Spam filtering, ranking and recommendation systems

Andras Benczur

Insitute for Computer Science and Control

Hungarian Academy of Sciences

benczur@sztaki.mta.hu

http://datamining.sztaki.hu

Supported by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956)

WP4: Applications of new tools and algorithms to real-world network structures

- Milestone M4: Spam Filtering
- Milestone M7: Protocols for large-scale network processing
- Milestone M13: Characterization of ranking of Wikipedia and other networks
- (Milestone M14: Characterization of time evolving Web structures; Contribution to recommender Milestones)

WP4 main goal: collaboration of Physicists, Mathematicians and CS for applying new theoretical results for practical problems

Overview

- Web classification, spam filtering
- Temporal ranking, Wikipedia experiments
- Last.fm network recommenders
- Twitter: Andreas Kaltenbrunner’s collection and a 1B’n Firehose
- Distributed systems for very large problems
- The SZTAKI Text Mining Center test bed

Hardware

- 50 x old dual core Hadoop
- 5 x 8-core Hadoop/HBASE
- 2 x 32-core 256GB
- 260TB net Isilon

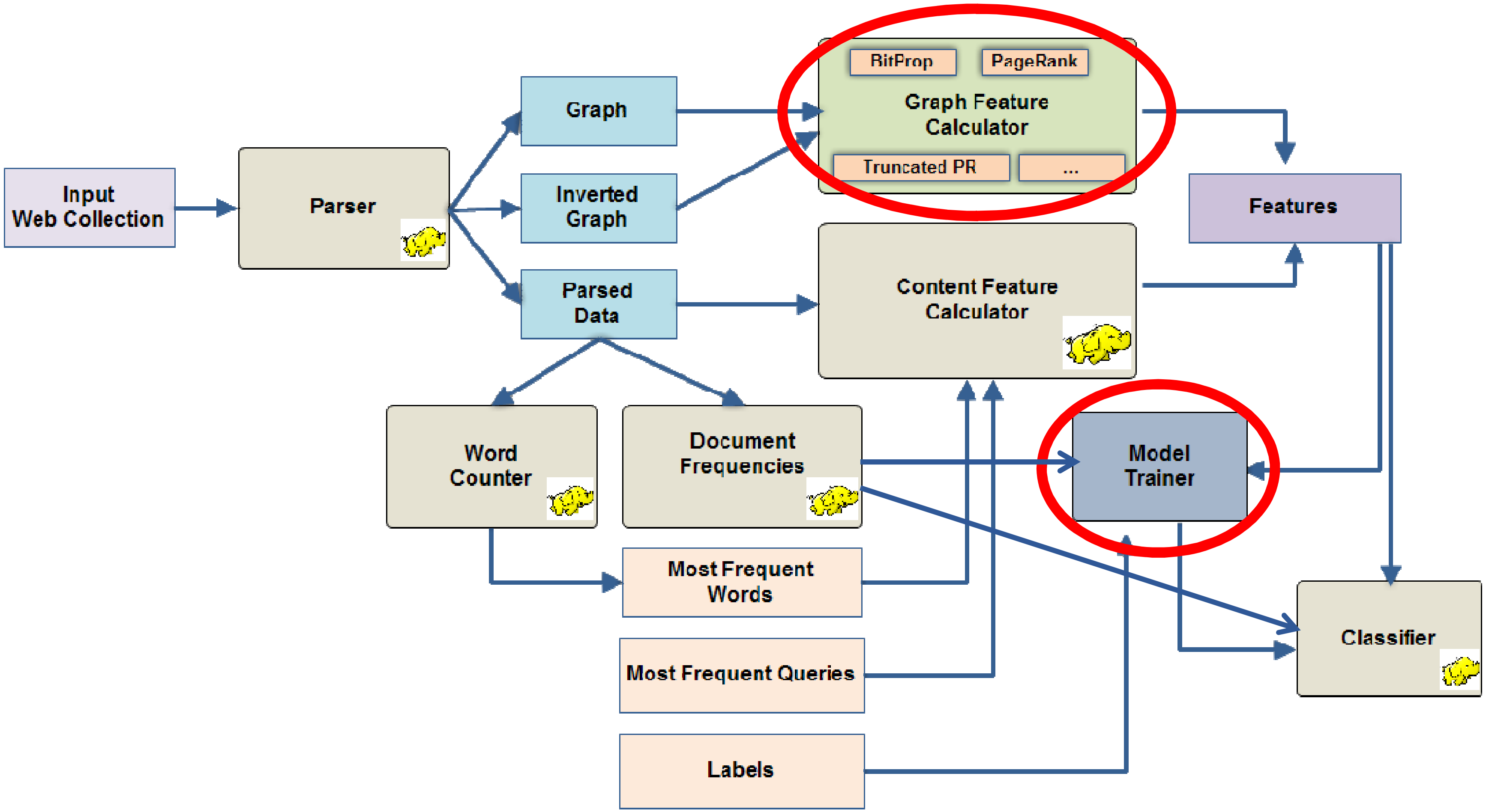
Selected publications

- A.Garzo, B.Daroczy, T.Kiss, D.Siklosi, and A.A.Benczur, **"Cross-Lingual Web Spam Classification"**, The 3rd Joint WICOW/AIRWeb Workshop on Web Quality in conj. WWW 2013, Rio de Janeiro, Brasil. May 13 (2013), Proceedings of the 22nd international conference on World Wide Web companion
- M.Erdelyi, A.A.Benczur, B.Daroczy, A.Garzo, T.Kiss and D.Siklosi, **"The classification power of Web features"**, Internet Mathematics, to appear (2013)
- J.Gobolos-Szabo, and A.A.Benczur, **"Temporal Wikipedia search by edits and linkage"**, SIGIR 2013 Workshop on Time-aware Information Access, 28 July - 1 August 2013, Dublin, Ireland
- R.Palovics, and A.A.Benczur, **"Temporal influence over the Last.fm social network"**, The 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2013 Niagara Falls, Canada, August 25-28, 2013

