Making predictions involving pairwise data

Aditya Menon and Charles Elkan

University of California, San Diego

September 17, 2010

Overview of talk

- Propose a new problem, dyadic label prediction, and explain its importance
  - Within-network classification is a special case
- Show how to learn supervised latent features to solve the dyadic label prediction problem
- Compare different approaches to the problem from different communities
- Highlight remaining challenges

Outline

The dyadic prediction problem

- Supervised learning:

  Labeled examples $(x_i, y_i) \rightarrow$ Predict label of unseen example x'

- Dyadic prediction:

  Labeled dyads $((r_i, c_i), y_i) \rightarrow$ Predict label of unseen dyad (r', c')

- Labels describe interactions between pairs of entities
  - **Example**: (user, movie) dyads with a label denoting the rating (collaborative filtering)
  - **Example**: (user, user) dyads with a label denoting whether the two users are friends (link prediction)

Dyadic prediction as matrix completion

- Imagine a matrix $X \in \mathcal{X}^{m \times n}$, with rows indexed by r_i and columns by c_i
- The space $\mathcal{X} = \mathcal{X}' \cup \{?\}$
  - Entries with value "?" are missing
- The dyadic prediction problem is to predict the value of the missing entries
- Henceforth call the r_i row objects, the c_i column objects

Dyadic prediction and link prediction

- Consider a graph where only some edges are observed.
- Link prediction means predicting the presence/absence of edges
- There is a two-way reduction between the problems
  - Link prediction is dyadic prediction on an adjacency matrix
  - Dyadic prediction is link prediction on a bipartite graph with nodes for the rows and columns
- Can apply link prediction methods for dyadic prediction, and vice versa
  - Will be necessary when comparing methods later in the talk

Latent feature methods for dyadic prediction

- Common strategy for dyadic prediction: learn latent features
- Simplest form: $X \approx UV^T$
  - $U \in \mathbb{R}^{m \times k}$
  - $V \in \mathbb{R}^{n \times k}$
  - $k \ll \min(m, n)$ is the number of latent features
- Learn U, V by optimizing (nonconvex) objective

$$||X - UV^T||_O^2 + \frac{\lambda_U}{2}||U||_F^2 + \frac{\lambda_V}{2}||V||_F^2$$

  where $||\cdot||_O^2$ is the Frobenius norm over non-missing entries
- Can be thought of as a form of regularized SVD

Outline

Label prediction for dyads

- Want to predict labels for individual row/column entities:

$$\begin{array}{l}\text{Labeled dyads } ((r_i, c_i), y_i) \\ \qquad + \qquad\qquad \rightarrow \text{Predict label of unseen entity } r' \\ \text{Labeled entities } (r_i, y_i^r)\end{array}$$

- Optionally, predict labels for dyads too
- Attach labels to row objects only, without loss of generality
- Let $y_i^r \in \{0, 1\}^L$ to allow multi-label prediction

Dyadic label prediction as matrix completion

- New problem is also a form of matrix completion
- Input is standard dyadic prediction matrix $X \in \mathcal{X}^{m \times n}$ and matrix $Y \in \mathcal{Y}^{m \times L}$
- Each column of Y is one tag
- As before, let $\mathcal{Y} = \{0, 1\} \cup \{?\}$ where "?" means missing
- Y can have any pattern of missing entries
- Goal is to fill in missing entries of Y
- Optionally, fill in missing entries of X, if any

Important real-world applications

- Predict if users in a collaborative filtering population will respond to an ad campaign
- Score suspiciousness of users in a social network, e.g. probability to be a terrorist
- Predict which strains of bacteria will appear in food processing plants [2]

Dyadic label prediction and supervised learning

- An extension of transductive supervised learning:
- We predict labels for individual examples, but:
  - Explicit features (side information) for examples may be absent
  - Relationship information between examples is known via the X matrix
  - Relationship information may have missing data
  - Optionally, predict relationship information also

Within-network classification

- Consider $G = (V, E)$, where nodes $V' \subseteq V$ have labels
- Predicting labels for nodes in $V \backslash V'$ is called within network classification
- An instance of dyadic label prediction:
  X is the adjacency matrix of G, while Y consists of node labels

Why is the dyadic interpretation useful?

