Affiliation Recommendation using Auxiliary Networks

Vishvas Vasuki, Nagarajan Natarajan, Zhengdong Lu, Inderjit Dhillon

Dept. of Computer Science, The University of Texas at Austin

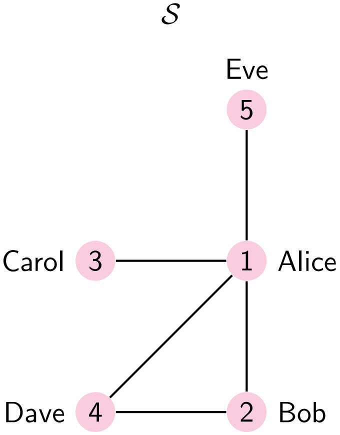
RecSys, 2010

Outline

- The problem
  - Introduce terminology, area.
  - What is the task?
- Modeling user-affiliation affinity.
  - Latent Factors Model
  - Graph Proximity Model
- Addressing scalability.
- Evaluation of the algorithms.
- Conclusions.

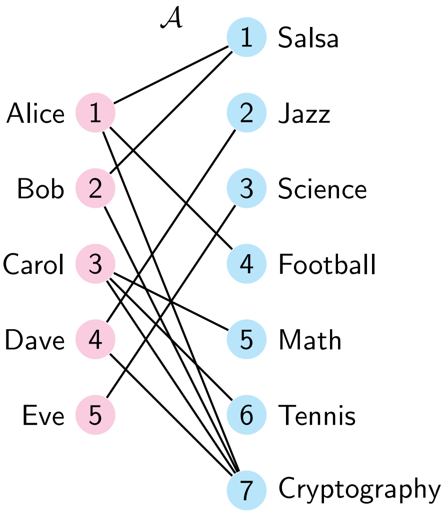
Social and Affiliation networks

Social network $\mathcal{S}$: An undirected graph.



- S: users $\times$ users .

Affiliation network $\mathcal{A}$: A bipartite graph.

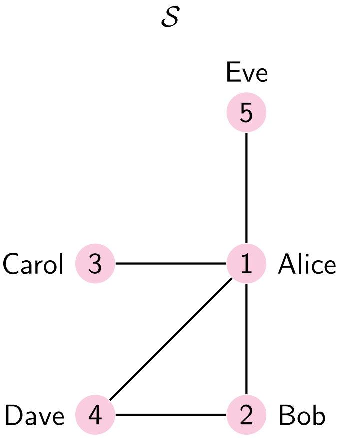


- A: users $\times$ groups/affiliations .

Affiliation networks

- Communities in social networks.
- Explicit / Implicit.
- Not necessarily among people — gene-disease network.

Social network analysis.



- Modelling network evolution.
- Link prediction.
- Community identification.

Our focus: Affiliation Recommendation.

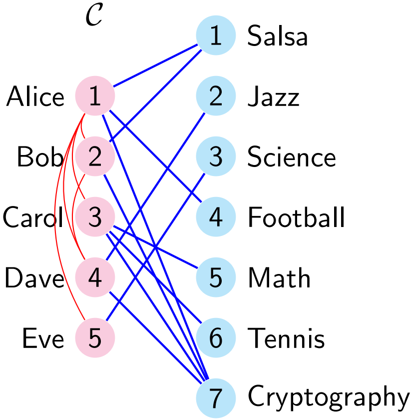
- Suggest communities to the users of a social network.
- Generalizable to the item recommendation problem.

- Can be thought of as link prediction in the affiliation network.
- Can we exploit auxiliary networks (like the friendship network)?

Modeling user-affiliation affinity

The combined network $\mathcal{C}$

$$\mathbf{C}(\lambda, \mathbf{D}) = \begin{bmatrix} \lambda S & A \\ A^T & \mathbf{D} \end{bmatrix}$$



- S: User-User adjacency.
- A: User-Affiliation adjacency.
- λ: relative weight associated with information in S.
- $\mathbf{D}$: unobserved (choices: $A^T A, \ldots$).

