An Analysis of Tag-Recommender Evaluation Procedures

Stephan Doerfel & Robert Jäschke

University of Kassel & L3S Research Center

doerfel@cs.uni-kassel.de & jaeschke@l3s.de

Social Bookmarking & Folksonomies

Social bookmarking systems: users can collect and manage resources like bookmarks, publications, images, videos, . . .

Folksonomy: underlying data structure that models the process of *users* creating *posts* by annotating *resources* with freely chosen keywords – so called *tags*

Through tagging, users collaboratively generate a corpus of publicly visible, annotated resources. Resources can be retrieved using the tags and can be shared with other users.

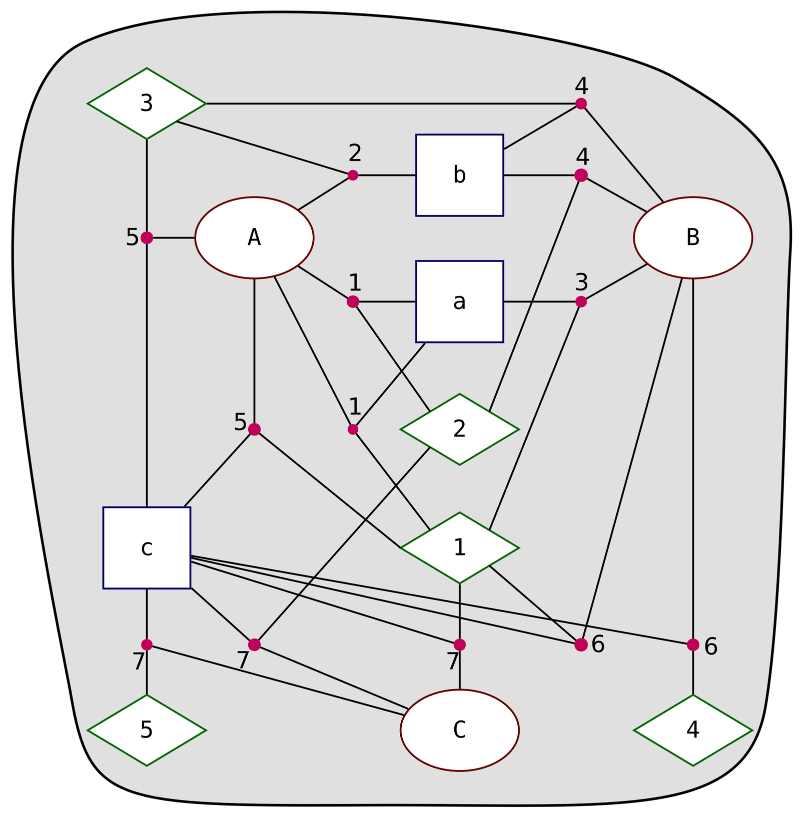
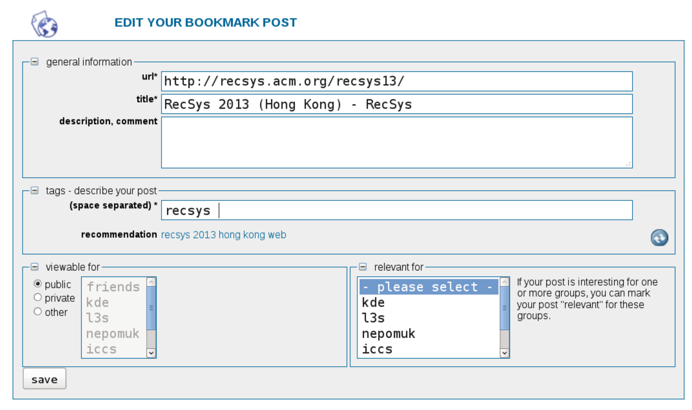


Figure 1: A folksonomy toy example with three users A, B, C (◯), three resources a, b, c (□), and five tags 1, 2, 3, 4, 5 (◇) in seven posts (•)

Tag Recommendations in Folksonomies



Tag recommenders assist users when they post a resource. The goal is to reduce the effort for users and to encourage the use of tags.

