DAI-Labor
TU Berlin

Alternative Similarity Functions for Graph Kernels

Jérôme Kunegis, Christian Bauckhage

DAI-Labor, Technische Universität Berlin

19th International Conference on Pattern Recognition

Tampa, December 2008

CC IRML
Information Retrieval
& Machine Learning

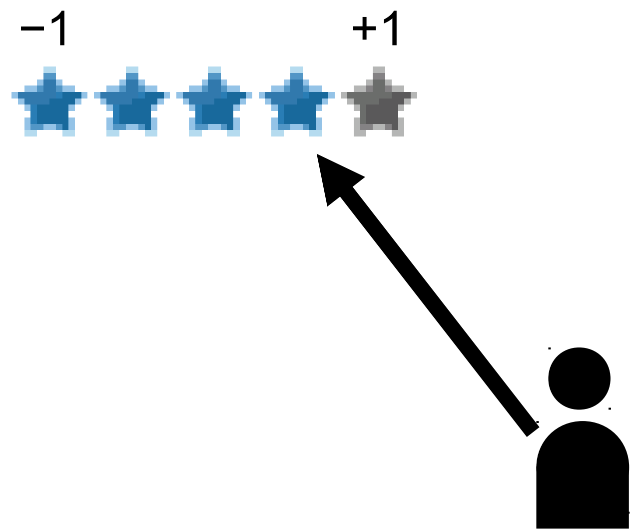
Outline

1. Collaborative Filtering
2. Graph Kernels
3. Similarity Functions
4. Evaluation
5. Conclusion

Dipl.-Inform. Jérôme Kunegis I kunegis@dai-lab.de

1. Collaborative Filtering – Problem Formulation

Users rate items



- Task: Predict ratings

Examples: Amazon, Netflix, …

Users/ Items	U_1	U_2	U_3	U_4	U_5
I_1	+1	+1	+1	?	+1
I_2	−1	−1	?	−1	+1
I_3	?	+1	−1	+1	?

1. Collaborative Filtering – Algebraic Representation

•Represent ratings as a sparse matrix R

•Ratings are signed: positive and negative values

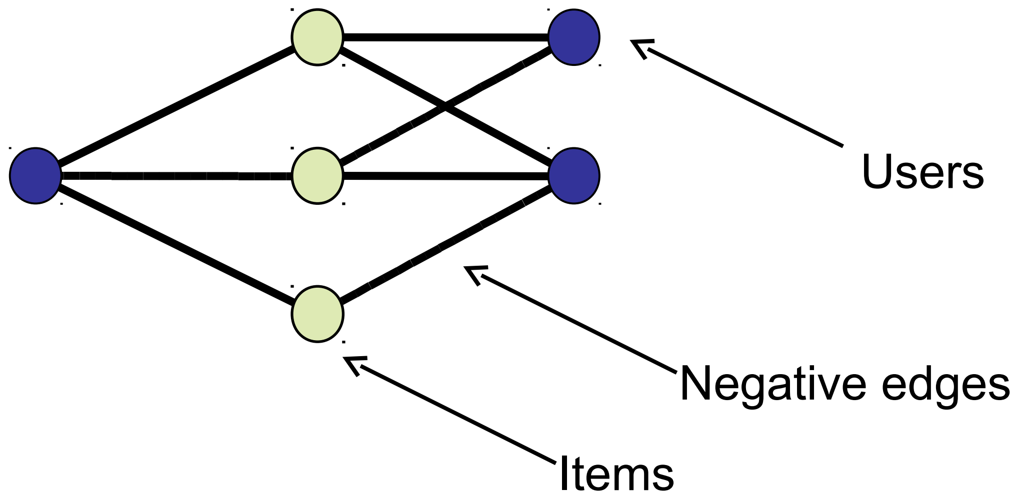
R =

Users/ Items	U_1	U_2	U_3	U_4	U_5
I_1	+1	+1	+1	?	+1
I_2	−1	−1	?	−1	+1
I_3	?	+1	−1	+1	?