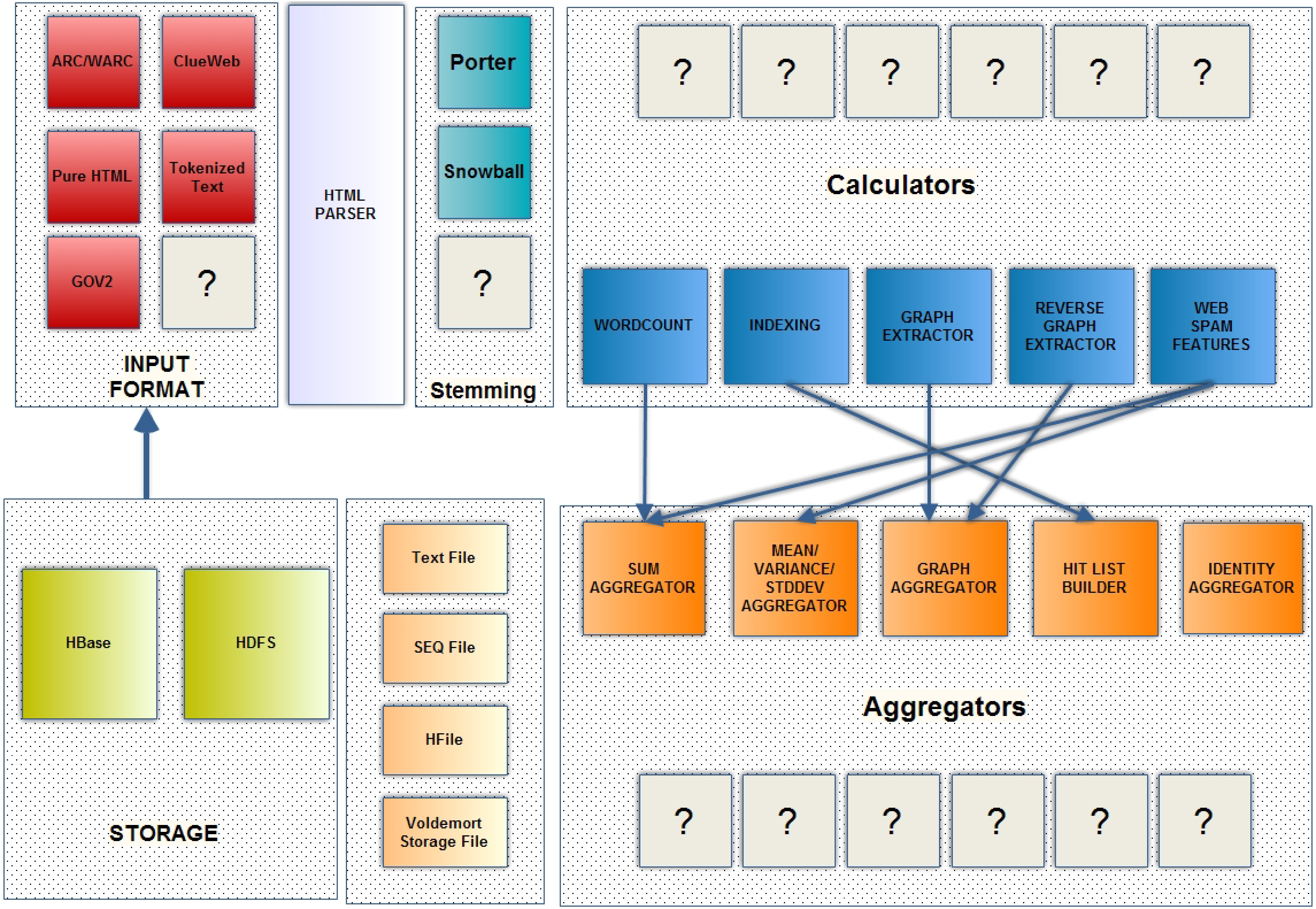
Web Classification

- Save resources, select quality and topic
- Legal regulation (porn, illicit content)
- Web scale data (Test: ClueWeb09 25TB – 0.5 Billion English language docs)
- Large set of features
  - Term frequency
    - tf.idf or BM25 scores for frequent terms
  - Content
    - DOM, HTML, HTTP elements
    - Appearance of popular terms
    - Term, n-gram statistics, compressibility
  - Linkage
    - PageRank (truncated variants; ratios)
    - Neighborhood (only approximate counting is possible)
    - TrustRank

Workflow (MapRed jobs indicated)



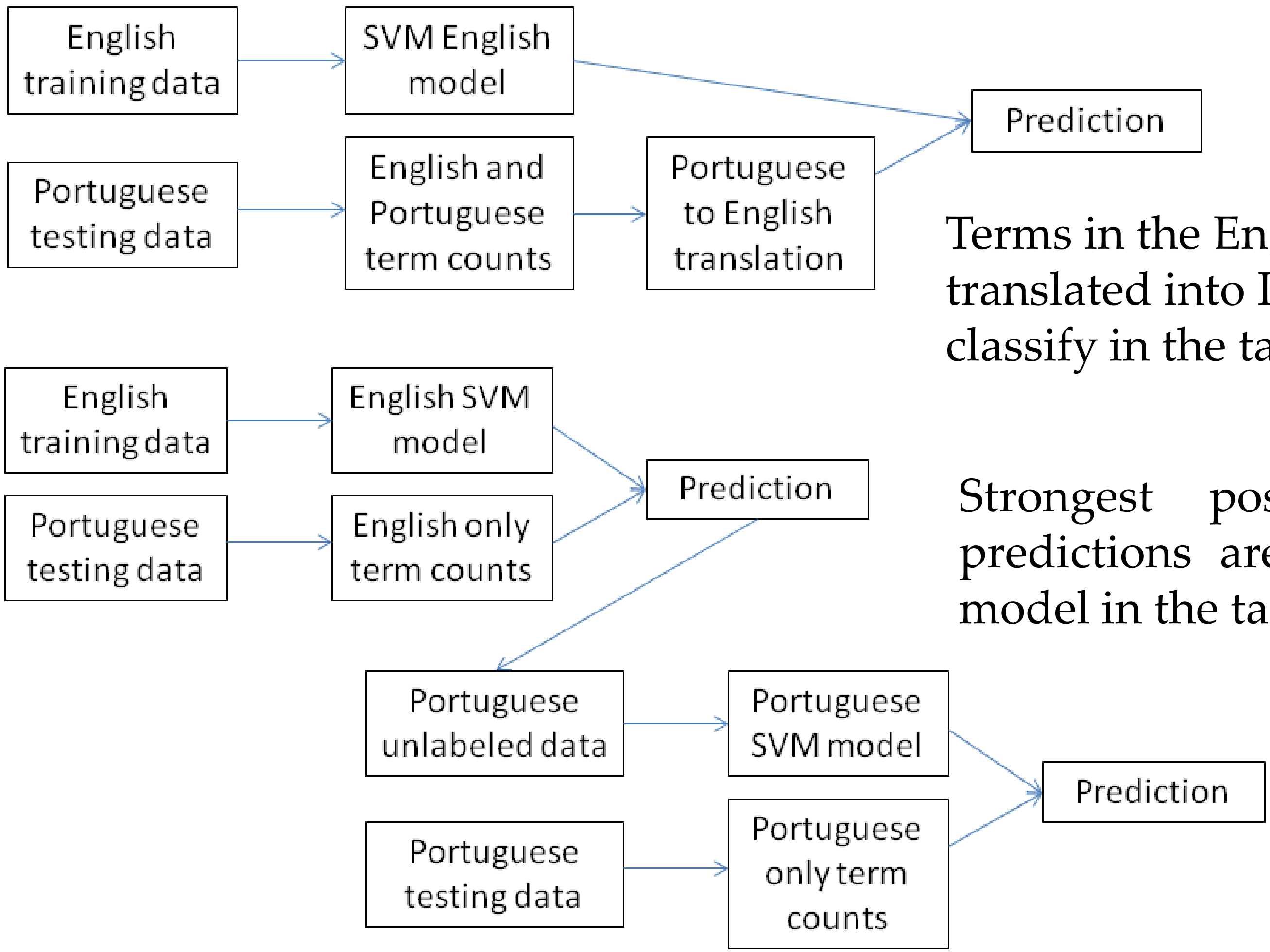
SZTAKI Web Processing Framework



https://github.com/garzoand/webspam-hadoop

Crosslingual Web Classification

- Expensive human labeling task language by language?
- How can models be "translated"?



Terms in the English model translated into Portuguese to classify in the target language.