Tag Recommendation Task: Given a user u and a resource r, recommend tags that the user u will find suitable for the resource r.

- Several algorithms have been proposed
- Evaluation often perfomed offline, using historical datasets
- Experiments suffer from data sparsity and the cold start problem
- *Cores* can densify data, remove low-frequency users, resources, and tags

Graph-Core

Remove nodes with less than l edges. Repeat iteratively, to yield the *graph-core* at level l.

Drawbacks:

- Tags need to occur in l posts but users or resources in just one with at least l tags.
- Posts get split.

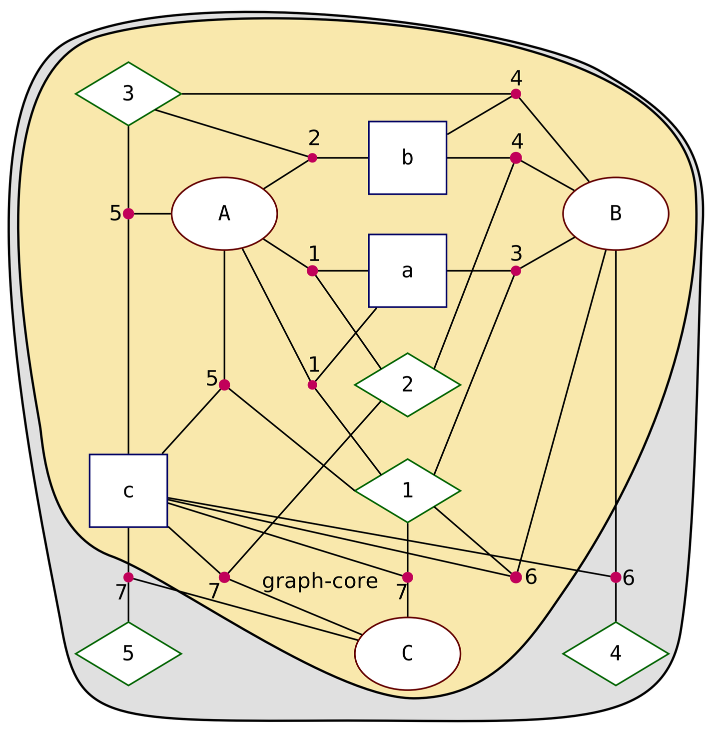


Figure 2: The graph-core at level 2.

Cores

- Folksonomies are modelled as graphs, where users, resources and tags form the node set. User u is connected to resource r and tag t by a hyperedge, if u assigned t to r.
- Cores reduce the dataset by iteratively removing nodes (and all connected edges) until all remaining nodes satisfy some specific property.
- Seidman and later Batagelj and Zaveršnik developed the theory on cores to analyze graphs.
- For tag recommendations they are used commonly to yield denser subgraphs of a folksonomy.

Does the choice of core influence the evaluation?

