Learning a Distance Metric for Structured Network Prediction

Stuart Andrews and Tony Jebara
Columbia University

Outline

- Introduction
  - Context, motivation & problem definition
- Contributions
  - Structured network characterization
  - Network prediction model
  - Distance-based score function
  - Maximum-margin learning
- Experiments
  - 1-Matchings on toy data
  - Equivalence networks on face images
  - Preliminary results on social networks
- Future & related work, summary and conclusions

Context

- Pattern classification
  - Inputs & outputs
  - Independent and identically distributed
- Pattern classification for structured objects
  - Sets of inputs & outputs
  - Model dependencies amongst output variables
- Parameterize model using a Mahalanobis distance metric

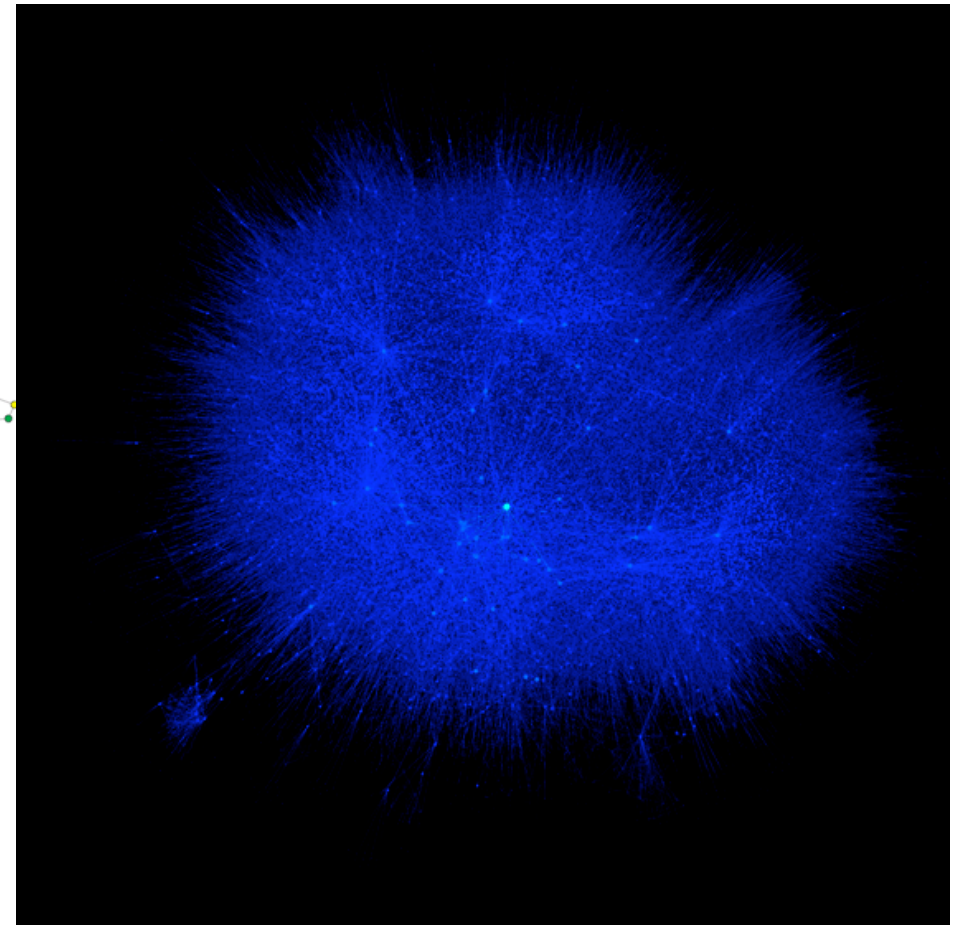
Motivation for structured network prediction

- Man made and natural formed networks exhibit a high degree of structural regularity

Motivation

- Scale free networks

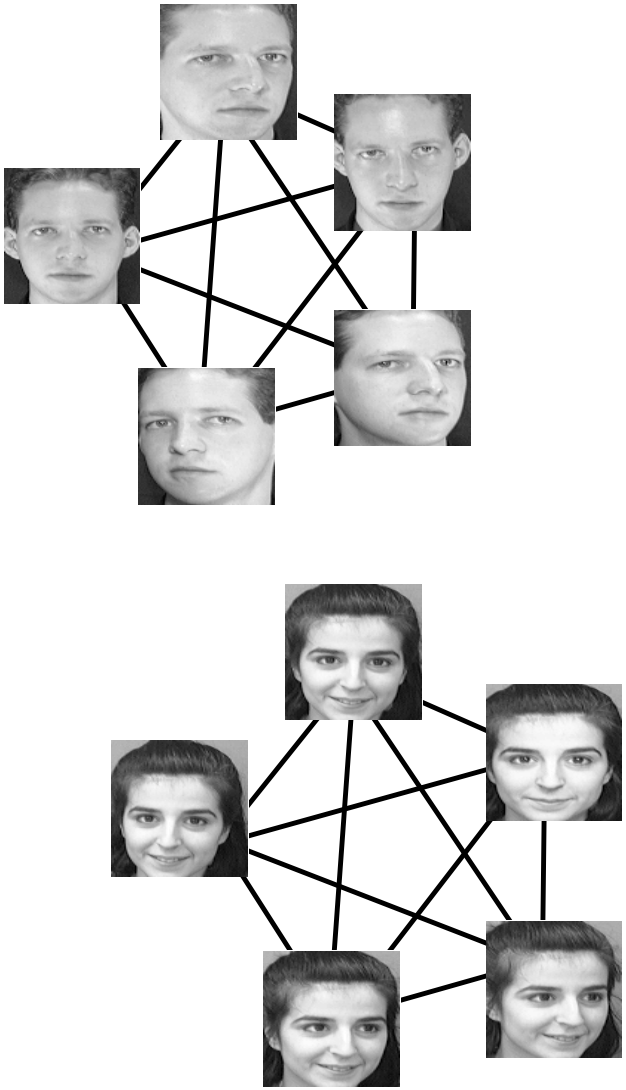
Protein-interaction network, Barabási & Oltvai, Nature Genetics, 2004

