Estimating Peer Similarity using Distance of Shared Files

Yuval Shavitt, Ela Weinsberg, **Udi Weinsberg**

Tel-Aviv University

Problem Setting

- Peer-to-Peer (p2p) networks are used by millions for sharing content
- Increasingly difficult to find useful content
  - Noise in user generated content (meta-data)
  - Extreme dimensions
  - Sparseness

Work Goal

- Suggest a new metric for peer similarity
  - Overcome the **sparseness** problem
- Improve ability to find content
  - Search algorithms
    - Similar peers are likely to hold relevant content
  - Collaborative filtering
    - Find “like-minded” peers

Key Concept

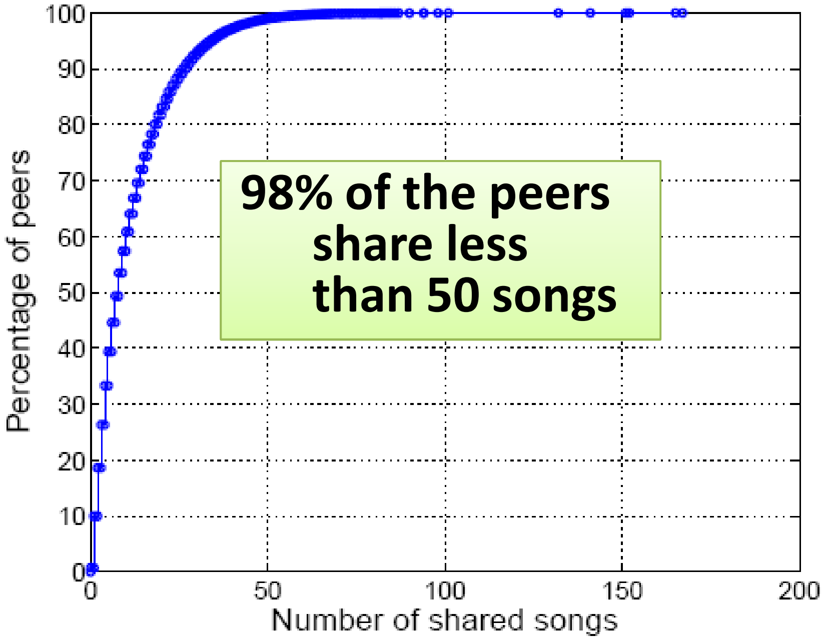
- Build a file similarity graph
  - Use data about all shared files
  - Weights of edges = distance between files
- Peer similarity is calculated using the distance between their shared files
  - No need for overlapping content between peers

Dataset

- Active crawl of Gnutella in 2007
- Crawled 1.2 million peers
- Only 35% of songs contain meta-data
- 530k distinct songs
  - Identified using “title|artist”
  - Accounting for spelling mistakes with edit distance

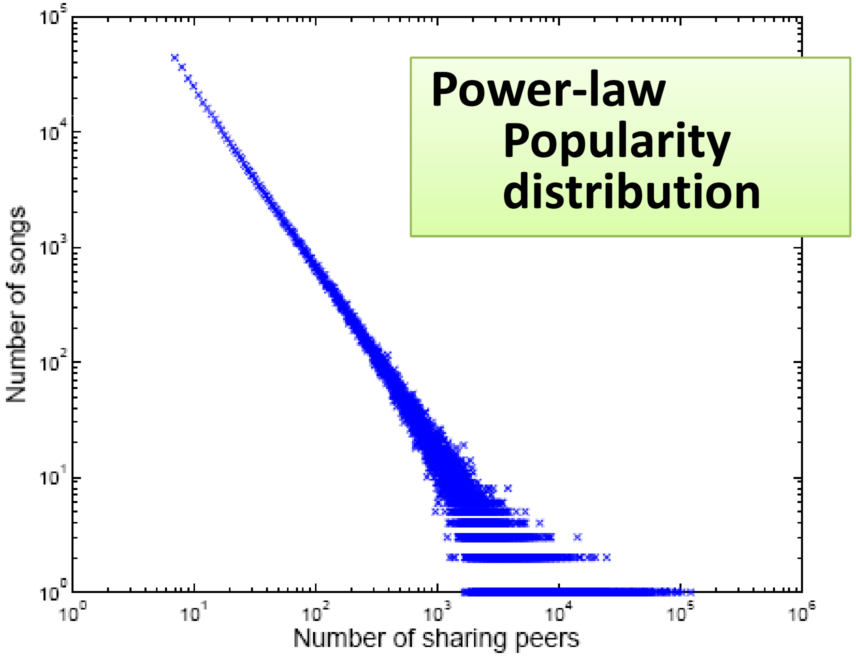
Dataset Statistics

- Using a sample of 100k peers (<10%)
- Over 511k songs remain (96%)