Strongest positive and negative predictions are used for training a model in the target language.

Crawler integration

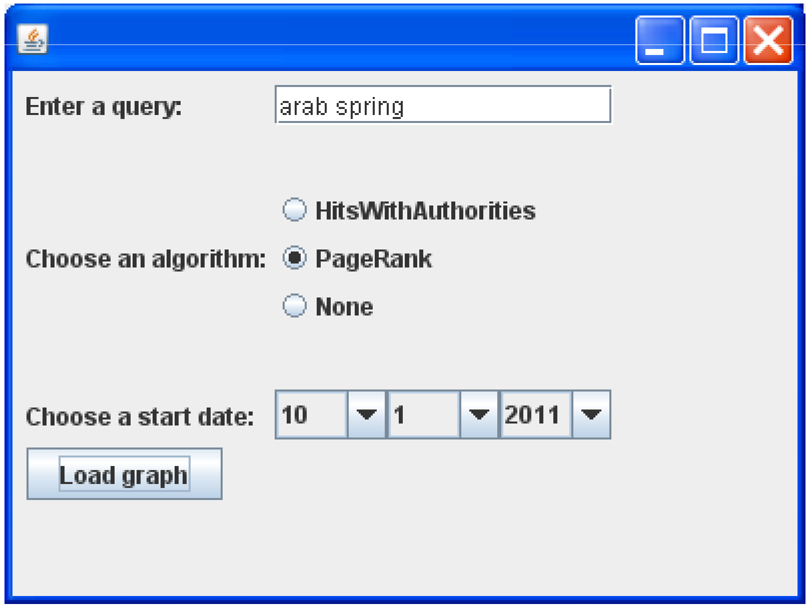
- Very good results by SVM on top of BM25
- BM25 of top terms can be aggregated in memory
- SVM training is “expensive”but …
- SVM learning just needs the support vectors

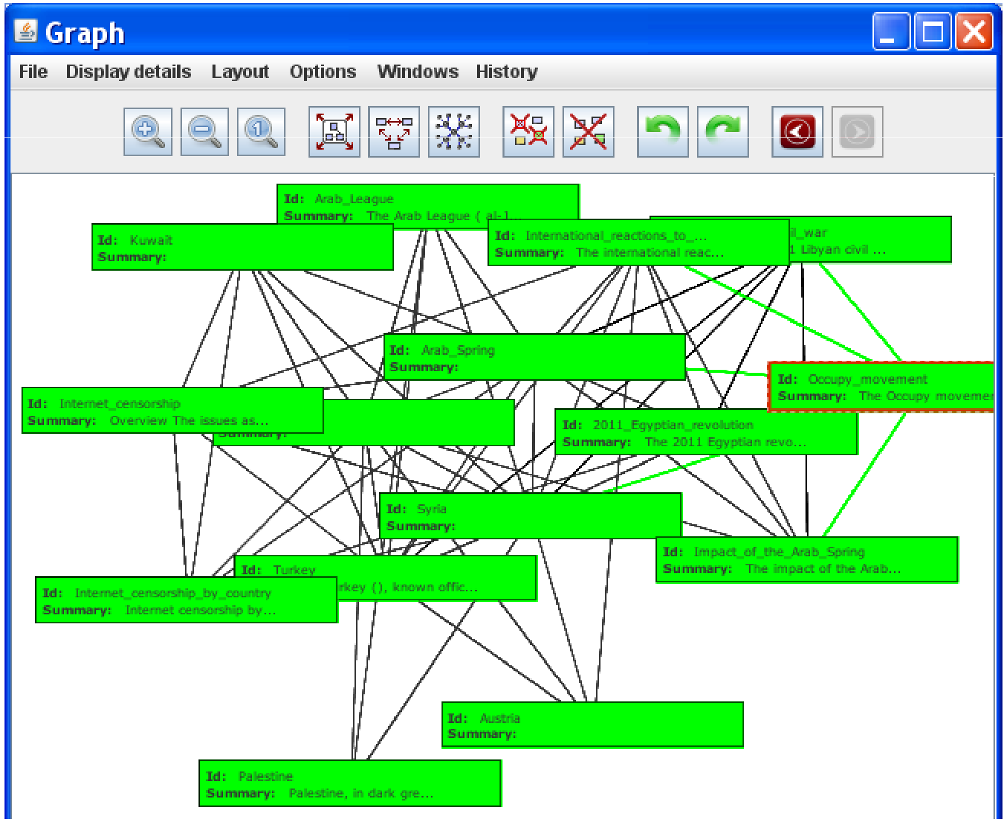
- Classification result is immediately available once sufficient number of sample pages (~100) crawled

Feature set	Spam	Genre	Quality	Avg
Public link based	0.655	0.614	0.519	0.587
Local content based	0.726	0.662	0.558	0.634
Local content + PageRank	0.757	0.713	0.540	0.660
Public content based	0.799	0.735	0.512	0.668
BM25	**0.876**	**0.805**	**0.584**	**0.739**
Public link + content	0.812	0.731	0.518	0.669
BM25 + local content	0.872	**0.816**	0.580	**0.754**
BM25 + public content	**0.891**	0.810	**0.612**	0.744
All combined	0.885	0.813	0.553	0.734

Research on Wikipedia

- Wikipedia great virtue is being utterly up-to-date
- Significant events usually have an immediate trace
- Chain of events – causes and effects – represented by links
- Find evolving stories by information on appearance of pages and links



