

Link prediction via matrix factorization

Charles Elkan

University of California, San Diego

September 6, 2011

Outline

- 1 Introduction: Three related prediction tasks
- 2 Link prediction in networks
- 3 Discussion

Link prediction

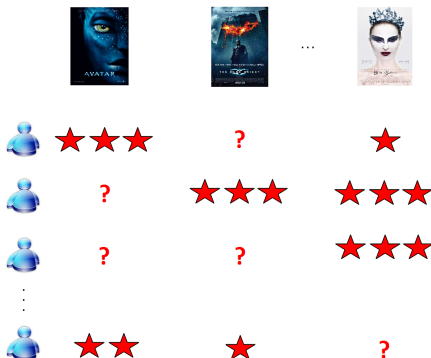
- Given current friendship edges, predict future edges.



- Application: Facebook.
- Popular method: Scores computed from graph topology, e.g. betweenness.

Collaborative filtering




















- Given ratings of movies by users, predict other ratings.



- Application: Netflix.
- Popular method: Matrix factorization.

Item response theory

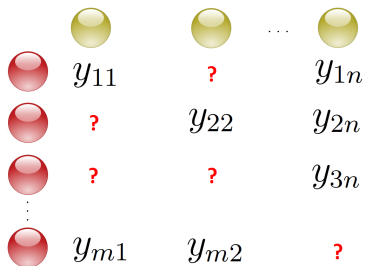
- Given answers by students to exam questions, predict performance on other questions.

	 	 	 
			
			
⋮			
			

- Applications: Adaptive testing, diagnosis of skills.
- Popular method: Latent trait (i.e. hidden feature) models.

Dyadic prediction in general

- Given labels for some pairs of items (some **dyads**), predict labels for other pairs.



- What if we have side-information, e.g. mobility data for people in a social network?

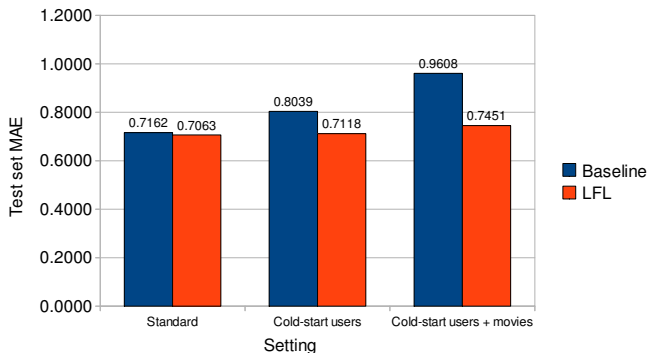
Matrix factorization

- Associate **latent feature values** with each user and movie.
- Each rating is the dot-product of corresponding latent vectors.
- Learn the most predictive vector for each user and movie.



Side-information solves the cold-start problem

- **Standard:** All users and movies have training data.
- **Cold-start users:** No ratings for 50 random users.
- **Double cold-start:**
No ratings for 50 random users **and** their movies.

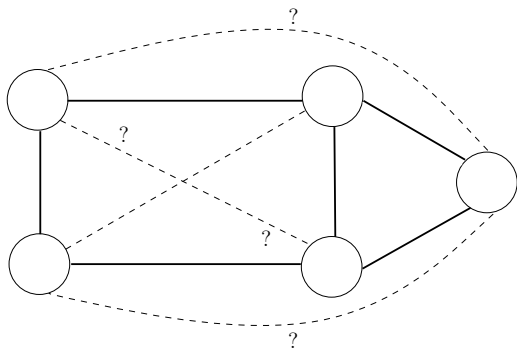


Outline

- 1 Introduction: Three related prediction tasks
- 2 Link prediction in networks
- 3 Discussion

Link prediction

- **Link prediction:** Given a **partially observed** graph, predict whether or not edges exist for the unknown-status dyads.



- Classic methods are unsupervised (non-learning) scores, e.g. betweenness, common neighbors, Katz, Adamic-Adar.

The bigger picture

- Solve a **predictive** problem.
 - ▶ Contrast: Non-predictive task, e.g. community detection.
- Maximize objective defined by an **application**, e.g. AUC.
 - ▶ Contrast: Algorithm but no goal function, e.g. betweenness.
- Learn from **all** available data.
 - ▶ Contrast: Use only graph structure, e.g. commute time.
- Allow **hubs**, **overlapping** groups, etc.
 - ▶ Contrast: Clusters, modularity.
- Make training time **linear** in number of edges.
 - ▶ Contrast: MCMC, betweenness, SVD.
- Compare accuracy to **best** current results.
 - ▶ Contrast: Compare only to classic methods.