98% of the peers share less than 50 songs

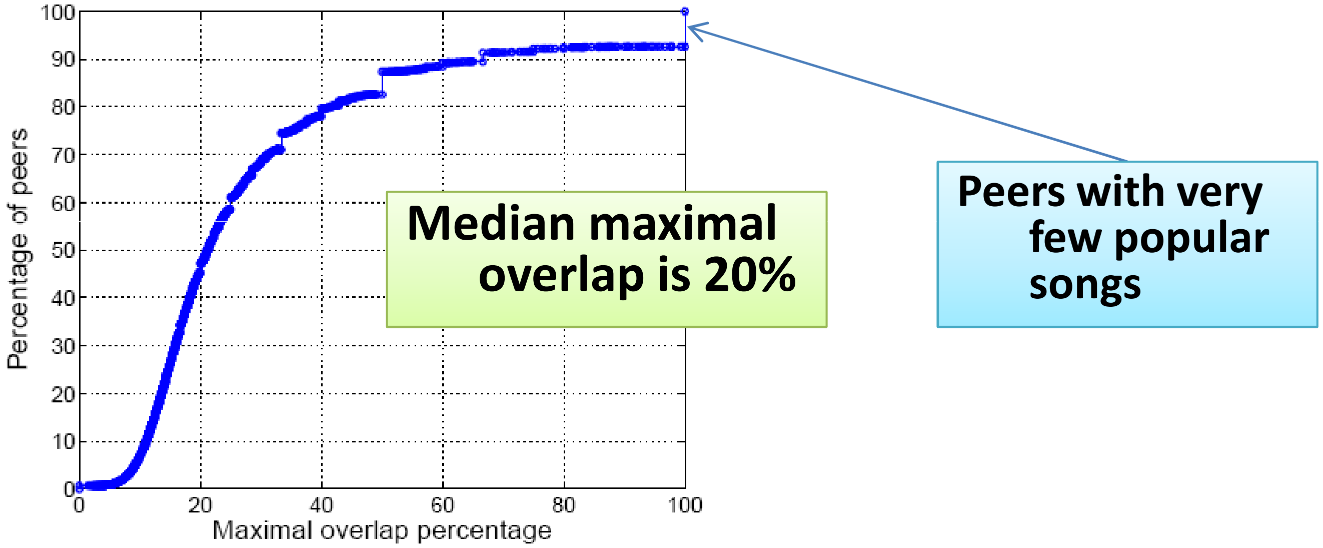
(a) Shared songs



Power-law Popularity distribution

(b) Popularity

Sparseness Problem



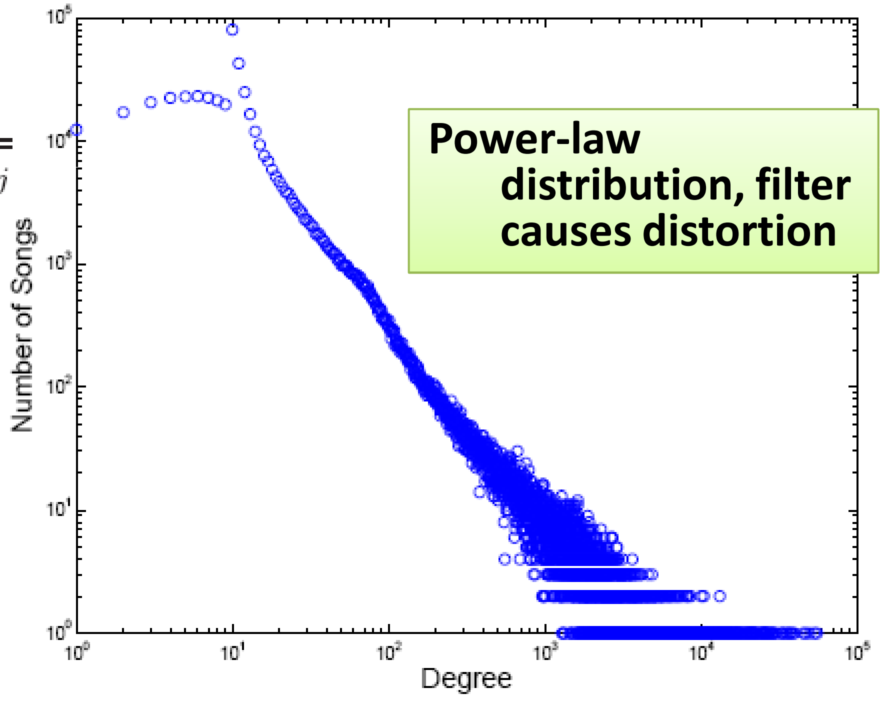
Median maximal overlap is 20%

Peers with very few popular songs

(b) Max overlap

File Similarity Graph