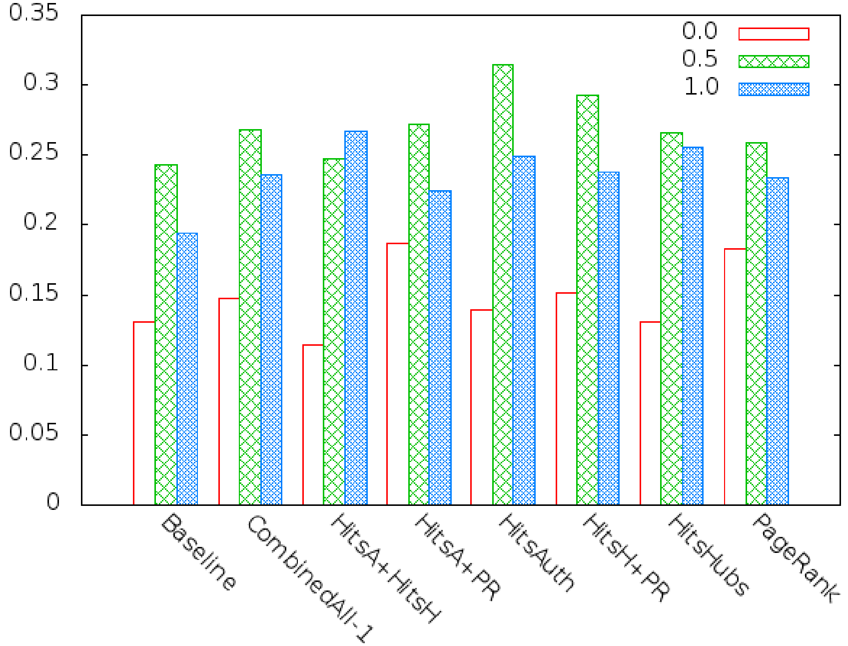


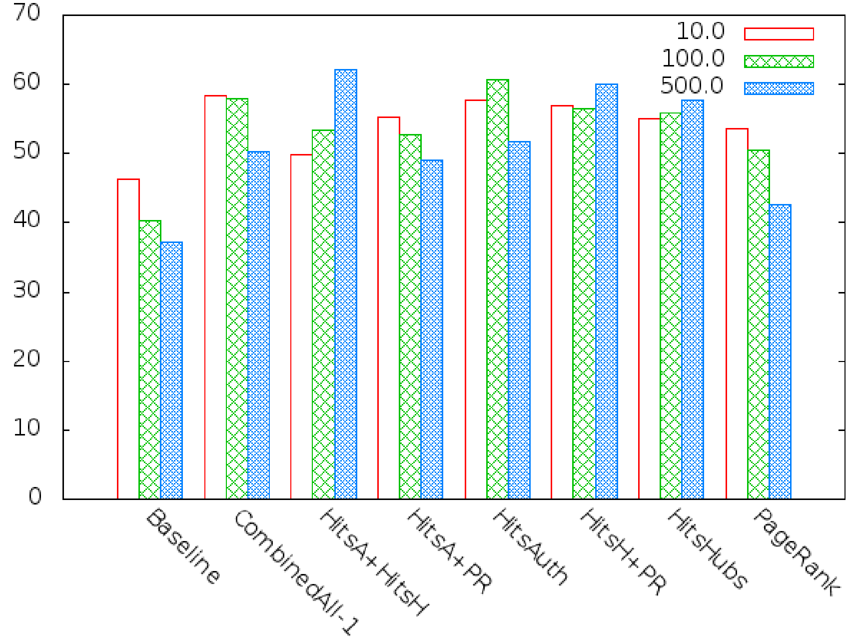
Measures of change

- Difference in #words
- Log of in and out degree
- Neighborhood
  - Search results form seed
  - Extend along changing edges&nodes
- Ranking
  - PageRank
  - HITS
  - Personalization on change and relevance
    - new method for HITS by supersources

		Sep ↓ Oct	Oct ↓ Nov
Muammar Gaddafi	content	0.044	0.18
	inlink	0.55	0.12
	outlink	0.033	0.04
	total	0.63	0.34
Death of Muammar Gaddafi	content	0	7.71
	inlink	0	4.21
	outlink	0	4.64
	total	0	16.6
Battle of Sirte (2011)	content	7.78	0.79
	inlink	4.78	0.21
	outlink	4.9	0.14
	total	17.5	1.1
National Transitional Council	content	0.15	0.08
	inlink	0.91	0.13
	outlink	5.68	0.29
	total	6.7	0.5

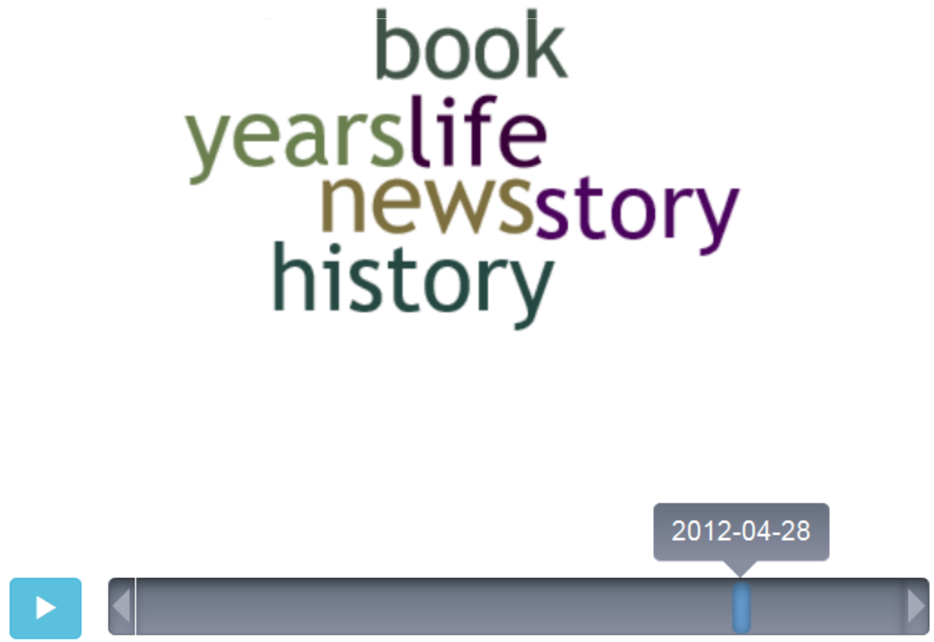
Experiments





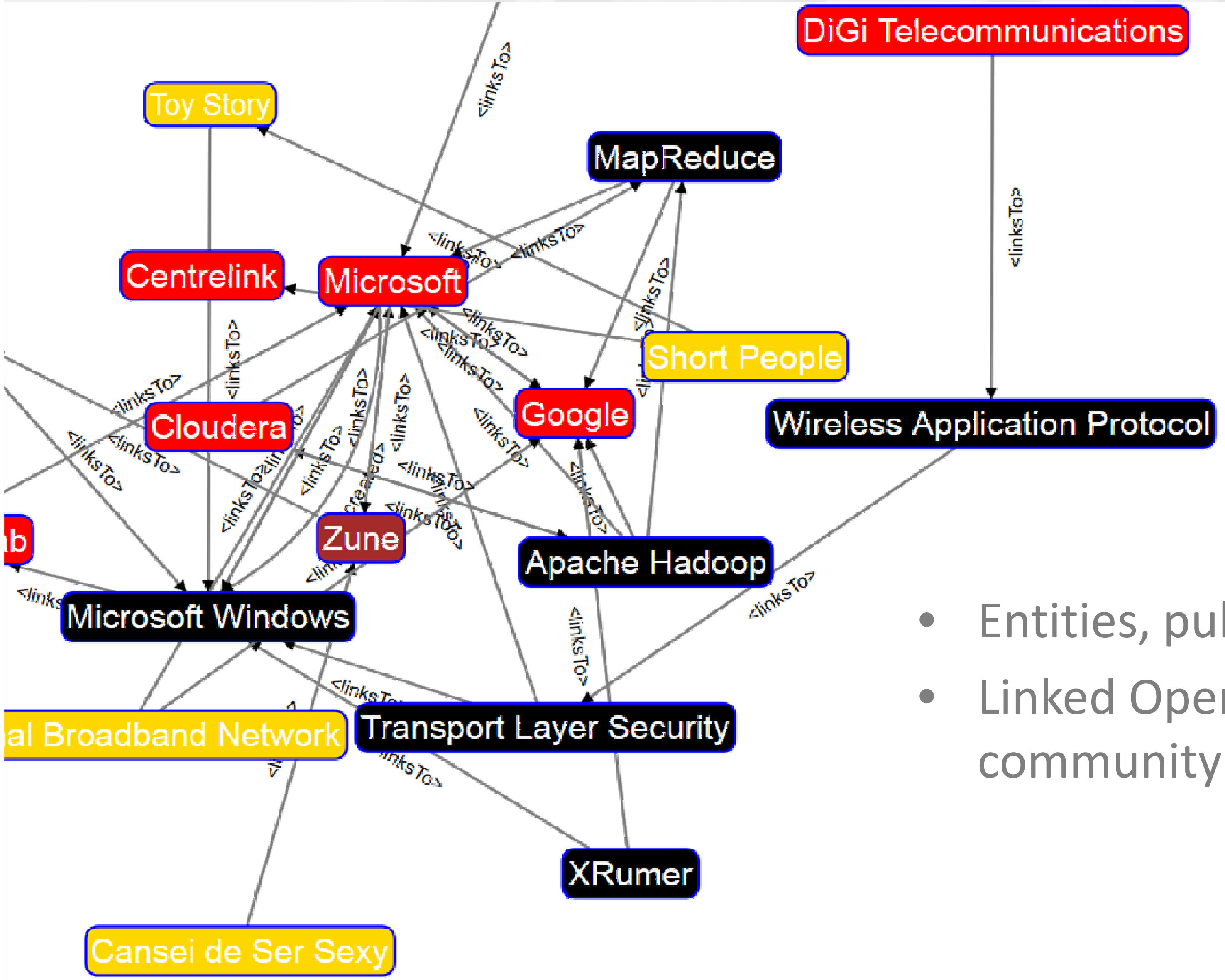
- NDCG @ 15 for best seed
- Increase in # edges in top 15
    - Best seed (100) and expansion (1000) sizes
    - Combination between relevance only (0), change (1) and avg (.5)