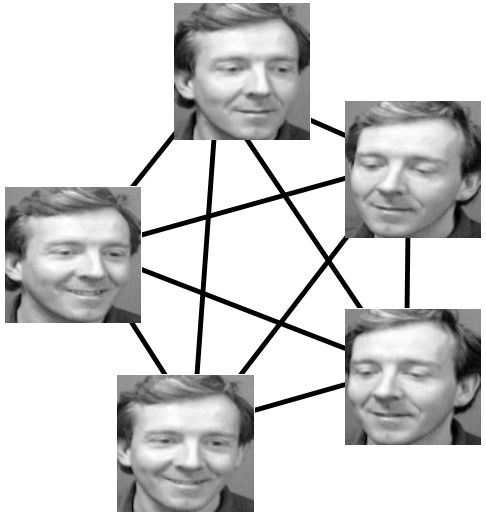


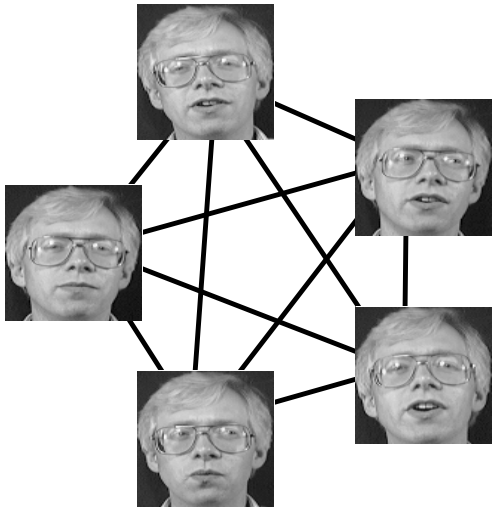
Jeffrey Heer, Berkeley

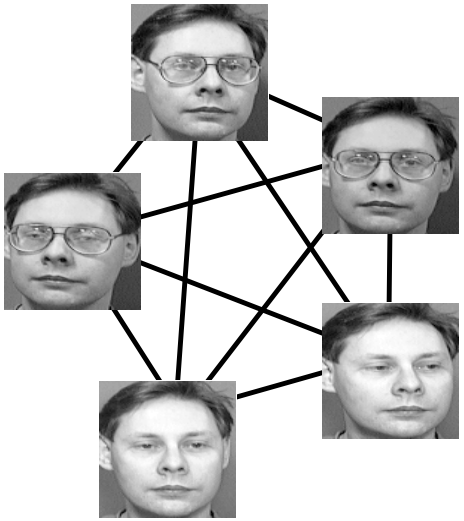
Motivation

- Equivalence networks





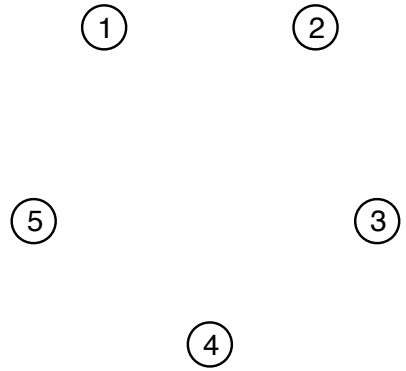




Equivalence network on Olivetti face images - union of vertex-disjoint complete subgraphs

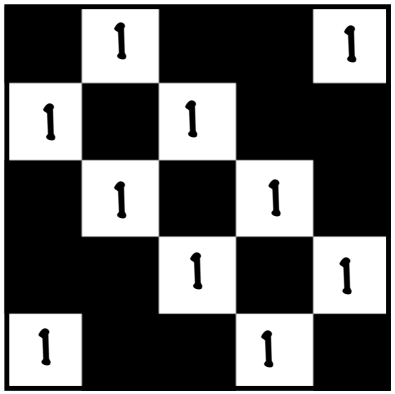
Structured network prediction

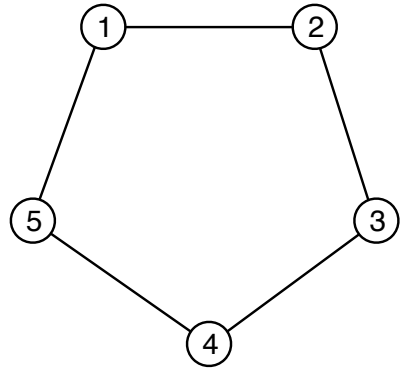
- Given
  - n entities with attributes $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
    $\mathbf{x}_k \in \mathbb{R}^d$
  - And a structural prior on networks