Combined latent/explicit feature approach

- Each node's **identity** influences its linking behavior.
- The identity of a node determines its latent features.
- Nodes also can have **side-information** predictive of linking.
 - ▶ For author-author linking, side-information can be words in authors' papers.
- Edges may also possess side-information.
 - ▶ For country-country conflict, side-information is geographic distance, trade volume, etc.

Latent feature model

- LFL model for binary link prediction has parameters
 - ▶ latent vectors $\alpha_i \in \mathbb{R}^k$ for each node i
 - ▶ scaling factors $\Lambda \in \mathbb{R}^{k \times k}$
 - ▶ weights $W \in \mathbb{R}^{d \times d}$ for node features
 - ▶ weights $v \in \mathbb{R}^{d'}$ for edge features.
- Node i has features x_i , dyad ij has features z_{ij} .
- Predicted label is

$$\hat{G}_{ij} = \sigma(\alpha_i^T \Lambda \alpha_j + x_i^T W x_j + v^T z_{ij})$$

for sigmoid function $\sigma(x) = \frac{1}{1 + \exp(-x)}$.

Latent feature training

- True label is G_{ij} , predicted label is \hat{G}_{ij} .
- Minimize regularized training loss:

$$\min_{\alpha, \Lambda, W, v} \sum_{(i,j) \in \mathcal{O}} \ell(G_{ij}, \hat{G}_{ij}) + \Omega(\alpha, \Lambda, W, v)$$

- Sum is **only** over known edges and known non-edges.
- Stochastic gradient descent (SGD) converges quickly.

Challenge: Class imbalance

- Vast majority of node-pairs do not link with each other.
- Area under ROC curve (AUC) is standard performance measure.
- For a random pair of positive and negative examples, AUC is the probability that the positive one has higher score.
 - ▶ Not influenced by relative size of positive and negative classes.
- Models trained to maximize accuracy are suboptimal.
 - ▶ **Sampling** is popular, but loses information.
 - ▶ **Weighting** is merely heuristic.

Optimizing AUC

- Empirical AUC counts concordant pairs

$$\text{AUC} \propto \sum_{p \in +, q \in -} \mathbf{1}[f_p - f_q > 0]$$

- Train LFL model to maximize approximation to AUC:

$$\min_{\alpha, \Lambda, W, v} \sum_{(i, j, k) \in \mathcal{D}} \ell(\hat{G}_{ij} - \hat{G}_{ik}, 1) + \Omega(\alpha, \Lambda, W, v)$$

where $\mathcal{D} = \{(i, j, k) : G_{ij} = 1, G_{ik} = 0\}$.

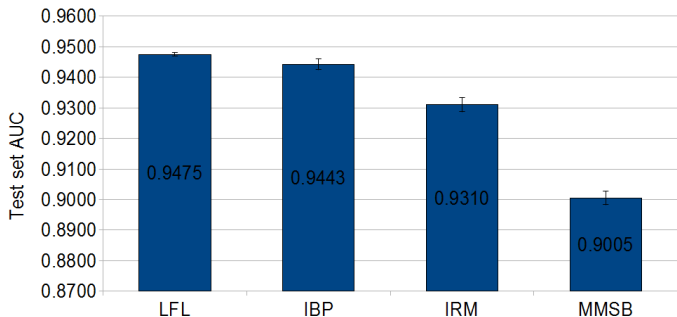
- With stochastic gradient descent, a fraction of one epoch is enough for convergence.

Experimental comparison

- Compare
 - ▶ **latent** features versus **unsupervised** scores
 - ▶ **latent** features versus **explicit** features.
- Datasets from applications of link prediction:
 - ▶ **Computational biology**: Protein-protein interaction network, metabolic interaction network
 - ▶ **Citation networks**: NIPS authors, condensed matter physicists
 - ▶ **Social phenomena**: Military conflicts between countries, U.S. electric power grid, multiclass relationships.

Multiclass link prediction

- **Alyawarra** dataset has **kinship** relations for 104 people {brother, sister, father, ... }.
- LFL outperforms Bayesian models, even infinite ones.