- We can let edges E be partially observed, combining link prediction with label prediction
- Can use existing methods for dyadic prediction for within-network classification
  - Exploit advantages of dyadic prediction methods such as ability to use side information
  - Learn latent features

Outline

Latent feature approach to dyadic label prediction

- Given features for row objects, predicting labels in Y is standard supervised learning
- But we don't have such features?
  - Can learn them using a latent feature approach
  - Model $X \approx UV^T$ and think of U as a feature representation for row objects
- Given U, learn a weight matrix W via ridge regression:

$$\min_W ||Y - UW^T||_F^2 + \frac{\lambda_W}{2}||W||_F^2$$

The SocDim approach

- SocDim method for within-network classification on G [3]
  - Compute modularity matrix from adjacency matrix X:

$$\mathcal{Q}(X) = X - \frac{1}{2|E|} dd^T$$

    where d is vector of node degrees
  - Latent features are eigenvectors of $\mathcal{Q}(X)$
  - Use latent features in standard supervised learning to predict Y
- Special case of our approach: G undirected, no missing edges, Y not multilabel, U unsupervised

Supervised latent feature approach

- We learn U to jointly model the data and label matrices, yielding supervised latent features:

$$\min_{U,V,W} ||X - UV^T||_F^2 + ||Y - UW^T||_F^2 + \frac{1}{2}(\lambda_U ||U||_F^2 + \lambda_V ||V||_F^2 + \lambda_W ||W||_F^2).$$

- Equivalent to

$$\min_{U,V,W} ||[XY] - U[V;W]^T||_F^2 + \frac{1}{2}(\lambda_U ||U||_F^2 + \lambda_V ||V||_F^2 + \lambda_W ||W||_F^2)$$

- Intuition: treat the tags as new movies

Why not use the reduction?

- If goal is predicting labels, reconstructing X is less important
- So, weight the "label movies" with a tradeoff parameter μ:

$$\min_{U,V,W} ||X - UV^T||_F^2 + \mu||Y - UW^T||_F^2 + \frac{1}{2}(\lambda_U||U||_F^2 + \lambda_V||V||_F^2 + \lambda_W||W||_F^2)$$

- Assuming no missing entries in X, essentially supervised matrix factorization (SMF) method [4]
  - SMF was designed for directed graphs, unlike SocDim

From SMF to dyadic prediction

- Move from SMF approach to one based on dyadic prediction
- Obtain important advantages
  - Deal with missing data in X
  - Allow arbitrary missingness in Y, including partially observed rows
- Specifically, use LFL approach [1]
  - Exploit side-information about the row objects
  - Predict calibrated probabilities for tags
  - Handle nominal and ordinal tags

Latent feature log-linear (LFL) model

- Assume discrete entries in input matrix X, say $\{1, \ldots, R\}$
- Per row and per column, have a latent feature vector for each outcome: U_i^r and V_j^r
- Posit log-linear probability model

$$p(X_{ij} = r | U, V) = \frac{\exp\,(U_i^r)^T V_j^r}{\sum_{r'} \exp\,(U_i^{r'})^T V_j^{r'}}$$

LFL inference and training

- Model is

$$p(X_{ij} = r|U, V) = \frac{\exp\,(U_i^r)^T V_j^r}{\sum_{r'} \exp\,(U_i^{r'})^T V_j^{r'}}$$

- For nominal outcomes, predict argmax $p(r|U, V)$
- For ordinal outcomes, predict $\sum_r r p(r|U, V)$
- Optimize MSE for ordinal outcomes
- Optimize log-likelihood for nominal outcomes; get well-calibrated predictions

Incorporating side-information

- Known features can be highly predictive for matrix entries
- They are essential to solve cold start problems, where there are no existing observations for a row/column
- Let a_i and b_j denote covariates for rows and columns respectively
- Extended model is

$$p(X_{ij} = r|U, V) \propto \exp(\ (U_i^r)^T V_j^r + (w^r)^T \begin{bmatrix} a_i & b_j \end{bmatrix}\ ).$$

- Weight vector w^r says how side-information predicts outcome r

Extending LFL to graphs

- Consider the following generalization of the LFL model:

$$p(X_{ij} = r|U, V, \Lambda) \propto \exp\left(U_i^r\right)^T \Lambda_{ij} V_j^r.$$

- Constrain latent features depending on nature of the graph:
  - If rows and columns are distinct sets of entities, let $\Lambda = I$
  - For asymmetric graphs, set $V = U$ and let Λ be unconstrained
  - For symmetric graphs, set $V = U$ and $\Lambda = I$