Post-Core

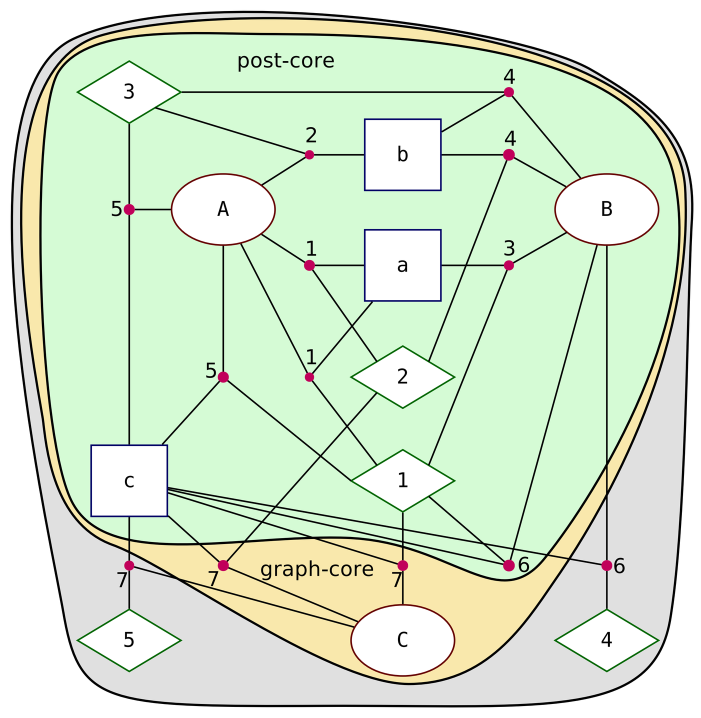


Figure 3: The post-core at level 2.

Remove nodes that do not occur in at least l posts. Repeat iteratively, to yield the *post-core* at level l.

Drawback: Posts still get split: individual tags and (thus tag assignments) are removed, others – possibly of the same post – stay in the core.

Public Datasets

BibSonomy: publications (publ) or bookmarks (book)
`http://www.bibsonomy.org/`

Delicious: bookmarks (deli)
`http://www.delicious.com/`

	#users	#res.	#tags	#tas	#posts	chosen l
publ	4 777	94 427	57 639	397 081	109 984	2, 3, 4, 5, 10
book	4 959	231 907	80 603	1 032 037	268 589	2, 3, 4, 5, 6
deli	75 071	2 999 487	397 028	17 280 065	7 268 305	2, 3, 5, 10, 20

The datasets can be downloaded from `http://www.tagora-project.eu/data/#delicious` and `http://www.kde.cs.uni-kassel.de/bibsonomy/dumps/`.

Cleansing

Before the experiments we conducted appropriate preprocessing to remove automatic imports by

- eliminating posts created at exactly the same time by the same user
- ignoring tag assignments with the tags *imported, public, system:imported, nn, system:unfiled*

Additionally, all tags were converted to lower case, and all characters which were neither numbers nor letters were removed.

Evaluation Methodology

LeavePostOut: For each user u conduct the following experiment: Select one post of u at random, remove it from the data and recommend tags for its resource for that user.

Repeat LeavePostOut for each user 5 times for statistical validity.

Evaluation: Compare the recommended tags to the actual tags of the left-out posts. Use precision@k, recall@k, and MAP to evaluate the recommender quality.

Evaluate influence of cores: Repeat the experiments on cores at different levels with five well established tag recommendation algorithms: *most popular tags*, *most popular tags by resource*, *most popular tags by user*, *adapted PageRank*, and *FolkRank*.