- Output
  - Network of similar entities with desired structure

$$\mathbf{y} = (y_{j,k})$$
$$y_{j,k} \in \{0, 1\}$$





Applications

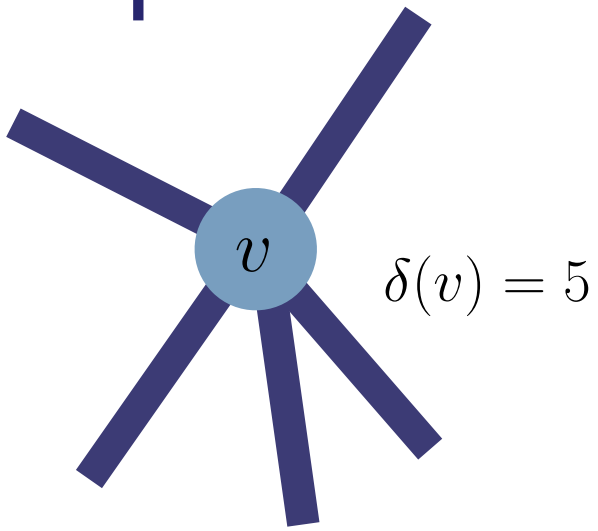
-
- Tasks
  - Initializing
  - Augmenting
  - Filtering of networks
- Domains
  - E-commerce
  - Social network analysis
  - Network biology

Challenges for SNP

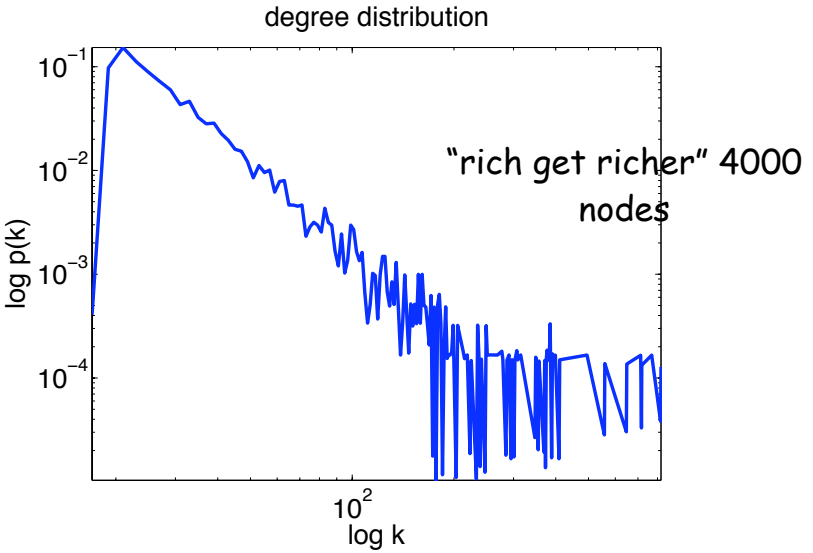
- How can we take structural prior into account?
  - Complex dependencies amongst atomic edge predictions
- What similarity should we use?
  - Avoid engineering similarity metric for each domain

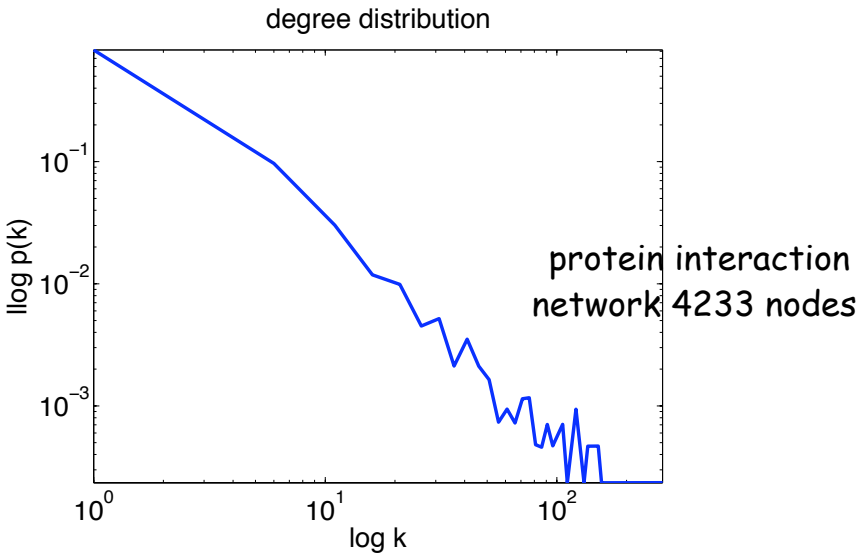
Structural network priors - 1

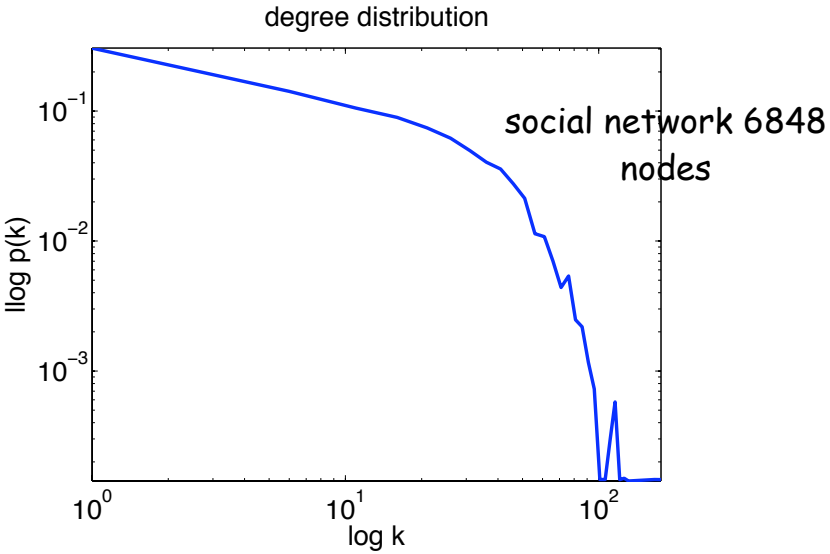
- Degree $\delta(v)$ of a node
  - Number of incident edges
  -
-
- Degree distribution
  - Probability of node having degree k, for all k