Using the LFL model for label prediction

- Idea: Fill in missing entries in X and also missing tags in Y
- Combined regularized optimization is

$$\min_{U,V,W} ||X - \mathcal{E}(X)||_{\mathcal{O}}^2 + \frac{1}{2}\left(\sum_r \lambda_U ||U^r||_F^2 + \lambda_V ||V^r||_F^2\right) + \sum_{(i,l)\in\mathcal{O}} \frac{e^{Y_{il}(W_l^T U_i)}}{1 + e^{W_l^T U_i}} + \frac{\lambda_W}{2}||W||_F^2$$

- If entries in X are ordinal then

$$\mathcal{E}(X)_{ij} = \sum_r r \cdot p(X_{ij} = r | U, V)$$

Outline

Summary of methods

- Three previously unrelated approaches to label prediction:
  - SocDim
  - SMF
  - LFL
- They haven't been compared before
- How do they differ?

Comparison of approaches

- Properties of the methods:

Item	SocDim	SMF	LFL
Supervised latent features?	No	Yes	Yes
Asymmetric graphs?	No	Yes	Yes
Handles missing data?	No	No	Yes
Finds latent features of?	Modularity	Data	Data
Single minimum?	Yes	No	No

- Many differences arise as a result of the objective function being optimized

Alternative objective functions

- Compare objective functions for a shared special case:
  - Since SocDim and SMF operate natively on graphs, assume X is a graph
  - Assume no missing data in X, for fairness to SocDim and SMF
  - Assume graph is undirected, as SocDim does
  - Don't learn latent features in a supervised manner, for fairness to SocDim

Comparing objective functions

- **SocDim**: if $\mathcal{Q}$ denotes the modularity matrix, then

$$\min_{U,\Lambda \text{ diagonal}} ||\mathcal{Q}(X) - U\Lambda U^T||_F^2$$

- **Supervised matrix factorization**:

$$\min_{U,\Lambda} ||X - U\Lambda U^T||_F^2 + \frac{\lambda_U}{2}||U||_F^2 + \frac{\lambda_\Lambda}{2}||\Lambda||_F^2$$

- **LFL**: denoting $\sigma(x) = 1/(1+e^{-x})$,

$$\min_{U} ||X - \sigma(UU^T)||_F^2 + \frac{\lambda_U}{2}||U||_F^2$$

- **In general**:

$$\min_{U,\Lambda} ||f(X) - g(U,\Lambda)||_F^2 + \frac{\lambda_U}{2}||U||_F^2 + \frac{\lambda_\Lambda}{2}||\Lambda||_F^2$$

SocDim versus LFL

- SocDim transforms the input X but LFL transforms the estimate
- Transforming the estimate ensures $[0, 1]$ predictions
- Transforming the input is analogous to spectral clustering:
  - The graph Laplacian normalizes nodes wrt their degrees
- Does the input transformation make a difference?
- Does SocDim perform similarly using the Laplacian instead of modularity?

SocDim versus SMF

- Without supervised features or missing data, two differences:
  - SocDim uses modularity matrix, while SMF uses data matrix
  - SocDim has closed form solution, while SMF does not
  - SocDim is immune to local optima
- Global optimum may offset issue that SocDim is unsupervised

Outline

Questions for empirical study

- Do supervised latent features help?
- Does immunity to local optima help?
- Which data transform is best? Does it matter?
  - Can using the Laplacian matrix with SocDim improve performance?
  - Can using the modularity or Laplacian matrix with SMF improve performance?
- Can naïve approaches to missing edges succeed?
  - Just impute row/column averages for missing entries?
  - If so, then SocDim and SMF can be applied to more problems

Datasets

- `blogcatalog`: Fully observed links between 2500 bloggers in a directory. Labels are users' interests, divided into 39 categories (multilabel problem)
- `senator`: "Yea" or "Nay" votes of 101 U.S. senators on 315 bills. Label is Republican or Democrat
- `usps`: Binarized grayscale 16×16 images of handwritten digits. We occlude some pixels, so X has missing entries. Labels are the true digits.
  - Shows how dyadic label prediction can solve a difficult version of a standard supervised learning task

Accuracy measures

- For `senator` and `usps` binary tasks, 0-1 error
- For `blogcatalog` multi-label task, F1-micro and F1-macro scores
  - Given true tags y_{il} and predictions $\hat{y}_{il}$