Latent factors model

Modeling $\mathcal{A}$ alone

- User-group affinity as product of low dimensional vectors: $A_{i,j} \approx \langle \mathbf{U}(i,:), \mathbf{G}(i,:) \rangle$.

$$A \approx \mathbf{U}\mathbf{G}^T$$

$$\text{rank}(\mathbf{U}) \leq k, \text{rank}(\mathbf{G}) \leq k$$

  U - User preferences; **G** - Affiliation characteristics.
- For user u, recommend affiliations with high affinity.

Modeling $\mathcal{C}$

- A **good model** will account for edges in S too.

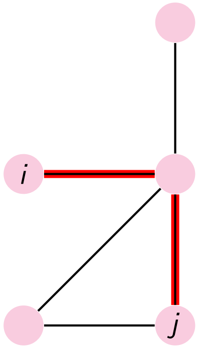
$$\mathbf{C}(\lambda, \mathbf{D}) = \begin{bmatrix} \lambda S & A \\ A^T & \mathbf{D} \end{bmatrix} \approx \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} \Lambda \begin{bmatrix} \mathbf{V}_1^T & \mathbf{V}_2^T \end{bmatrix}$$

$$\text{rank}(\mathbf{V}_i) \leq k, \text{rank}(\Lambda) \leq k$$

- So $A \approx \mathbf{V}_1 \Lambda \mathbf{V}_2^T$.
- $\mathbf{V}_1 \Lambda \mathbf{V}_2^T$ is a similarity score matrix for ranking potential affiliations.

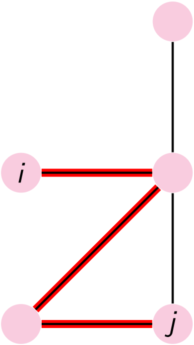
Graph proximity model

Proximity between users in a social network



$(\mathbf{C}^2)_{i,j}$: Number of paths of length 2 between i and j.

Proximity between users in a social network



$(\mathbf{C}^3)_{i,j}$: Number of paths of length 3 between i and j.

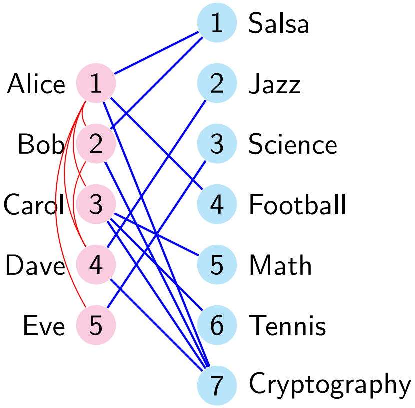
Graph Proximity Model

- Proximity$(i, j) = \beta^2(\mathbf{C}^2)_{i,j} + \beta^3(\mathbf{C}^3)_{i,j} + \ldots$
- Known as Katz measure, when the series is convergent, i.e. $\|\beta\mathbf{C}\|_2 < 1$.
- A practical and good approximation: Truncated Katz,

$$\text{tKatz}(\mathbf{C}, \beta, k) = \sum_{i=1}^{k} \beta^i \mathbf{C}^i.$$

- Recommend affiliations based on proximity in $\mathbf{C}$.

Types of paths considered



$$\text{Eve} \xrightarrow{S} \text{Alice} \xrightarrow{A} \text{Cryptography (in } \mathbf{C}^2\text{)}$$

$$\text{Eve} \xrightarrow{S} \text{Alice} \xrightarrow{AA^T} \text{Bob} \xrightarrow{A} \text{Cryptography (in } \mathbf{C}^4\text{)}$$

Scalability

Real world networks are huge!

- *Orkut* (sub)network [Mislove,2007] is about 3 million users and 8 million groups.
- Recall $\text{tKatz}(\mathbf{C}, \beta, k) = \sum_{i=1}^{k} \beta^i \mathbf{C}^i$.
- $\mathbf{C}^i$ gets denser — prohibitively expensive computations and memory usage.