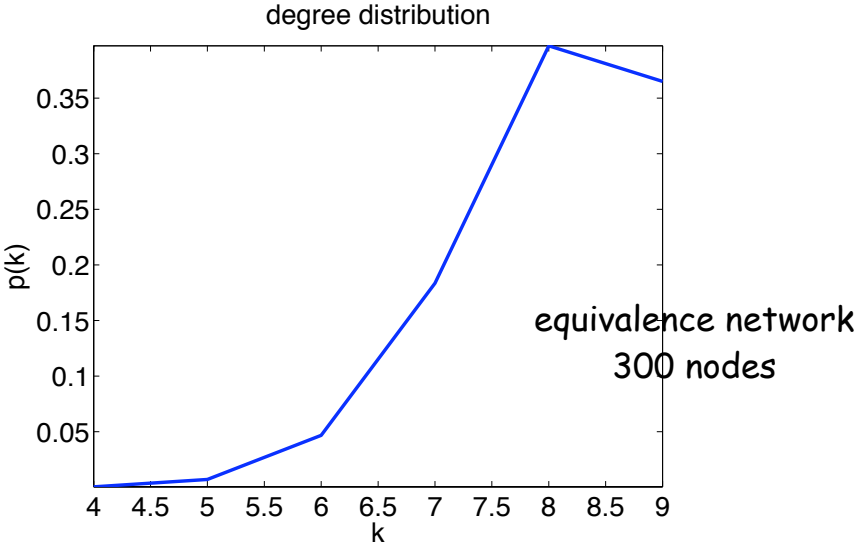


Degree distributions





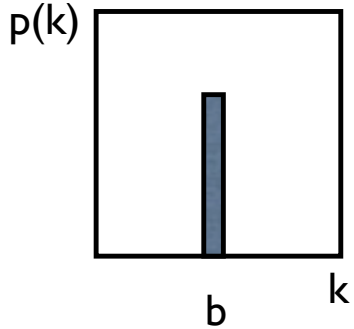




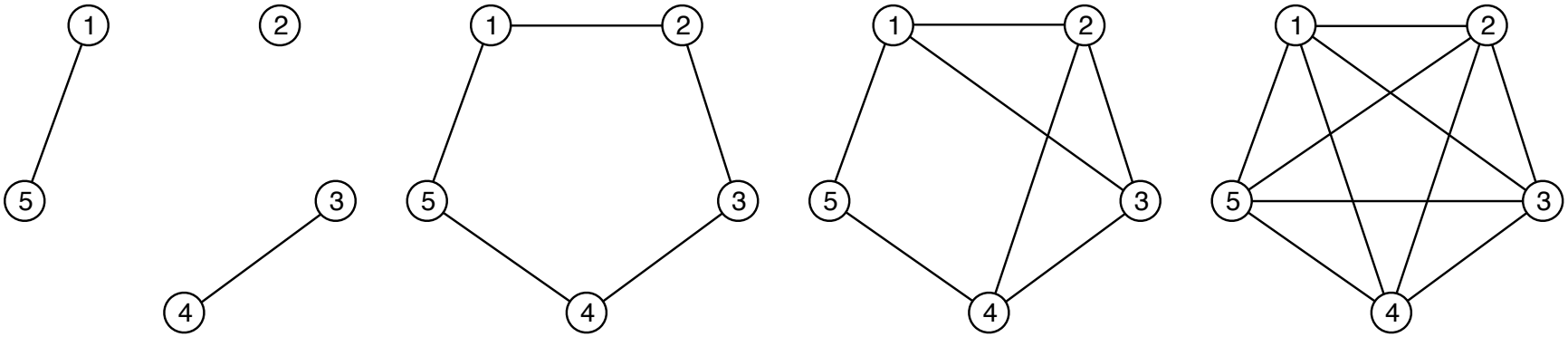
Structural network priors - 2

- Combinatorial families
  - Chains
  - Trees & forests
  - Cycles
  - Unions of disjoint complete subgraphs
  - Generalized matchings

B-matchings



- A b-matching has $\delta(v) = b$ for (almost) all v



1-matching 2-matching 3-matching 4-matching

$$y_{j,k} \in \{0,1\}$$

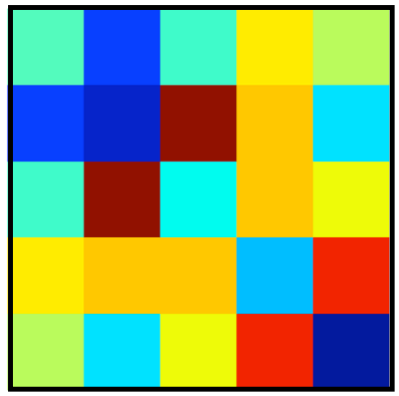
$$\mathbf{y} \in \mathcal{B} \quad \Leftrightarrow \quad \sum_j y_{j,k} = b \,\forall k \qquad \sum_k y_{j,k} = b \,\forall j$$

- We consider B-matching networks $\mathcal{B}$ because they are flexible and efficient