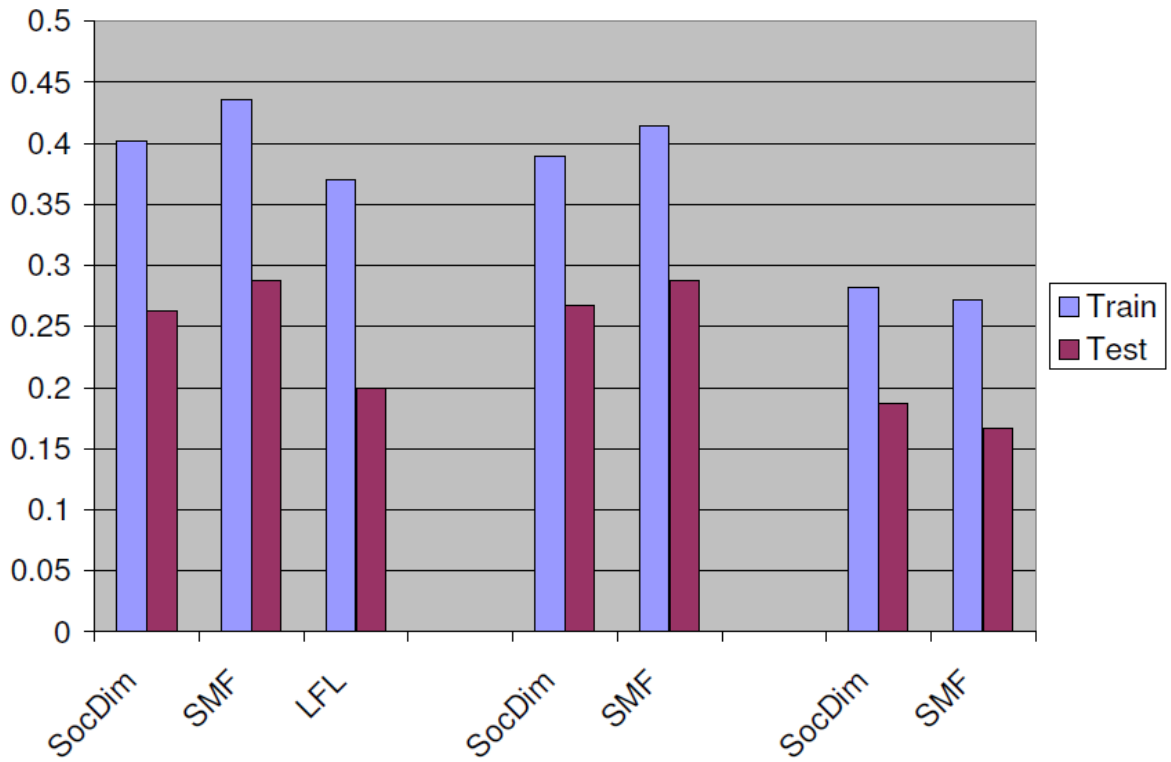
$$\text{micro} = 2\frac{\sum_{i,l} y_{il}\hat{y}_{il}}{\sum_{i,l} y_{il} + \hat{y}_{il}}$$

$$\text{macro} = \frac{2}{L}\sum_{l}\frac{\sum_{i} y_{il}\hat{y}_{il}}{\sum_{i} y_{il} + \hat{y}_{il}}$$