Trend detection



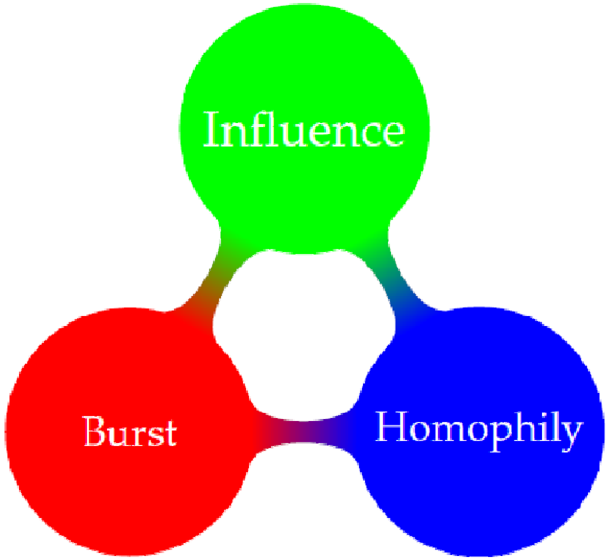
- So far, work on algorithmic challenges only
- Millions of relevant docs
- Real time user app
- Approximate data structures for counting

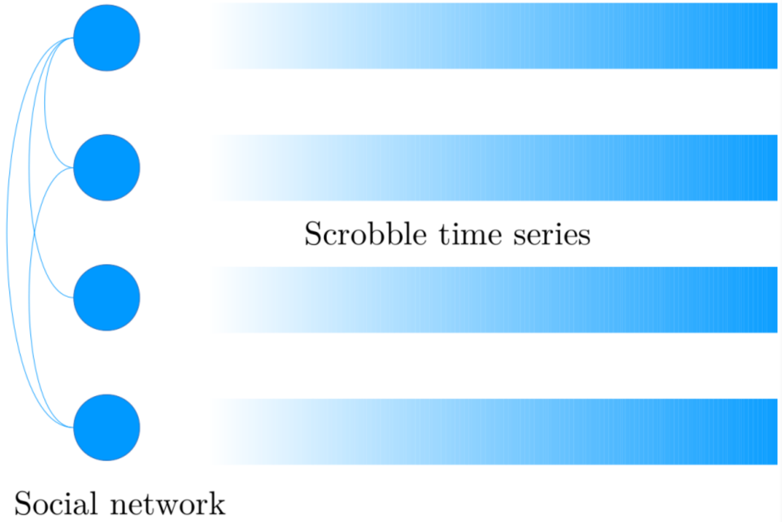
Plans with subgraph ranking

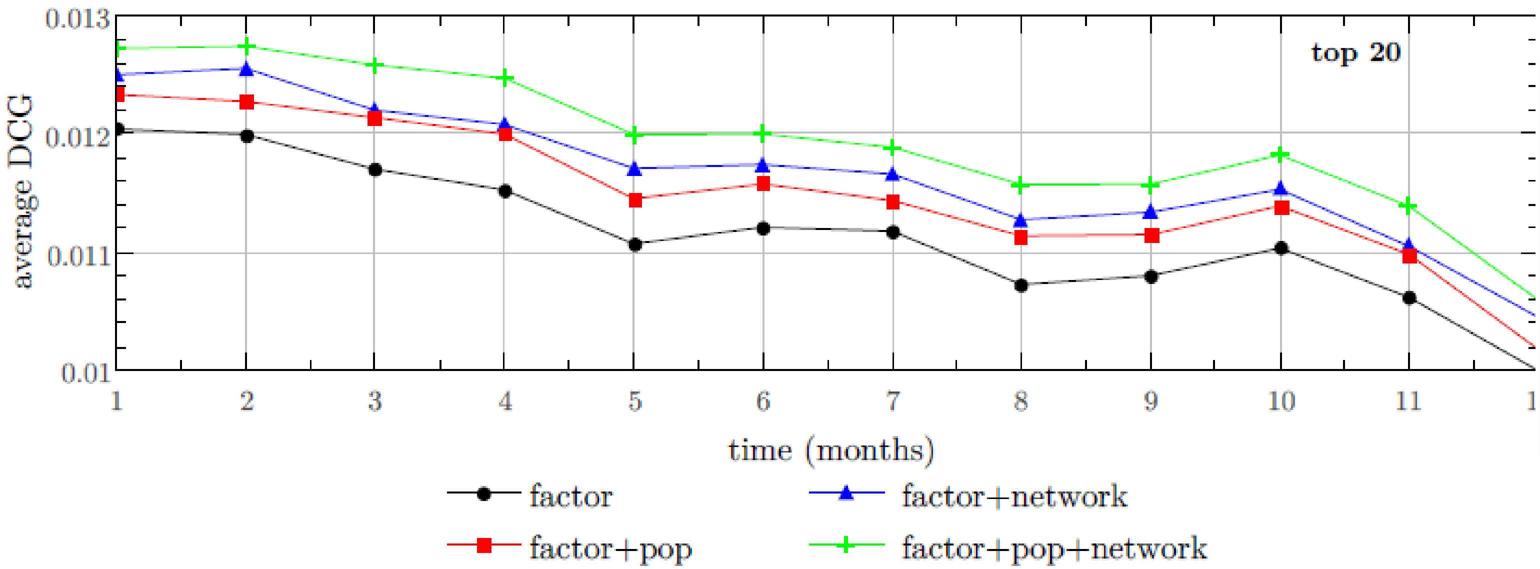


- Entities, publications, …
- Linked Open Data community