Dipl.-Inform. Jérôme Kunegis I kunegis@dai-lab.de

1. Collaborative Filtering – Bipartite Rating Graph

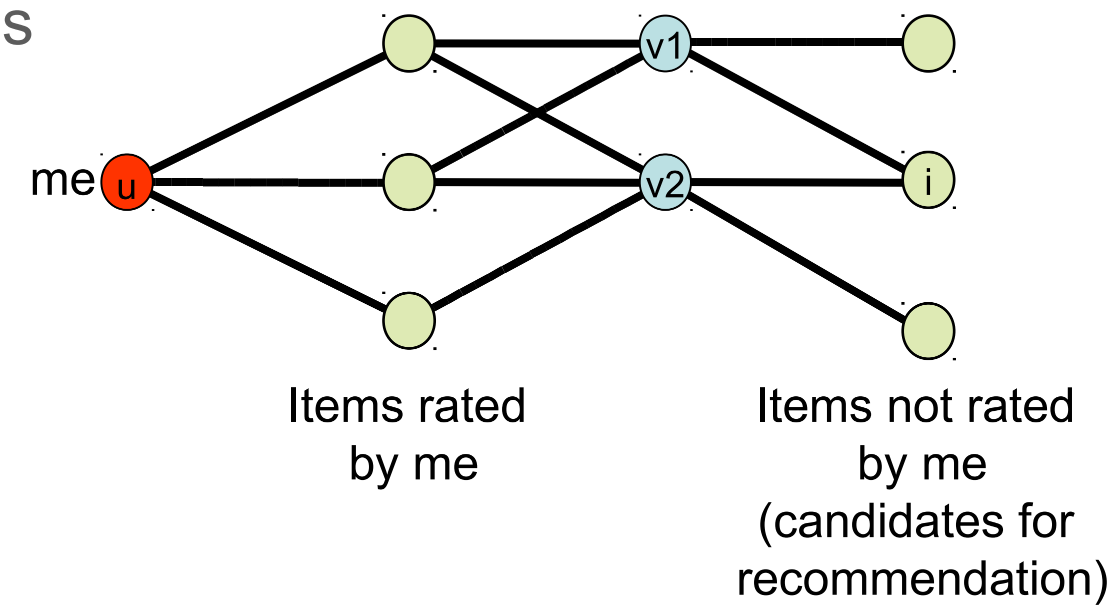
Ratings as a graph



- Rating graph is bipartite and undirected
- Edges are weighted by signed values
- Adjacency matrix A given by A = [0 R; R’ 0]
- Degree matrix D = diag(sum(A))

1. Collaborative Filtering – Prediction Algorithm

Recommend items



For rating prediction, compute weighted mean of known ratings by other users, weighted by user similarity:

$$P(u,i) = \left(\Sigma_v \, \mathrm{sim}(u,v)\right)^{-1} \Sigma_v \, \mathrm{sim}(u,v) \, R_{vi}$$

Dipl.-Inform. Jérôme Kunegis I kunegis@dai-lab.de

2. Graph Kernels – Definition

- Function sim(u,v) is mostly a **kernel:** symmetric and positive (semi-)definite
- Define graph kernels using the adjacency matrix A:

$K_{FOR} = (I + D - A)^{-1}$ (forest kernel)

$K_{EXP} = \exp(\alpha A) = \Sigma_{i=0}^{\infty} \alpha^i / i!\, A^i$ (exponential kernel)

$K_{NEU} = (I - \alpha A)^{-1} = \Sigma_{i=0}^{\infty} \alpha^i A^i$ (von Neumann kernel)

for $0 < \alpha < 1$

Other, similar variants exist.

3. Similarity Functions

- Graph kernels can be used as a similarity function

$$\mathrm{sim}(u,v) = K_{uv}$$

- To define a distance based on K, write K = UU' and define

$$d(u,v)^2 = (U_u - U_v)^2 = K_{ii} + K_{jj} - K_{ij} - K_{ji}$$

- Use this distance to define alternative similarity functions:

$\mathrm{sim}_{Eu} = \sigma^2 / d^2$ (inverted Euclidean distance)