Predictive Model

- Maximum weight b-matching as predictive model

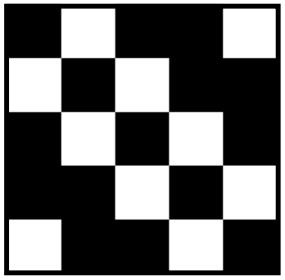
1. Receive nodes and attributes

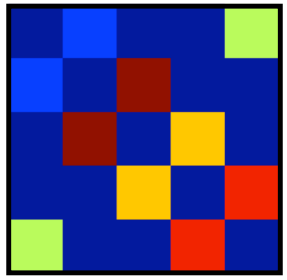


2. Compute edge weights $\mathbf{s} = (s_{j,k}) \qquad s_{j,k} \in \mathbb{R}$

3. Select a b-matching with maximal weight

$$\max_{\mathbf{y} \in \mathcal{B}} \sum_{j,k} y_{j,k} s_{j,k} = \max_{\mathbf{y} \in \mathcal{B}} \mathbf{y}^T \mathbf{s}$$





- B-matchings requires $\mathcal{O}(n^3)$ time

Structured network prediction

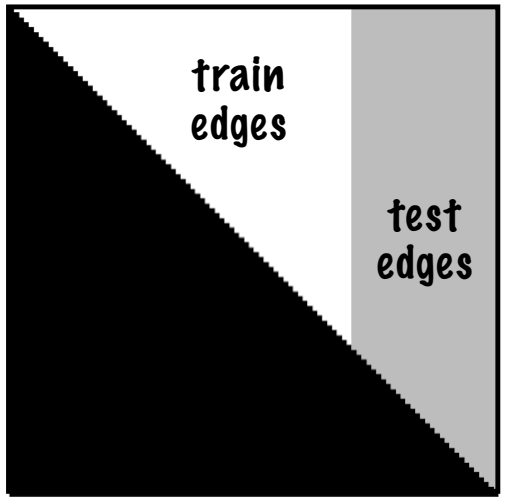
- The question that remains is how do we compute the weights?

Learning the weights

-
- Weights are parameterized by a Mahalanobis distance metric
- $$s_{j,k} = (x_j - x_k)^T Q (x_j - x_k) \qquad Q \succeq 0$$
-
- In other words, we want to find the best linear transformation (rotation & scaling) to facilitate b-matching

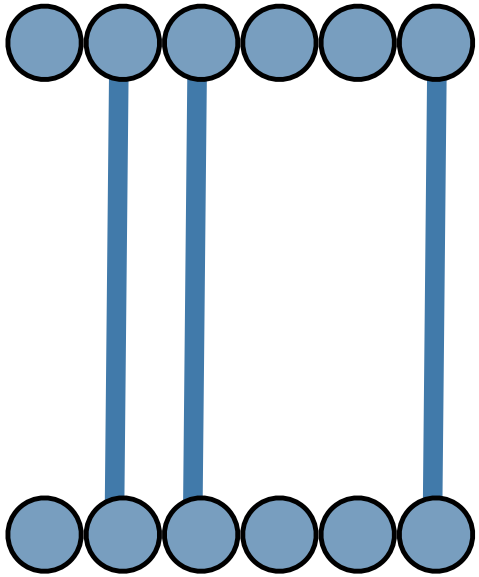
Learning the weights

-
- We propose to learn the weights from one or more partially observed networks
  - We observe the attributes of all nodes
  - But only a subset of the edges
  -
- Transductive approach
  - Learn weights to “fit” training edges
  - While structured network prediction is performed over training and test edges