So how does the model scale?

- A plausible solution...
- Use low rank approximations — $\mathbf{C} \approx \mathbf{V}\Lambda\mathbf{V}^T$.
- Then, $\mathbf{C}^i \approx \mathbf{V}\Lambda^i\mathbf{V}^T$. [Submitted]

Smarter solutions...

- $\text{tKatz}(\mathbf{C}; \beta, 3)_{12} = \beta A + \beta^2 \lambda S A + \beta^3(\lambda^2 S^2 A + A A^T A)$.
- $(AA^T)^i, S^i, (AA^T)^j S^i$ get denser.

$$A = U_A \Sigma_A V_A^T$$
$$S = U_S \Sigma_S U_S^T$$

- Approximate A and S using common subspace of U_A and U_S.

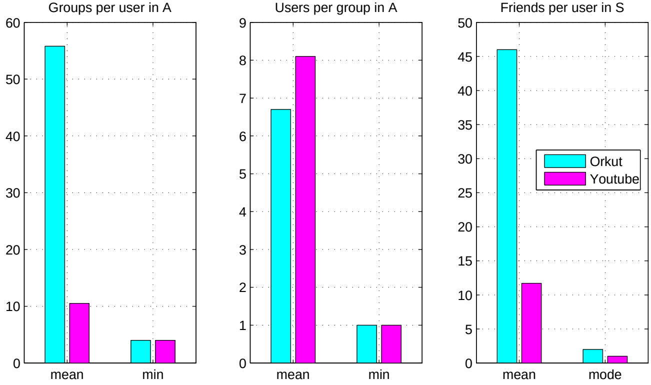
$$A \approx Q D_A V^T$$
$$S \approx Q D_S Q^T$$
$$Q = f(U_A, U_S), Q^T Q = I, V^T V = I$$

- Efficiently compute the terms now! e.g. $(AA^T)^j S^i \approx Q (D_A D_A^T)^j {D_S}^i Q^T$.
- Clustered low-rank approximations [Submitted].

Evaluation of the algorithms

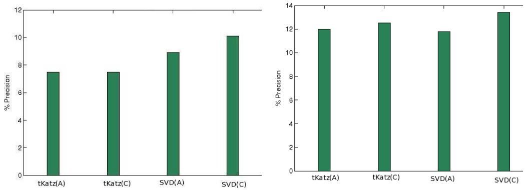
Data sets

Extracted social and affiliation networks: *Orkut* and *Youtube* data sets [Mislove,2007]; Orkut: $N_u = 9123, N_g = 75546$. Youtube: $N_u = 16575, N_g = 21326$.



Evaluation methods

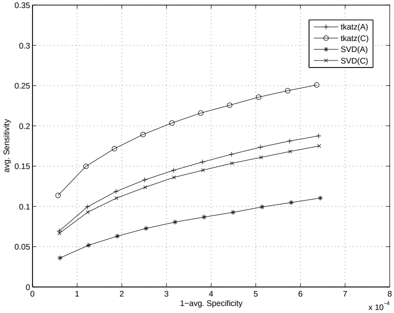
- Choosing appropriate evaluation method — Depends on the end user of the recommendation system.
- "Global" sensitivity vs "Per-user" sensitivity.
- Using Global sensitivity...



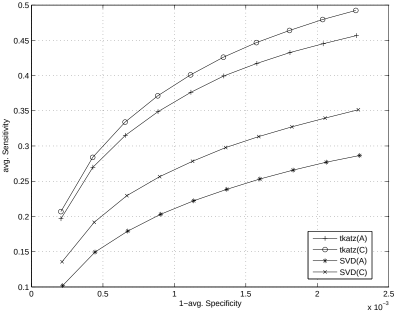
(a) Orkut dataset (b) Youtube dataset

Results: “Per-user” sensitivity

Consider the top k recommendations made for a user for $k = 5, 10, \ldots, 50$.