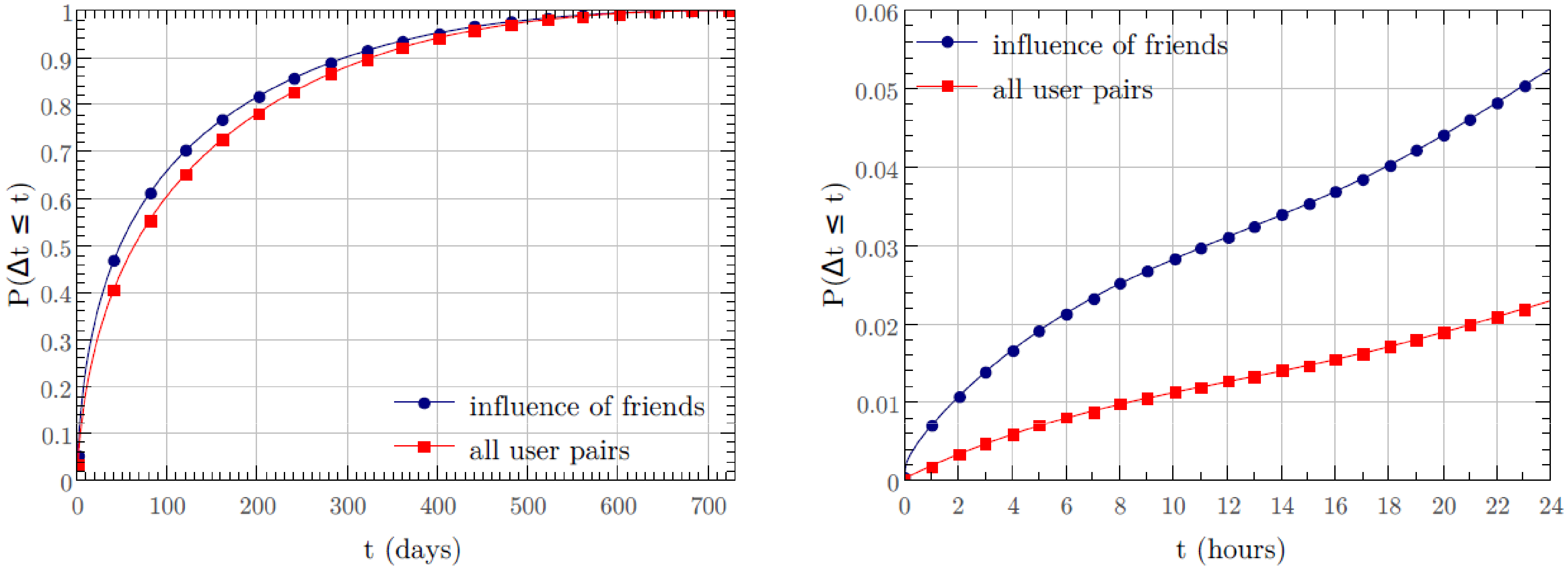
Network Influence in Recommenders





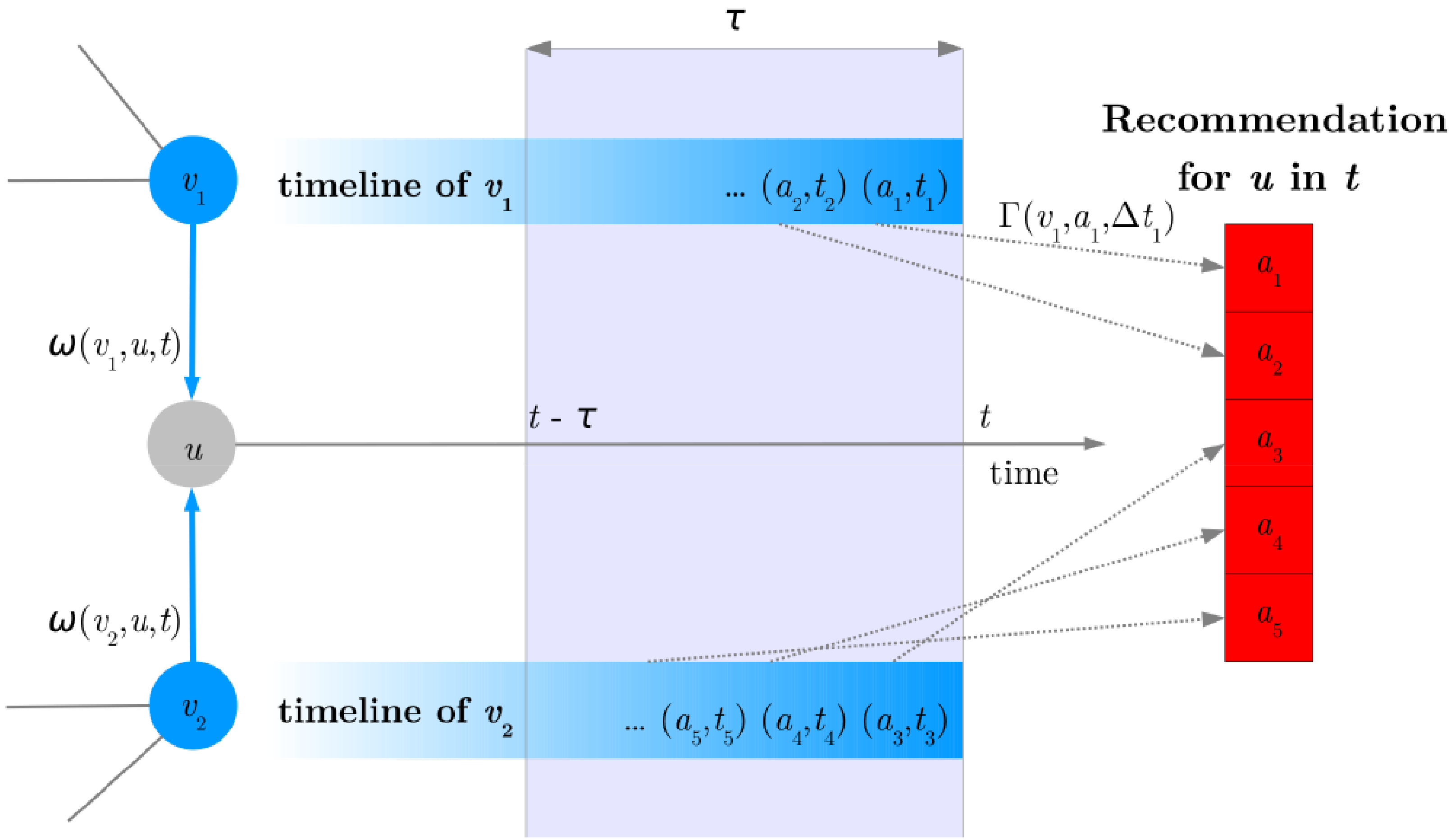


Observed influence



- User influenced if scrobbles new artist first time after a friend
- Delay is time elapsed after friend's last scrobble
- Baseline: random users scrobbling by coincidence before a first time scrobble