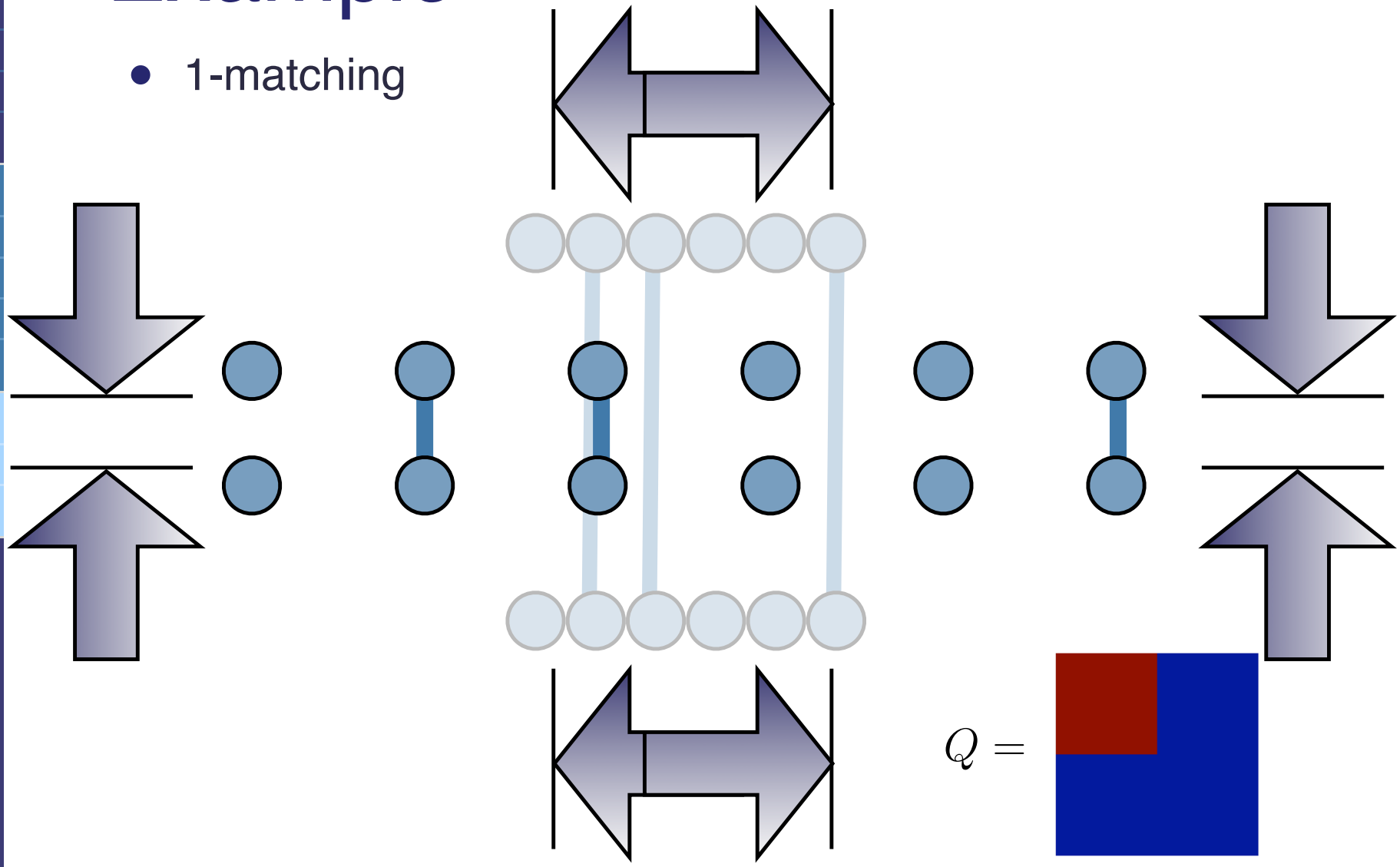
Example

- Given the following nodes & edges



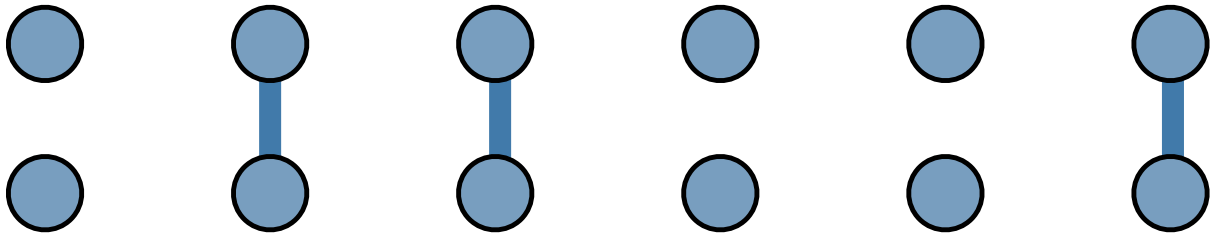
Example

- 1-matching



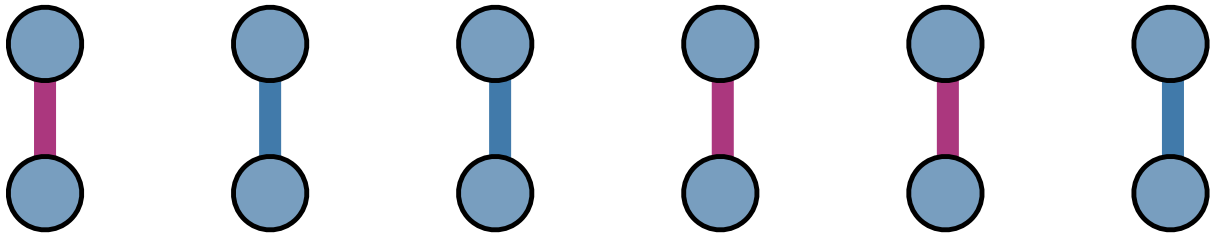
Example

- 1-matching



Example

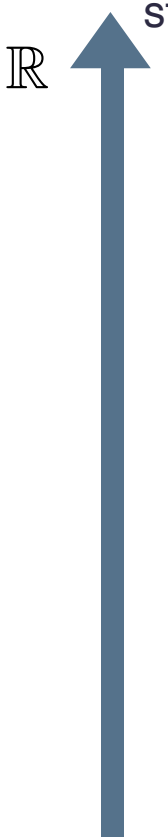
- 1-matching



Maximum-margin

Taskar et al. 2005

- We use the dual-extragradient algorithm to learn Q
  - Define the margin to be the minimum gap between the predictive values of the true structure $y \in \mathcal{B}$ and each possible alternative structure $y_1, y_2, \ldots \in \mathcal{B}$