- 10-fold cross-validation

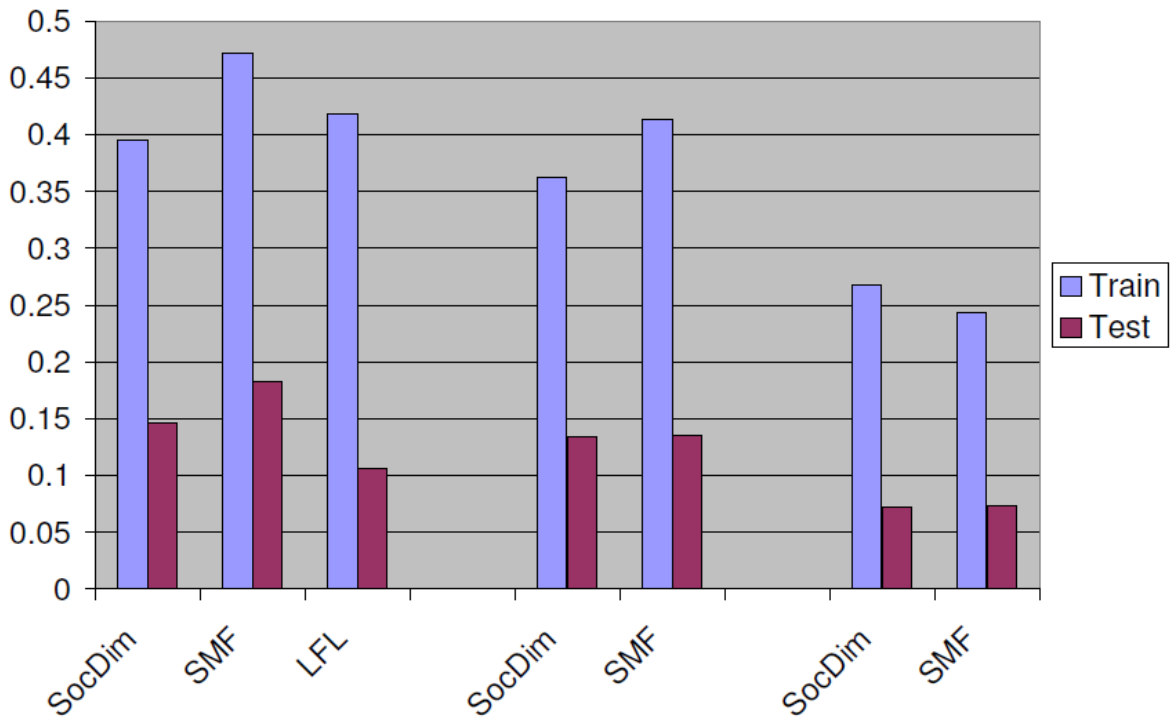
F1-micro results on blogcatalog

- Left to right: adjacency matrix, modularity, Laplacian
- Blue training, red test. Higher is better
- SMF is best. Raw data matrix is as good modularity
- All methods overfit, despite ℓ_2 regularization



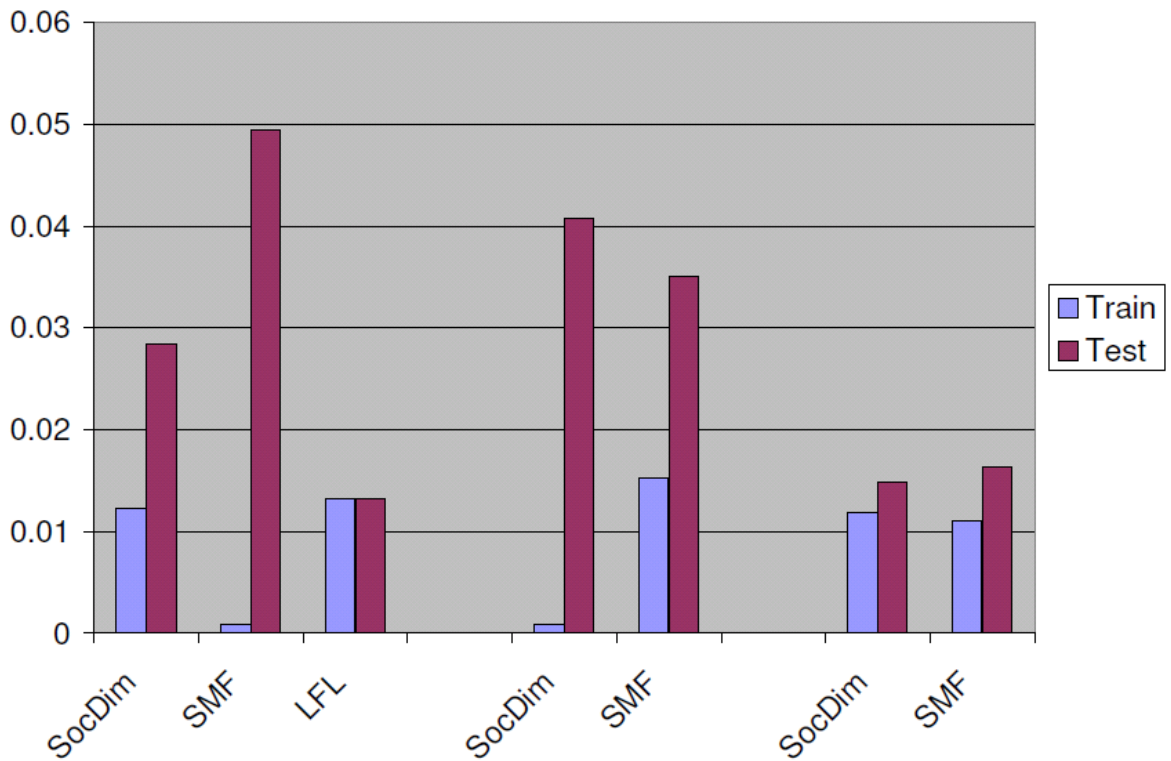
F1-macro results on blogcatalog

- Left to right: adjacency matrix, modularity, Laplacian
- Blue training, red test. Higher is better
- SMF is also best. Raw data matrix is best
- All methods overfit