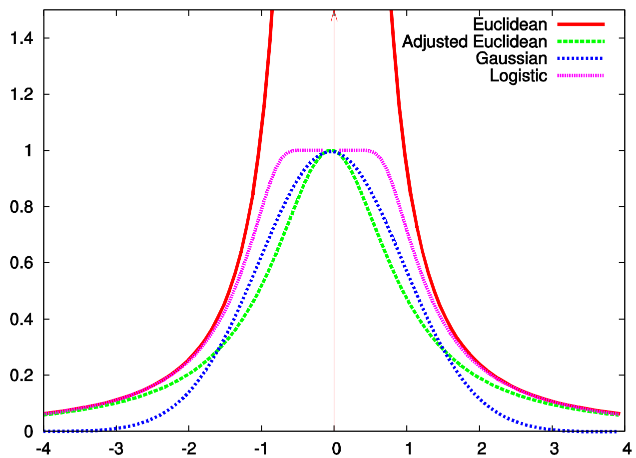
$\mathrm{sim}_{EuA} = 1 / (1 + d^2 / \sigma^2)$ (adjusted Euclidean distance)

$\mathrm{sim}_{Ga} = \exp(-½\, d^2 / \sigma^2)$ (Gaussian kernel)

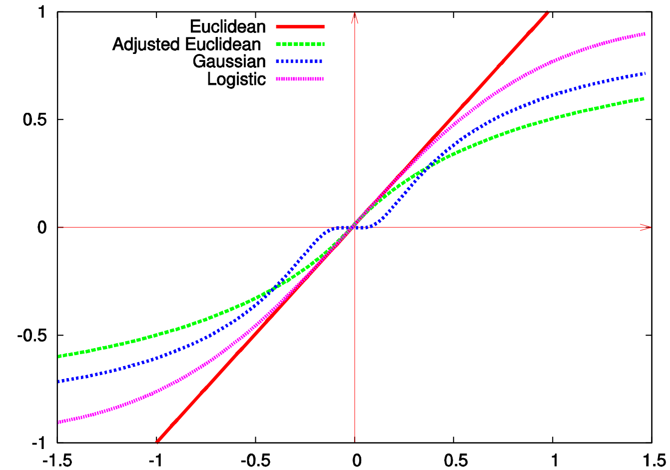
The parameter σ can be thought of as a scale parameter

Dipl.-Inform. Jérôme Kunegis I kunegis@dai-lab.de

3. Similarity Functions – Comparison



(a) d



(b) 1/d

Show the similarity function in function of *d* and *1/d*

- For comparison, show $\mathrm{sim}_{\mathrm{Sigmoid}} = \tanh(\sigma^2 / d^2)$
- Note: Gaussian function is double sigmoid in function of *1/d*
- Note: Unadjusted Euclidean gives unbounded weight for small distances

Dipl.-Inform. Jérôme Kunegis I kunegis@dai-lab.de

4. Evaluation

Use the Netflix Prize corpus of user-movie ratings

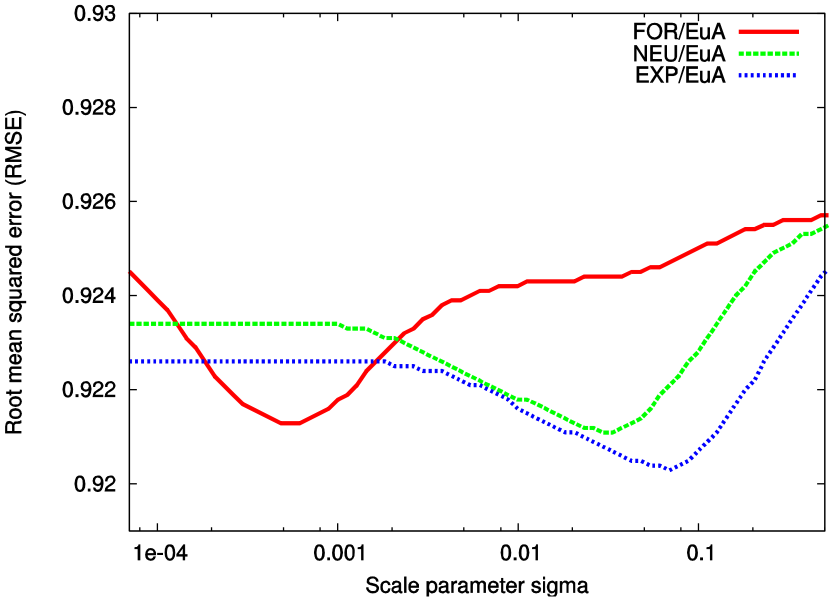
- Subset of 3,216 users,1,307 movies and 57,507 ratings