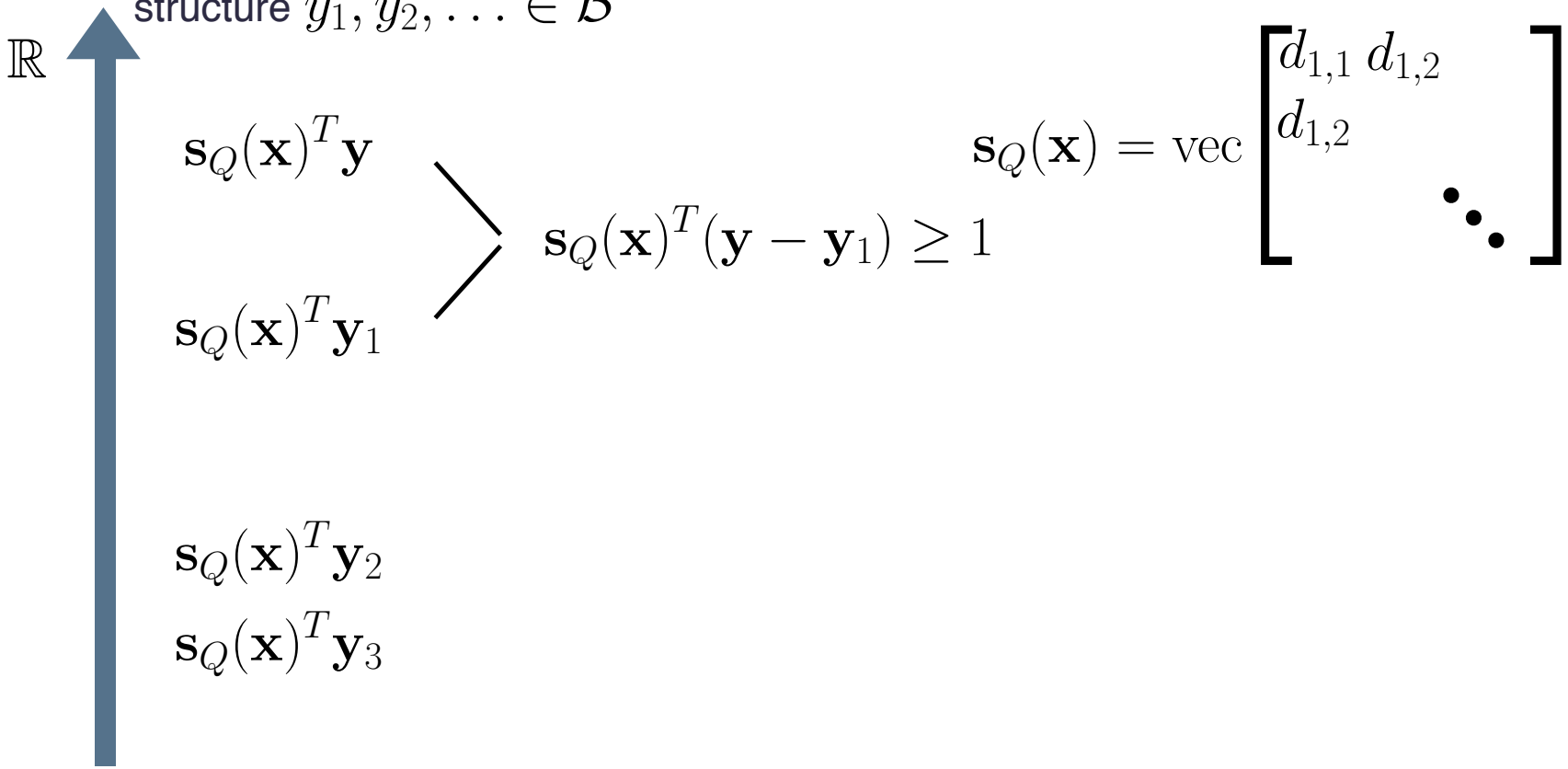


$$\mathbf{s}_Q(\mathbf{x}) = \text{vec}\begin{bmatrix} d_{1,1}\, d_{1,2} & & \\ d_{1,2} & & \\ & & \ddots \end{bmatrix}$$