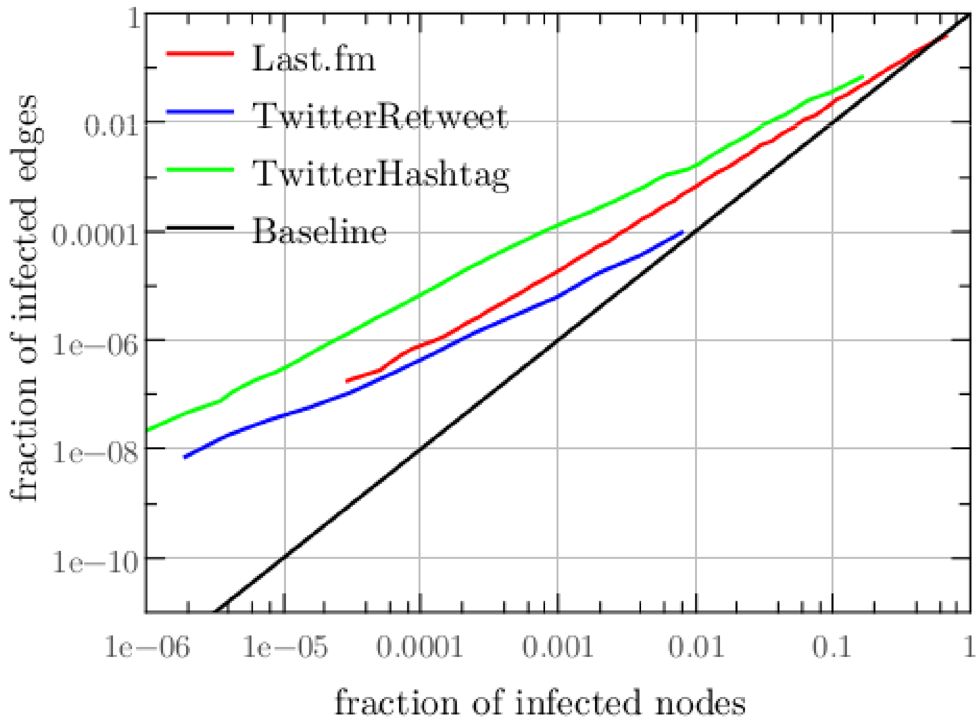
Influence recommendation



$$\hat{r}(u,a,t) = \sum_{v \in n(u)} \Gamma(\Delta t(v,u,a))\omega(v,u,t)$$

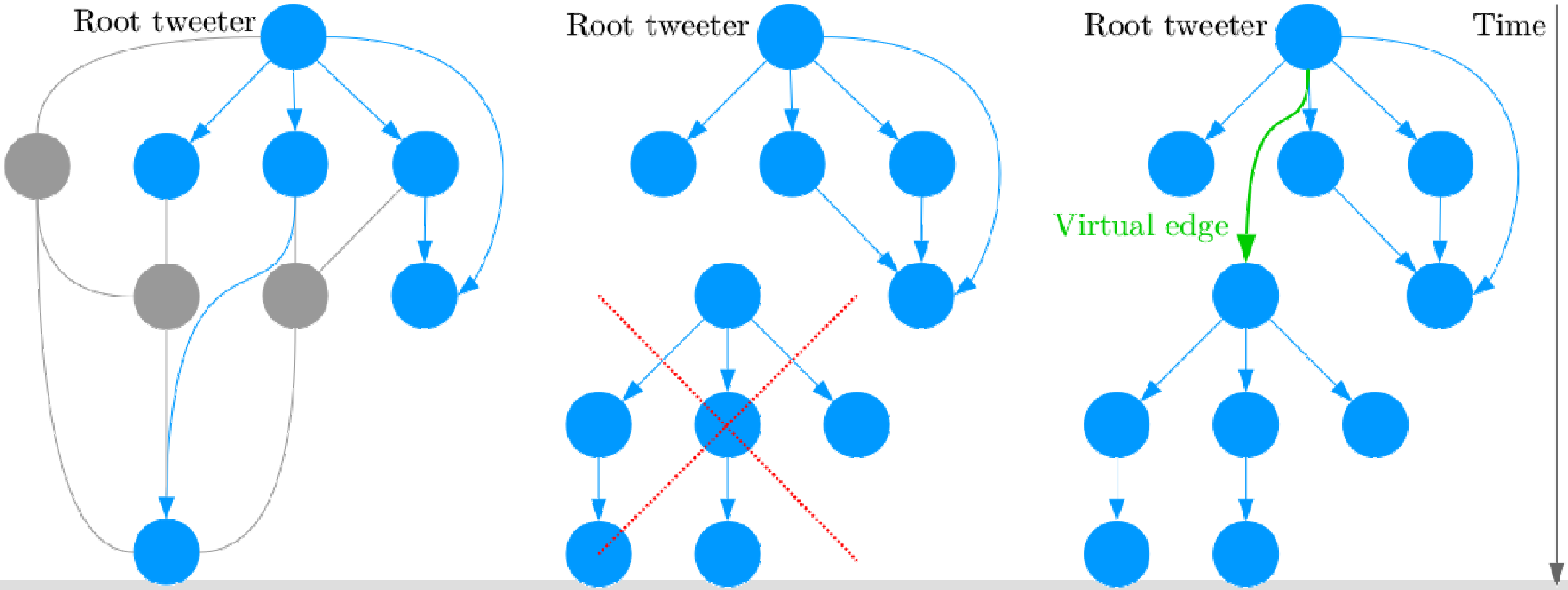
Densification law (under progress)

- Number of edges in spanned subgraph for users who scrobbled a given artist
- Small communities have larger edge density than random
- Looking for models, explanation
  - Several data sets
  - One model adds edge proportional to friends' **earliest adoption time**