Recommendation Performance Depends on Core Type and Level

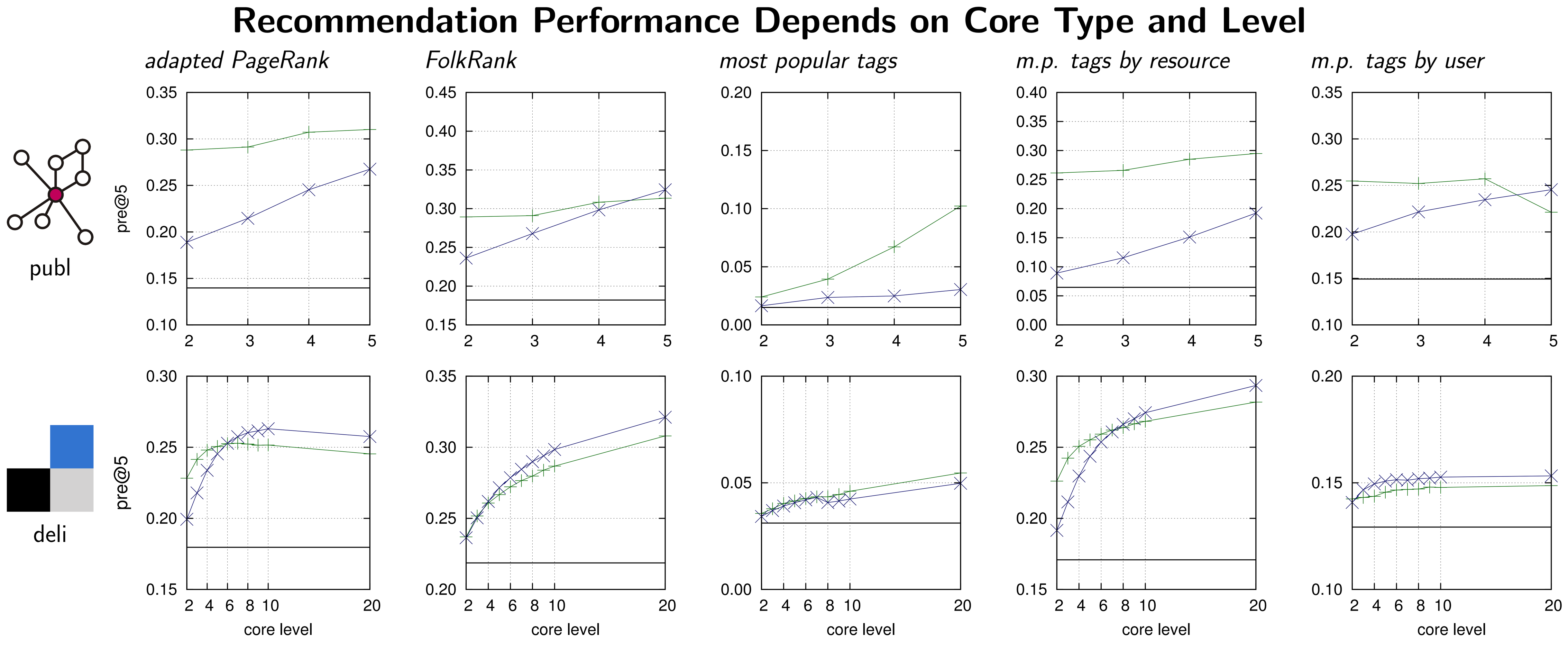


Figure 4: The performance of different tag recommendation algorithms using the graph-core as dataset and samples from the graph-core (×) or the post-core (+) for LeavePostOut. For the latter we use the property that the post-core is always a subset of the graph-core.

Recommender Ranking Correlation

Table 1: The mean pairwise Pearson's r, the number of discordant pairs d (with standard deviation σ) in the algorithm rankings on graph-cores and post-cores of different levels. Each ranking is a list of the five recommendation algorithms, ordered by their recommendation quality according to one of the measures MAP, pre@5 or rec@5.

dataset	metric	avg. r	σ	avg. d	σ
publ	MAP	0.890	0.093	1.491	1.069
publ	pre@5	0.886	0.101	1.636	1.007
publ	rec@5	0.894	0.099	1.564	1.014
book	MAP	0.899	0.093	1.491	1.069
book	pre@5	0.870	0.116	1.564	1.151
book	rec@5	0.902	0.091	1.455	1.068
deli	MAP	0.989	0.011	0.545	0.503
deli	pre@5	0.987	0.012	0.545	0.503
deli	rec@5	0.988	0.011	0.545	0.503

Results:

- High correlations among the rankings of algorithms
- On the smaller datasets (publ, book) on average one or two recommenders switch their positions in their ranking
- Higher consistency on the larger dataset deli

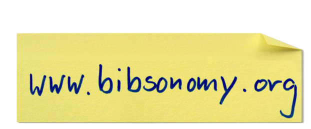
Conclusion

- Recommenders perform differently in different core setups of the same dataset.
- Evaluating recommender performance on another core type or at another core level might cause changes in the results
- Focusing on one particular core can produce non-stable results.
- No guarantee that the best recommender in one setup is also the best in another setup (even on the same dataset).
- But: Even cores at higher levels yield correlated results to those of the raw-data.

→ Evaluation should always be performed either directly on the raw data or on several core types and levels.

→ Compare recommenders on several smaller subsets of the raw data to get an impression of their overall performance.

Recommendations



- Test your recommender within a real system!
- Framework for tag and item recommendations
- Contact: `doerfel@cs.uni-kassel.de`