- Files are vertices
- Link weight is the number of peers sharing both
- Normalize similarity with popularity: $\widehat{w}_{ij} = \frac{w_{ij}}{\sqrt{C_i \cdot C_j}}$
- Filter
  - Keep only top 40%
  - And no less than 10



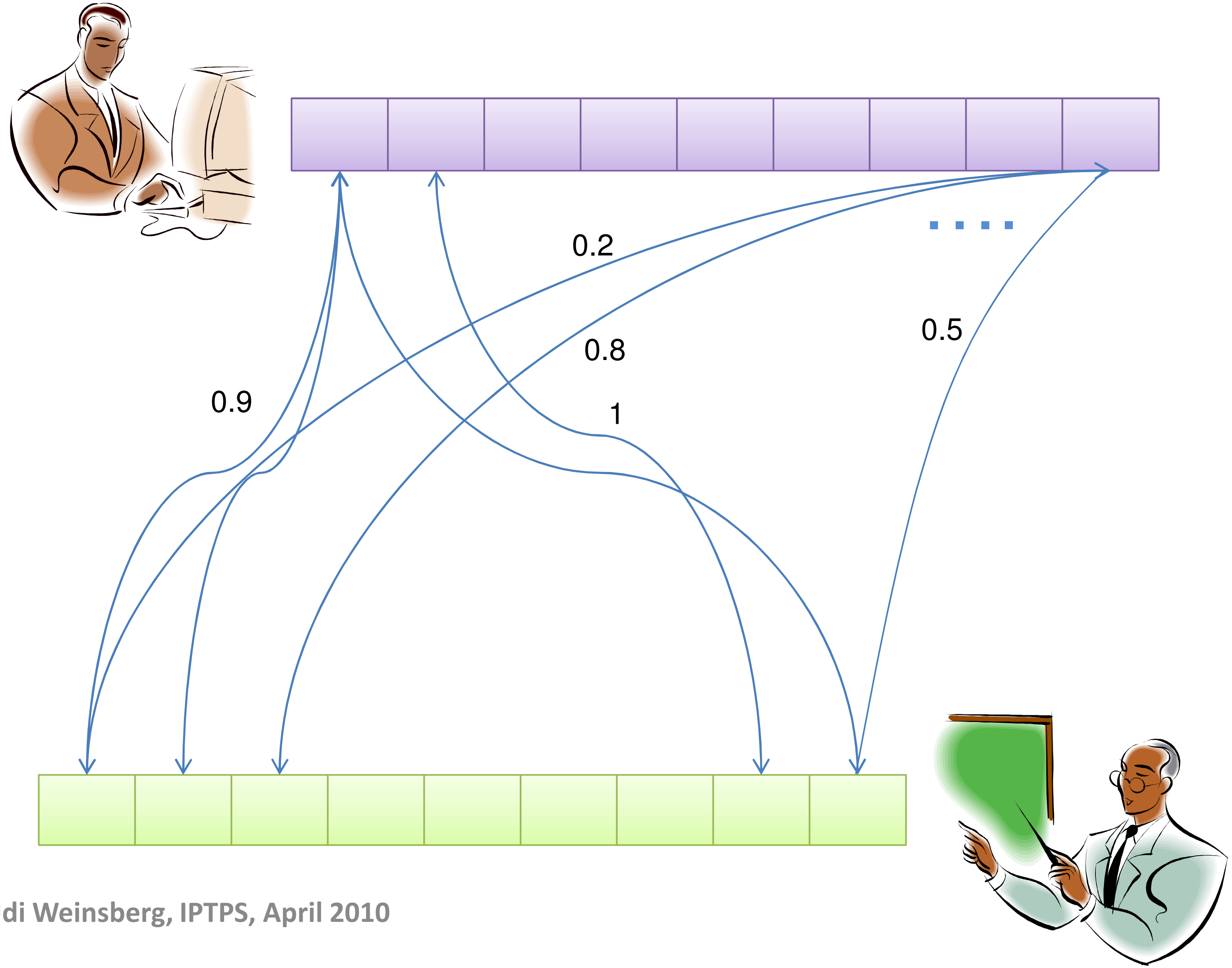
Power-law distribution, filter causes distortion