- Compute the Root mean squared error (RMSE)
- Cross-validation

Dipl.-Inform. Jérôme Kunegis I kunegis@dai-lab.de

4. Evaluation – (Adjusted) Euclidean

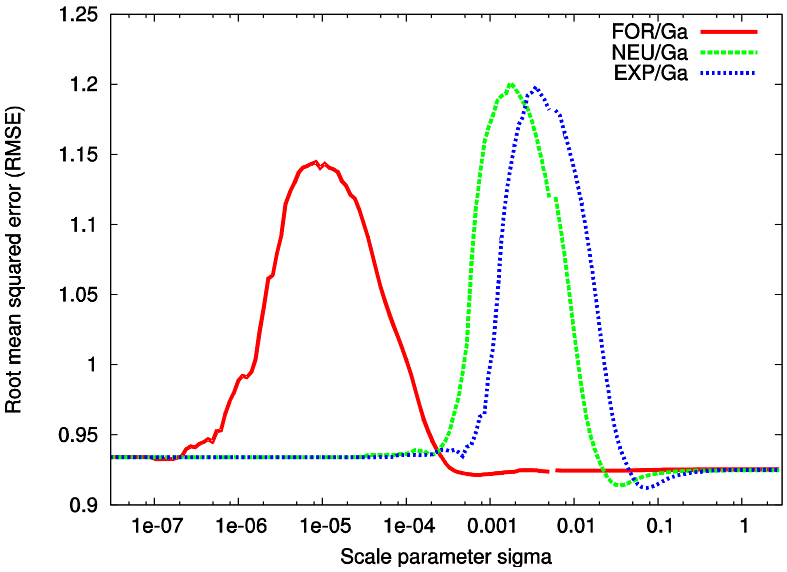
RMSE for adjusted Euclidean in function of σ for all three graph kernels.



- For small σ, corresponds to unadjusted Euclidean
- For large σ, corresponds to uniform weighting
- Best performance for specific values of σ
- Exponential kernel is best

4. Evaluation – Gaussian

RMSE for Gaussian similarity in function of σ for all three graph kernels.

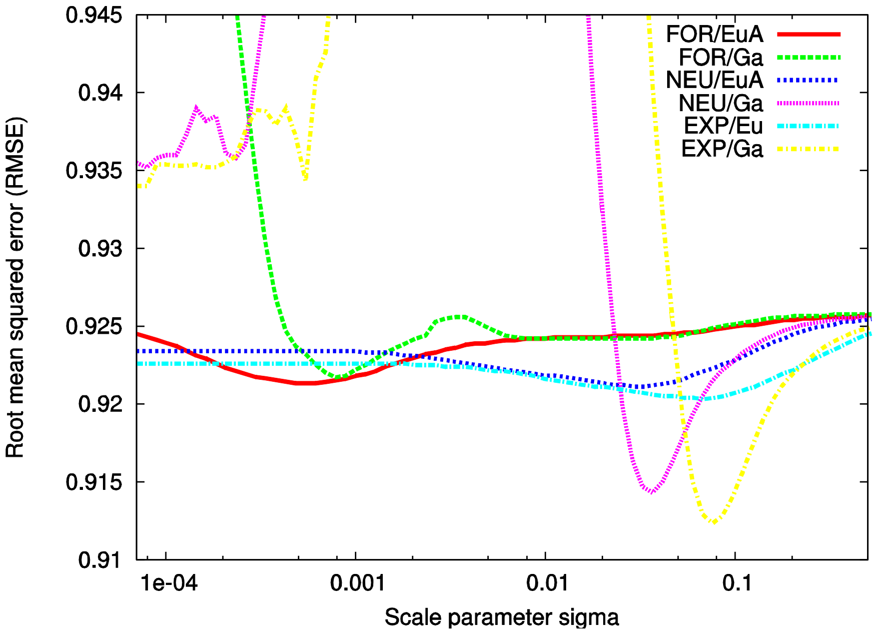


- For large σ, corresponds to uniform weighting

- Best performance for specific values of σ
- Exponential kernel is best
- Very bad performance for many values of σ

4. Evaluation – Overall Results

RMSE for all



- Best performance by Gaussian similarity when tuned properly
- Exponential kernel is best

5. Conclusion

Summary:

- Use the distance spanned by graph kernels
- Have to adjust σ for good results

Future work:

- Estimate σ in closed form
- Other recommendation scenarios
- Use directly (no weighted mean)

THANK YOU