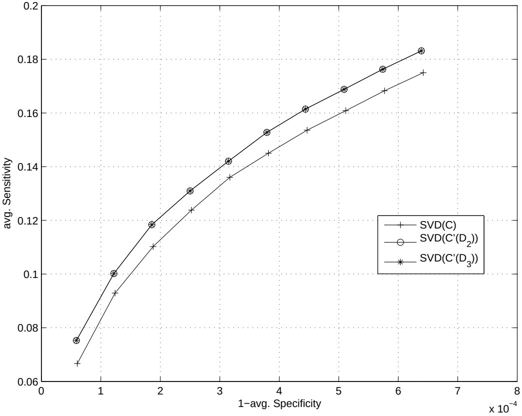
(c) Orkut dataset



(d) Youtube dataset

Similarity between affiliations in the combined network?

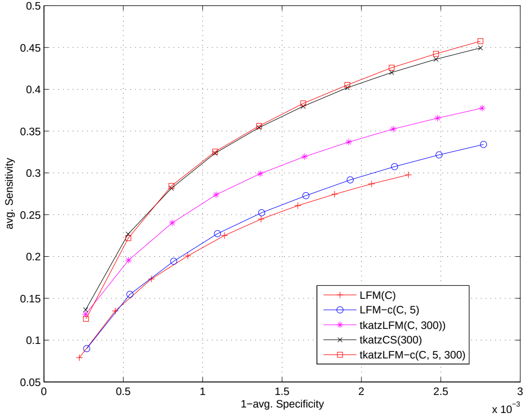
$$\text{Recall } \mathbf{C} = \begin{bmatrix} \lambda S & A \\ A^T & \mathbf{D} \end{bmatrix}$$



$$\mathbf{D} = 0, \mathbf{D}_2 = \mathbf{A}^T\mathbf{A}, \mathbf{D}_3 = \lambda\mathbf{A}^T\mathbf{A}.$$

Information from **D** may be redundant!

Scalable approximations: Youtube

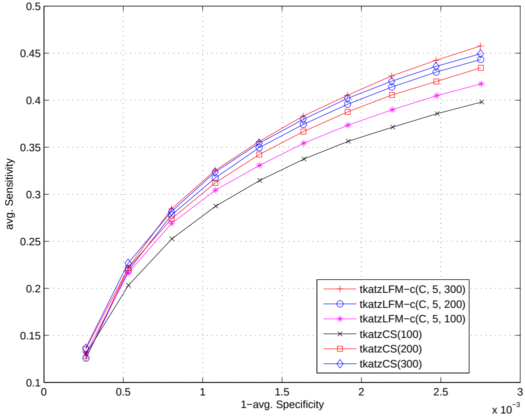


tKatzLFM: tKatz on low-rank approximation.

tKatzCS: tKatz on low-rank approximation using common subspace.

and other clustered approximation variants...

Quality of approximations



Conclusions

Summary

- Friendship network is indeed useful in recommending affiliations!
- Community recommendation – link prediction perspective.
- Two ways of modeling the information from auxiliary networks — Latent Factor and Graph Proximity models.

Future work

- Using affiliation networks for link prediction in friendship networks – Seems harder.
- More sources of information – How do you use them all?
- More scalable models.

References

- Vishvas Vasuki, Nagarajan Natarajan, Zhengdong Lu, Inderjit Dhillon. Affiliation recommendations using auxiliary networks. RECSYS, 2010.
- Vishvas Vasuki, Nagarajan Natarajan, Zhengdong Lu, Berkant Savas, Inderjit Dhillon. Scalable affiliation recommendations using auxiliary networks. Submitted to Transactions on Intelligent Systems and Technology, 2010.
- Alan Mislove etal. Measurement and analysis of online social networks, *In IMC '07: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 29-42, NY, USA, 2007. ACM.

Thank you!