(a) Degree distribution

Peer Similarity Estimation (1)

- Create a bi-partite graph connecting the files of every two peers
- Connect files in the two sides with links:
  - If exact same file – weight is 1
  - Otherwise – use normalized similarity along the shortest path between the files

Distance Estimation

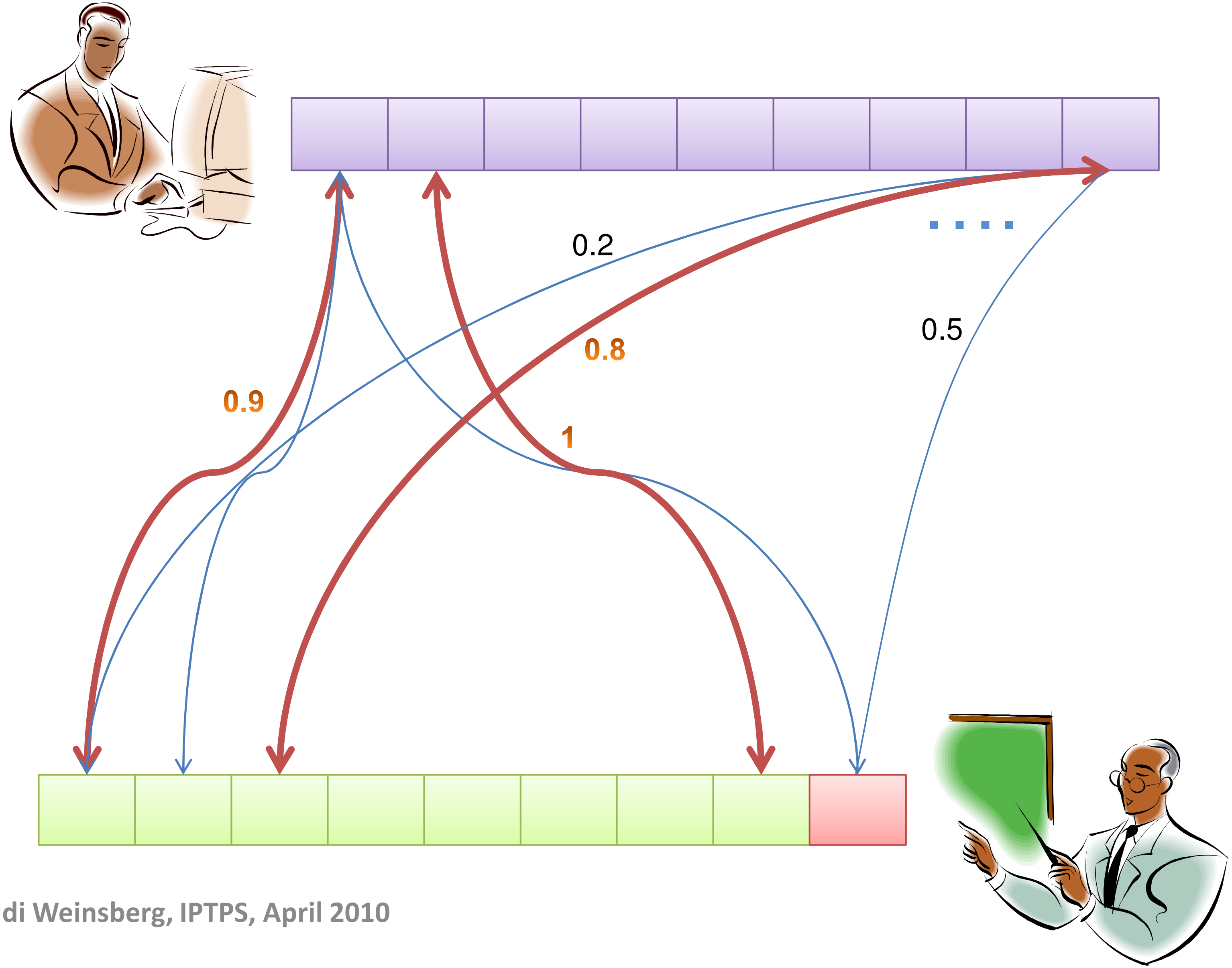


Peer Similarity Estimation (2)

- Run *maximal weighted matching* on the bi-partite
  - Find the “best” matching links between files