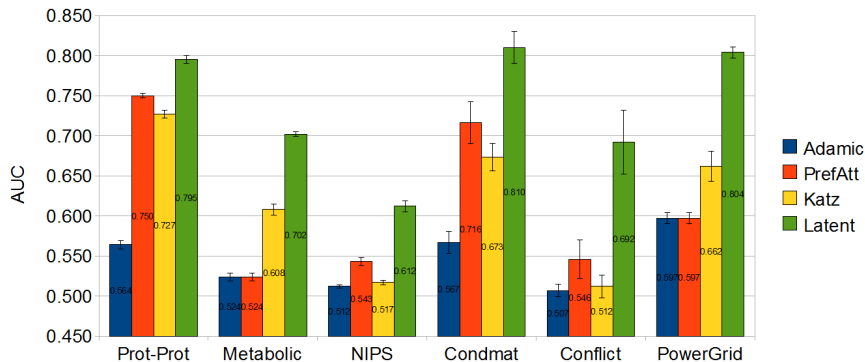
Binary link prediction datasets

	nodes	$ O^+ $	$ O^- $	+ve:-ve ratio	mean degree
Prot-Prot	2617	23710	6,824,979	1 : 300	9.1
Metabolic	668	5564	440,660	1 : 80	8.3
NIPS	2865	9466	8,198,759	1 : 866	3.3
Condmat	14230	2392	429,232	1 : 179	0.17
Conflict	130	320	16580	1 : 52	2.5
PowerGrid	4941	13188	24,400,293	1 : 2000	2.7

- Protein-protein interaction data from Noble. Per protein: 76 features.
- Metabolic interactions of *S. cerevisiae* from the KEGG/PATHWAY database. Per protein: 157 phylogenetic features, 145 gene expression features, 23 location features.
- NIPS. Per author: 100 LSI features from vocabulary of 14,035 words.
- Condensed-matter physicists [Newman]. Use node-pairs 2 hops away in first five years.
- Military disputes [MID 3.0]. Per country: population, GDP, polity. Per dyad: 6 features, e.g. geographic distance.
- US electric power grid network [Watts and Strogatz].

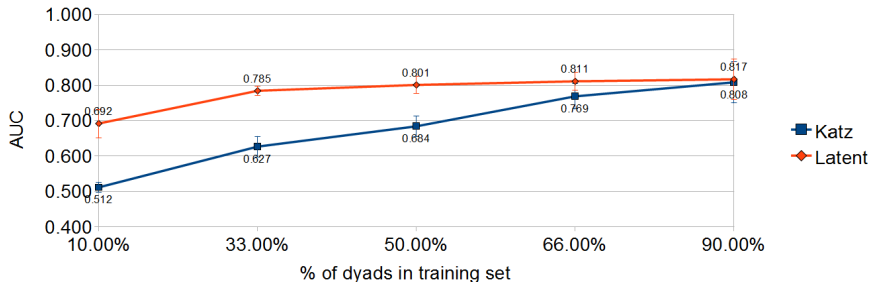
Latent features versus unsupervised scores

- Latent features are more predictive of linking behavior.



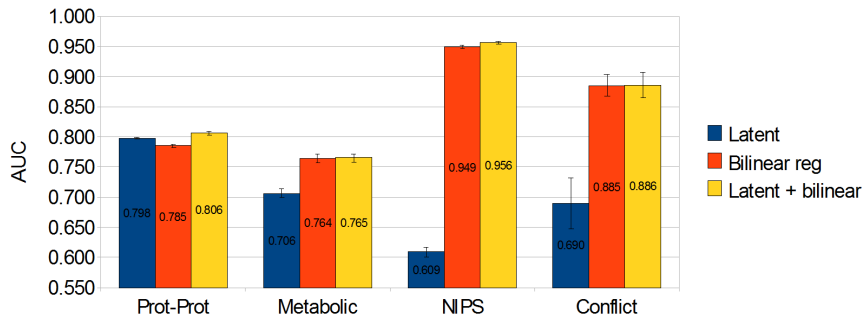
Learning curves

- Unsupervised scores need many edges to be known.
- Latent features are predictive with fewer known edges.
- For the military conflicts dataset:



Latent features combined with side-information

- Difficult to infer latent structure more predictive than side-information.
- But combining the two is beneficial:



Related paper in Session 19, Thursday am

- *Kernels for Link Prediction with Latent Feature Models*, Nguyen and Mamitsuka, ECML 2011.
- Fruit fly protein-protein interaction network, 2007 data.
- Connected component with minimum degree 8: 701 nodes (713).
- 100 latent features, tenfold CV: AUC 0.756 \pm 0.012.
- Better than IBP (0.725), comparable to kernel method.

Outline

- 1 Introduction: Three related prediction tasks
- 2 Link prediction in networks
- 3 Discussion**

If time allowed

- Scaling up to Facebook-size datasets: better AUC than supervised random walks.
- Predicting labels for nodes, e.g. who will play Farmville (within network/collective/semi-supervised classification).

Conclusions

- Many prediction tasks involve pairs of entities: collaborative filtering, friend suggestion, and more.
- Learning latent features always gives better accuracy than any non-learning method.
- The most accurate predictions combine latent features with explicit features of nodes and of dyads.
- You don't need EM, variational Bayes, MCMC, infinite number of parameters, etc.

References I