{aligned} A_{bamberg} &= 0 \\ B_{bamberg} &= 0 \\ C_{bamberg} &= 1673.1 \end{aligned}$$

$sim(v^{\star}_{BOB}, v^{\circ}_{BOB,bamberg}) = 0.958$ **update required**

$sim(v^{\star}_{TOM}, v^{\circ}_{TOM,bamberg}) = 0.989$ **no update required**

PINTS Evaluation

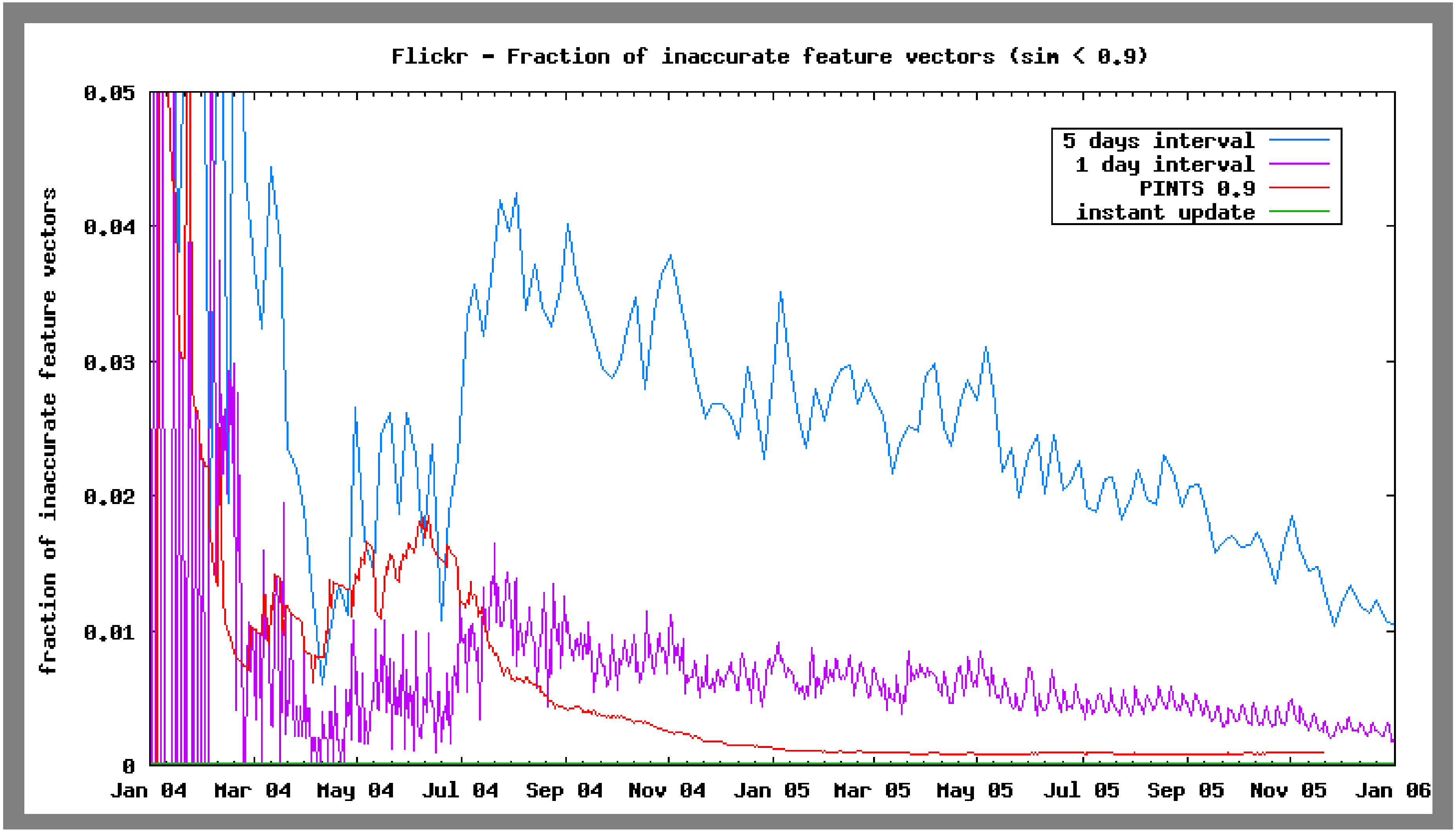
Objectives

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

Methodology