  - The matching *M* is the sum of links weight
- Peer similarity

$$P(p_i, p_j) = \frac{M}{\min\{|p_i|, |p_j|\}}$$

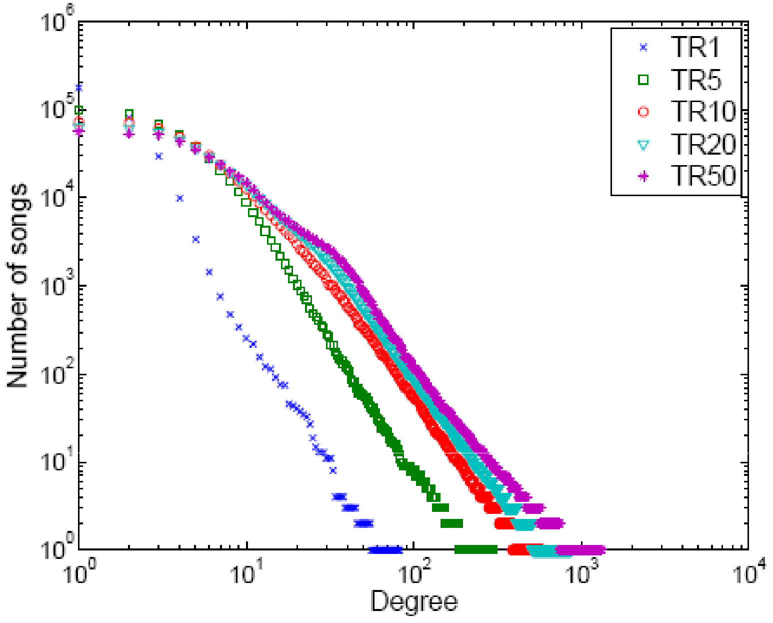
Maximal Weighted Matching



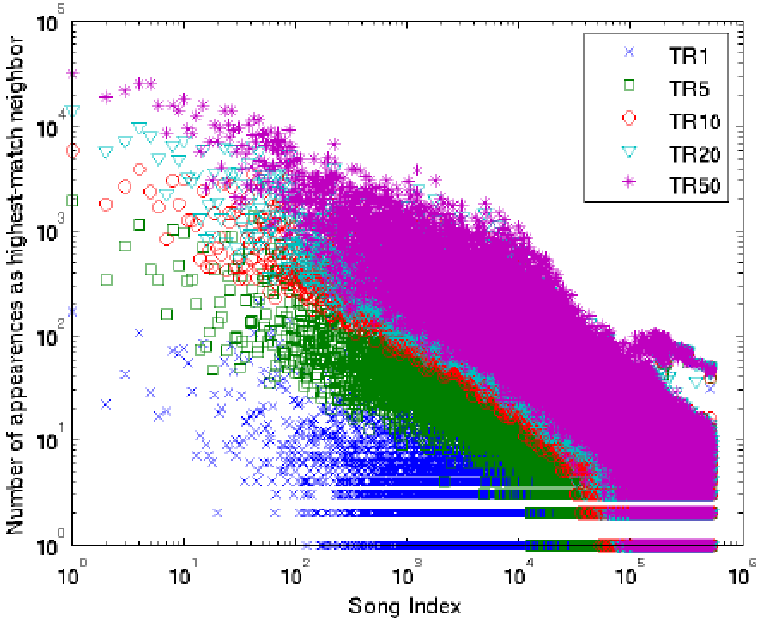
Distance Estimation Issues

- File similarity graph can have connected components
  - Some distances are infinite
- All pairs shortest paths can be costly
  - Reduce the size of the similarity graph
  - Limit the search depth