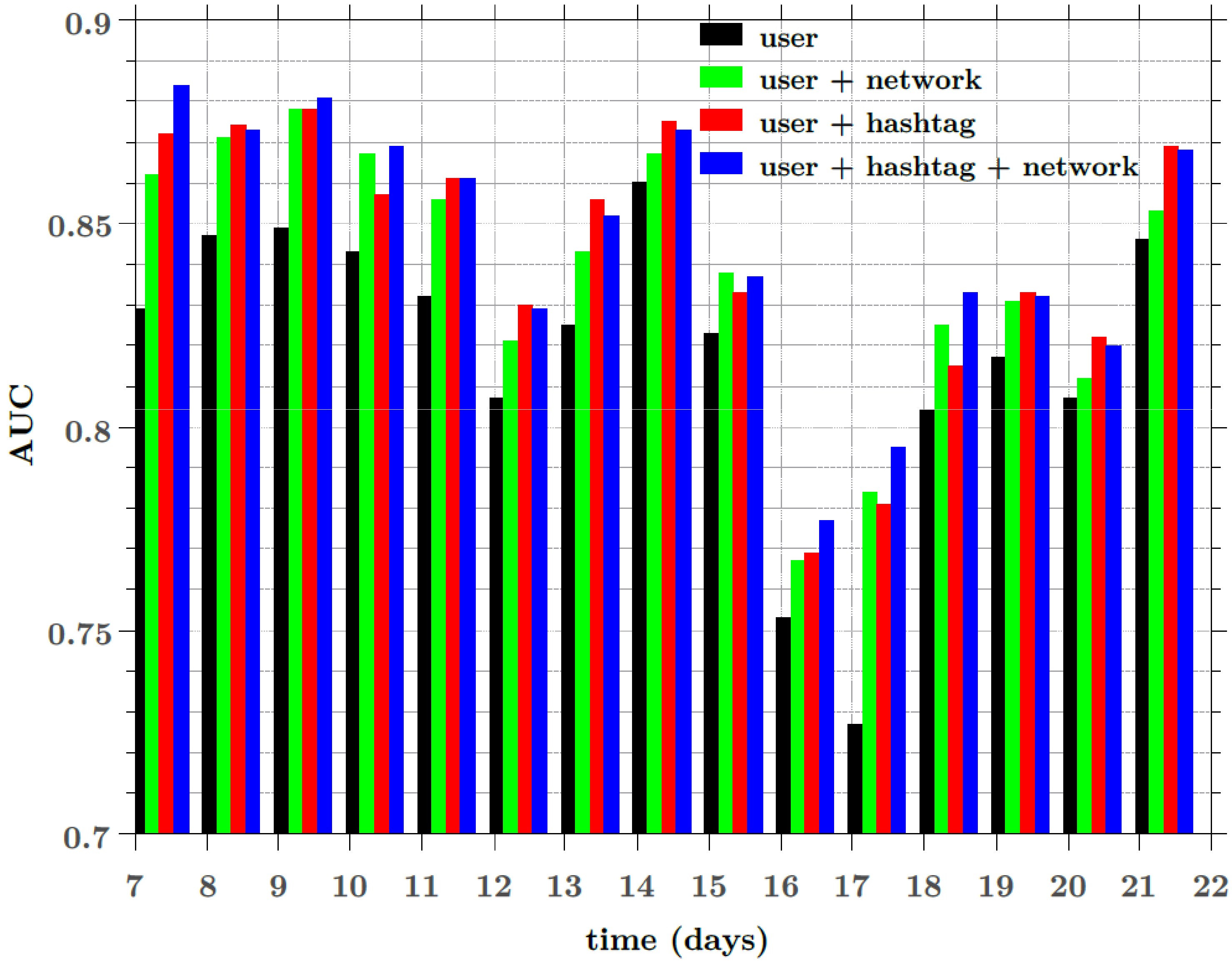


Apply for Twitter: retweets

- Twitter four topic crawl ("10o","occupy","20n","yosoy132").
  - Obtained by Andreas Kaltenbrunner
  - Follower network: 10^6 users; Tweets: ~ 10^5 - 10^6 per topic
- We crawled the social network (who follows who)
- Needed since we only know the ROOT of a retweet sequence
- Approximate only

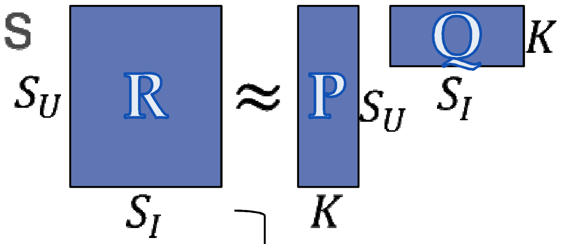


Prediction for retweet cascade size



The Matrix Factorization recommender

- Model
  - How we approximate user preferences
  - $\hat{r}_{u,i} = {p_u}^T q_i$



- Objective function (error function)
  - What we want to minimize or optimize?
  - E.g. optimize for RMSE with regularization

$$\mathrm{L} = \sum_{(u,i)\in Train}\left(\hat{r}_{u,i} - r_{u,i}\right)^2 + \lambda_U \sum_{u=1}^{S_U} \|P_u\|^2 + \lambda_I \sum_{i=1}^{S_I} \|Q_i\|^2$$

Learning

- Learning method
  - How we improve the objective function?
  - E.g. stochastic gradient descent (SGD)

Source of next slides:
Domonkos Tikk, CEO, Gravity

BRISMF model

- Biased Regularized Incremental Simultaneous Matrix Factorization
- Apply regularization to prevent overfitting
- To further decrease RMSE using bias values
- Model:

$$\hat{r}_{ui} = \vec{p}_u \vec{q}_i + b_u + c_i = \sum_{k=1}^{K} p_{uk} q_{ki} + b_u + c_i$$

BRISMF Learning

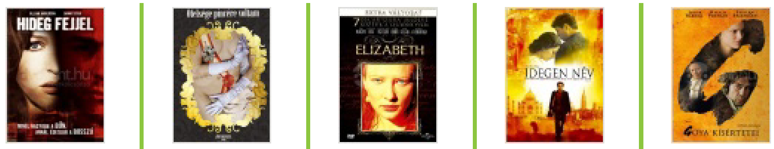
- Loss function

$$\sum_{(u,i)\in R_{train}} \left(r_{ui} - \sum_{k=1}^{K} p_{uk} q_{ki} - b_u - c_i \right)^2 + \lambda \sum_{(u,k)} p_{uk}^2 + \lambda \sum_{(i,k)} q_{ki}^2 + \lambda \sum_{u} b_u^2 + \lambda \sum_{i} c_i^2$$

- SGD update rules

$$\Delta p_{uk} = \eta\left(e_{ui} q_{ki} - \lambda p_{uk}\right) \quad \Delta q_{ki} = \eta\left(e_{ui} p_{uk} - \lambda q_{ki}\right)$$

$$\Delta b_u = \eta\left(e_{ui} - \lambda b_u\right) \quad \Delta c_i = \eta\left(e_{ui} - \lambda c_i\right)$$



R

	1	4		3	
			4	4	
	4		2		4

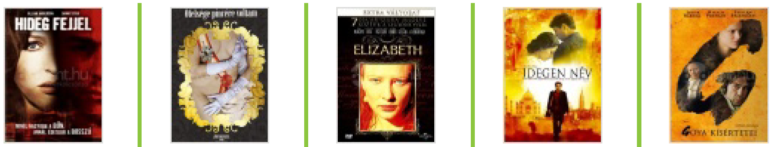
P

1,2	-0,4
1,2	0,9
0,5	-0,4

Q

1,5	0,9	-1,3	-0,1	0.6
-0,2	0,5	-0,4	1,6	0,3

Spam, Ranking and Recommenders



R

1	4	3.3	3	2.4
-0.5	3.5	4	4	1.5
4	4.9	2	1.1	4

P

1,4	1,1
0,9	1,9
2,5	-0,3

Q

1,5	2,1	1,0	0.7	1.6
-1,0	0,8	1,6	1,8	0,0

Influence Learning by Gradient Descent

- Present influence recommender:
  - heuristic weighted network learning
  - no artist based learning part
- Heuristic combination of the influence and factor models
  - Is it likely that user v influences user u on artist a?
  - Can user a be influenced at all in case of artist a?
- Use SGD method to learn user and artist factors

$$\hat{r}_{uat} = \sum_{v} \Gamma(\Delta t)(\vec{p}_v \vec{q}_a + b_v + c_i)$$

Conclusions

- Web classification plans to integrate with BUbiNG, use SZTAKI cluster to test the crawler
- Temporal ranking in Wikipedia, Twitter – trends, changes, events
- Ranking for subgraph selection, new applications
- Twitter
  - Understand the 1TBdata
  - Find influences in the user graph that we collect for Andreas' data
- Distributed machine learning and graph algorithms

Data sets and test bed

- Web classification
  - ClueWeb
  - Portuguese archive
  - Source codes released
- Twitter
  - Topical collection around four hashtags (Andreas Kaltenbrunner)
  - 1+Bio firehose
- The SZTAKI Text Mining Center

  http://info.ilab.sztaki.hu/vwo/second/vwo

Plans for Period II

- Research on crawler and classification integration strategies
- Modeling information diffusion and community densification
- Applying network models in recommender systems
  (e.g. geolocation, see Robert’s talk in the afternoon)