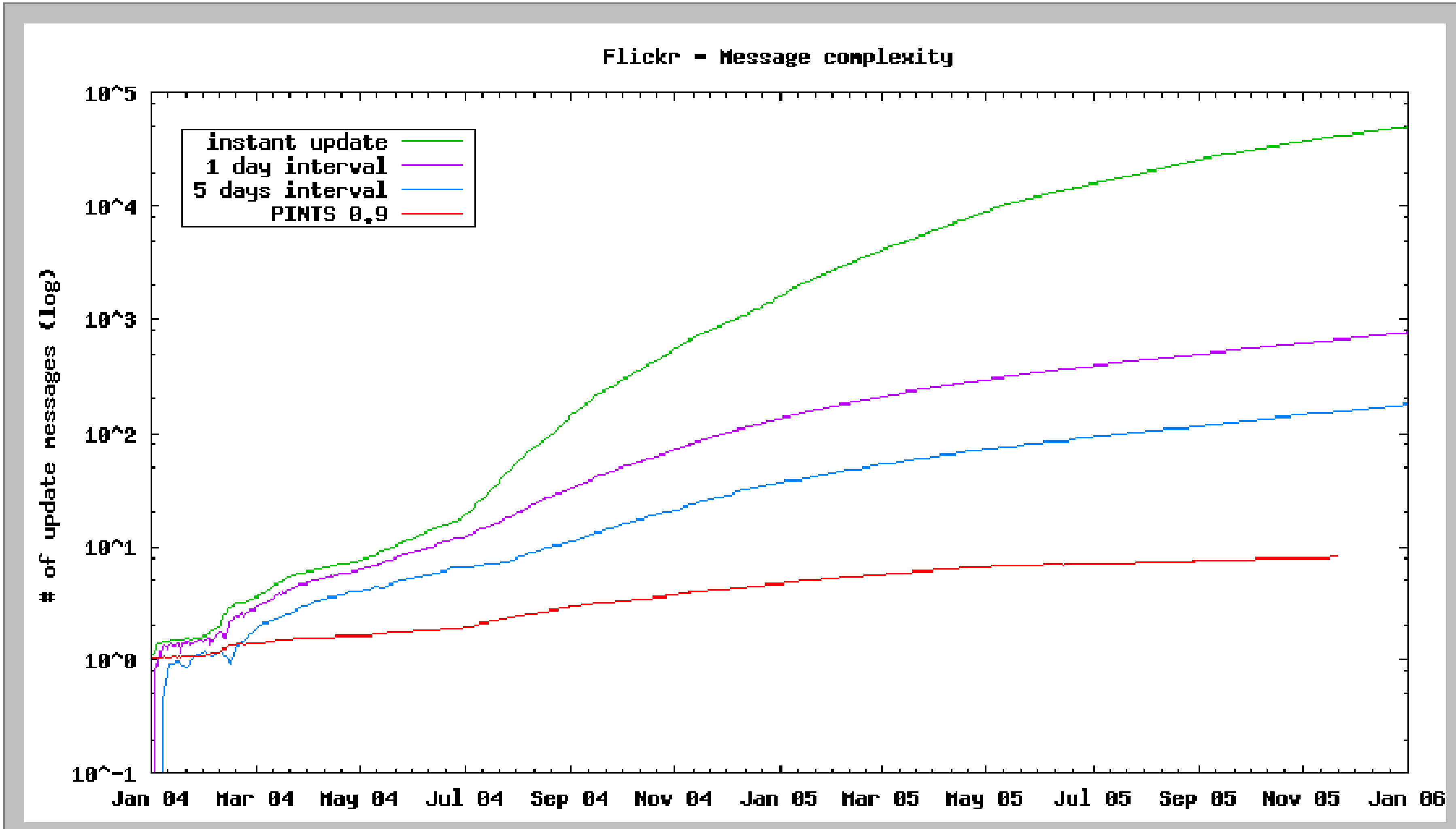
- use real world tagging traces (time-ordered tas assignm.)
  - flickr.com (~320k users, ~1.6m tags, ~28.2m resources)
  - del.icio.us (~533k users, ~2.3m tags, ~17.3m resources)
- replay tagging traces in P2P simulation
- measure inaccurate feature vectors, message complexity
- evaluate against interval-based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