Reducing Similarity Graph Size



(b) Degree distribution



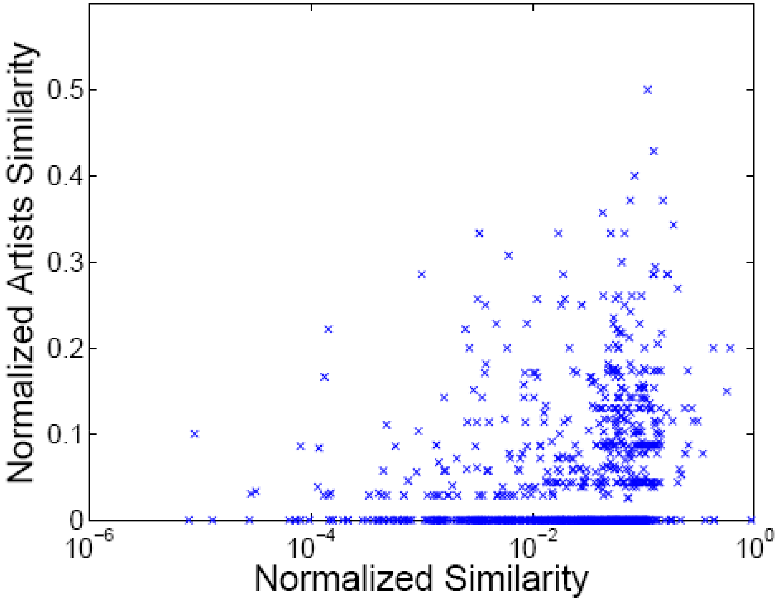
(c) Similarity sub-networks

- For each file, take only the top N nearest neighboring files
- Distribution almost overlap for N≥10

Limit Search Depth

- Stop searching files once reached *K* times the distance of the first finding
  - Distance between files become asymmetric
  - Depends on the peer we start from
- For *K*≥1.5 links removed are unlikely to be selected in the maximum matching
  - Asymmetric links are mostly low-similarity links
  - Hence will not be selected in the matching

Meta-data and Similarity



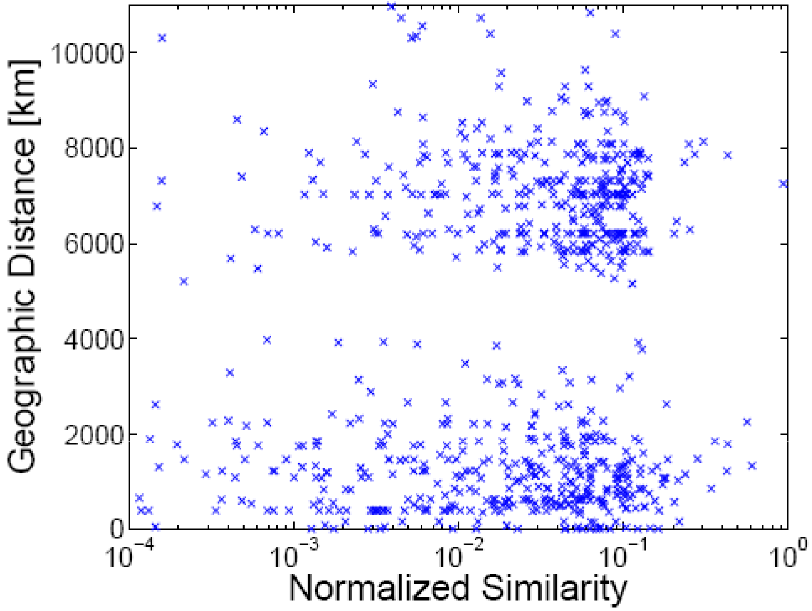
(a) Artists

- Similarity between peers *i* and *j* using artists

$$(|A_i \cap A_j|) / \min\{|A_i|, |A_j|\}$$

- Normalized similarity matches meta-data

Geography and Similarity



(b) Geography

- Comparing the distance with similarity
- No direct correlation!

Conclusions

- A metric for similarity between peers
- Evaluation using song files shared in Gnutella
  - Metric reflects the similarity of peer preferences in music
- Geography is not necessarily a good indication for peer similarity!

Thank You!

Udi Weinsberg
udiw@eng.tau.ac.il