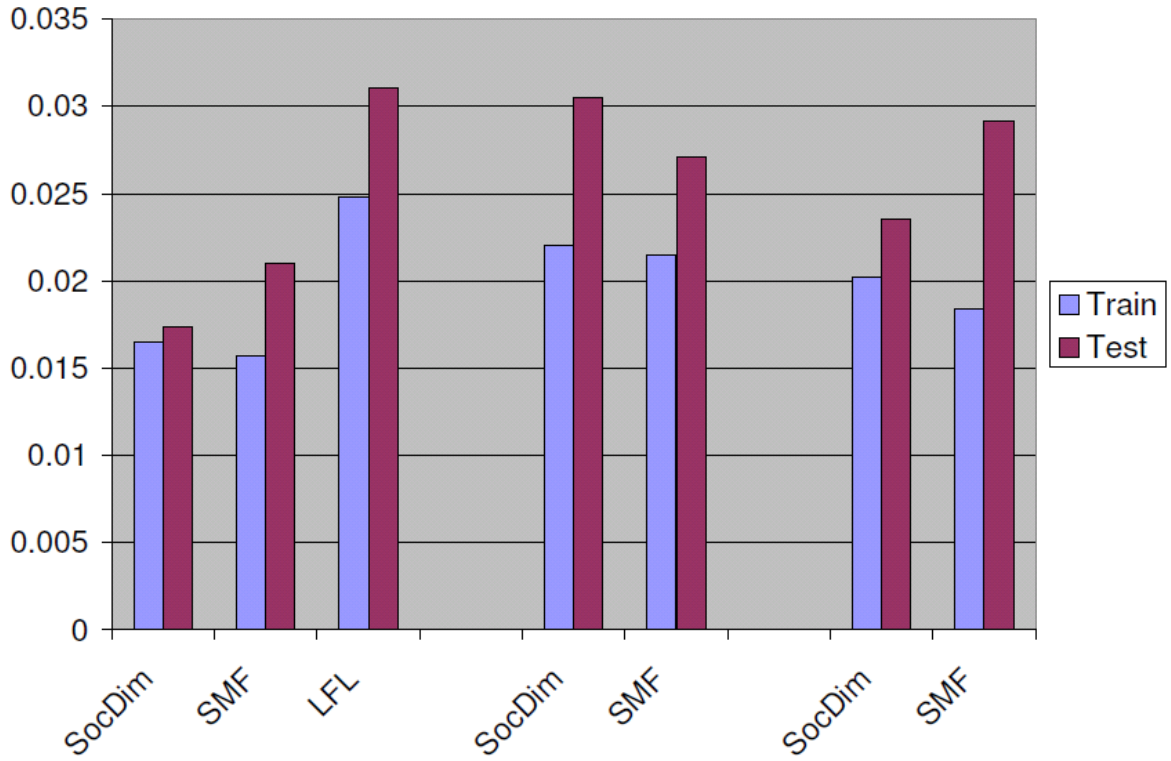
Results on `senator`

- Left to right: adjacency matrix, modularity, Laplacian
- Blue training, red test. Lower is better.
- LFL is best
- Other two methods overfit badly



Results on usps

- Left to right: adjacency matrix, modularity, Laplacian
- Blue training, red test. Lower is better.
- SocDim is best, despite ignoring missing values
- Raw data matrix is best



Outline

Conclusions

- Unified label prediction, within-network prediction,
- Unified collaborative filtering with cold start and link prediction with side-information
- Unified label prediction and within-network prediction,
- Showed how to use supervised latent features to predict labels and links
- Experiments show that good regularization is an open problem

Outline

References

Aditya Krishna Menon and Charles Elkan.

Dyadic prediction using a latent feature log-linear model.

`http://arxiv.org/abs/1006.2156`, 2010.

Purnamrita Sarkar, Lujie Chen, and Artur Dubrawski.

Dynamic network model for predicting occurrences of salmonella at food facilities.

In *Proceedings of the BioSecure International Workshop*, pages 56–63. Springer, 2008.

Lei Tang and Huan Liu.

Relational learning via latent social dimensions.

In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 817–826. ACM, 2009.

Shenghuo Zhu, Kai Yu, Yun Chi, and Yihong Gong.

Combining content and link for classification using matrix factorization.

In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 487–494. ACM, 2007.