Maximum-margin

Taskar et al. 2005

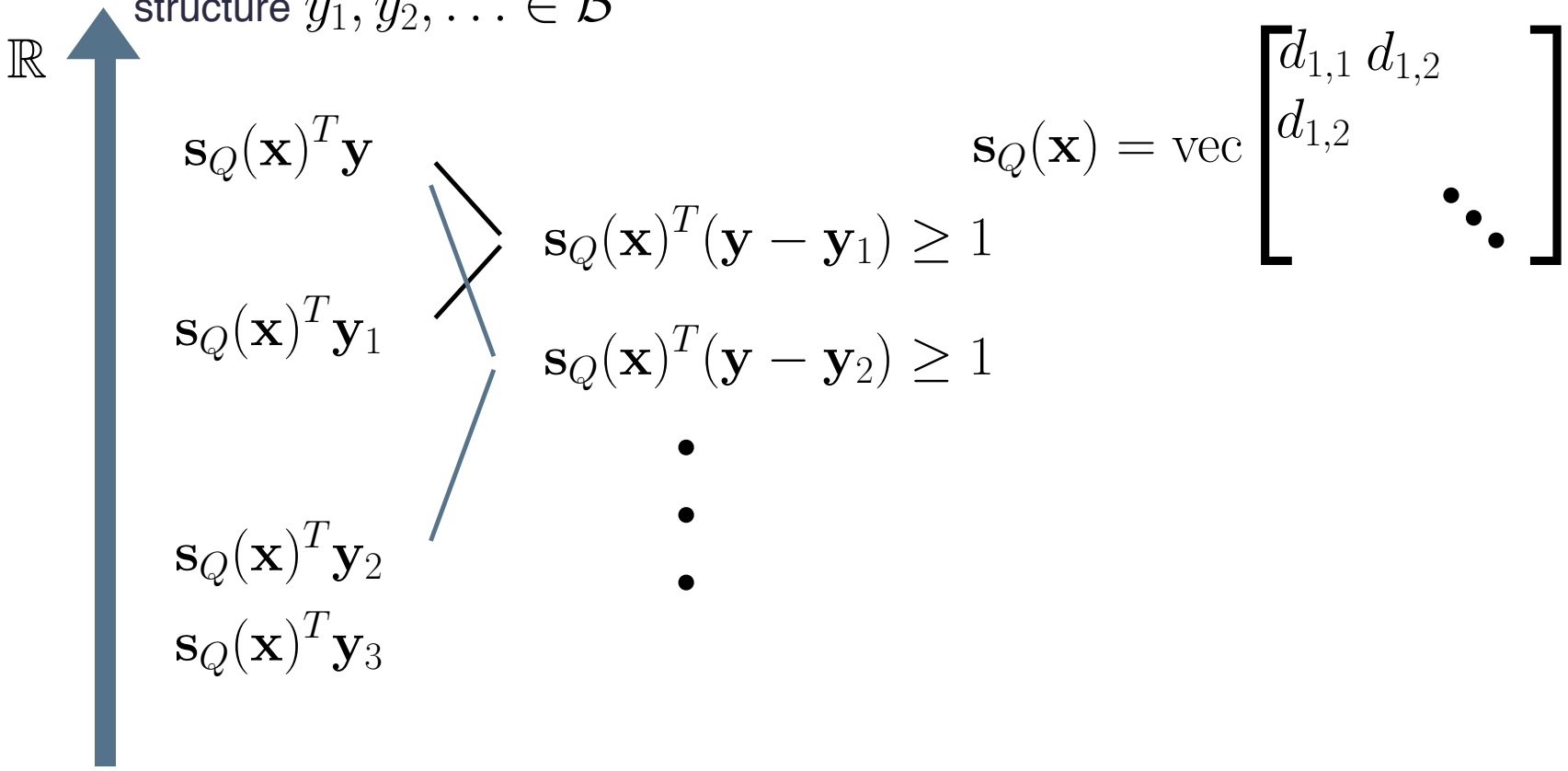
- We use the dual-extragradient algorithm to learn Q
  - Define the margin to be the minimum gap between the predictive values of the true structure $y \in \mathcal{B}$ and each possible alternative structure $y_1, y_2, \ldots \in \mathcal{B}$



Maximum-margin

Taskar et al. 2005

- We use the dual-extragradient algorithm to learn Q
  - Define the margin to be the minimum gap between the predictive values of the true structure $y \in \mathcal{B}$ and each possible alternative structure $y_1, y_2, \ldots \in \mathcal{B}$



Maximum-margin

- You can think of the dual extragradient algorithm as successively minimizing the violation of the gap constraints
- Each iteration focusses on “worst offending network”

1. $\mathbf{y}_{\text{bad}} = \underset{\tilde{\mathbf{y}} \in \mathcal{B}}{\operatorname{argmin}} \, \mathbf{s}_Q(\mathbf{x})^T \tilde{\mathbf{y}}$

Maximum-margin

- You can think of the dual extragradient algorithm as successively minimizing the violation of the gap constraints
- Each iteration focusses on “worst offending network”

1. $\mathbf{y}_{\text{bad}} = \underset{\tilde{\mathbf{y}} \in \mathcal{B}}{\operatorname{argmin}} \, \mathbf{s}_Q(\mathbf{x})^T \tilde{\mathbf{y}}$
2. $Q = Q - \epsilon \dfrac{\partial \text{gap}(\mathbf{y}, \mathbf{y}_{\text{bad}})}{\partial Q}$

Maximum-margin

- You can think of the dual extragradient algorithm as successively minimizing the violation of the gap constraints
- Each iteration focusses on "worst offending network"

1. $\mathbf{y}_{\text{bad}} = \underset{\tilde{\mathbf{y}} \in \mathcal{B}}{\operatorname{argmin}} \, \mathbf{s}_Q(\mathbf{x})^T \tilde{\mathbf{y}}$

2. $Q = Q - \epsilon \dfrac{\partial \text{gap}(\mathbf{y}, \mathbf{y}_{\text{bad}})}{\partial Q}$

$$d_{j,k} = (x_j - x_k)^T Q (x_j - x_k)$$
$$= \langle Q, (x_j - x_k)(x_j - x_k)^T \rangle$$

linear in Q

$$\left(\sum_{jk \in \text{FP}} (x_j - x_k)(x_j - x_k)^T - \sum_{jk \in \text{FN}} (x_j - x_k)(x_j - x_k)^T \right)$$

Maximum-margin

- You can think of the dual extragradient algorithm as successively minimizing the violation of the gap constraints
- Each iteration focusses on "worst offending network"

1. $\mathbf{y}_{\text{bad}} = \underset{\tilde{\mathbf{y}} \in \mathcal{B}}{\operatorname{argmin}} \, \mathbf{s}_Q(\mathbf{x})^T \tilde{\mathbf{y}}$

2. $Q = Q - \epsilon \dfrac{\partial \text{gap}(\mathbf{y}, \mathbf{y}_{\text{bad}})}{\partial Q}$

$$d_{j,k} = (x_j - x_k)^T Q (x_j - x_k)$$
$$= \langle Q, (x_j - x_k)(x_j - x_k)^T \rangle$$

Caveat: this is not the whole story!
Thanks to Simon Lacoste-Julien for help debugging

$$\left(\sum_{jk \in \text{FP}} (x_j - x_k)(x_j - x_k)^T - \sum_{jk \in \text{FN}} (x_j - x_k)(x_j - x_k)^T \right)$$

Experiments

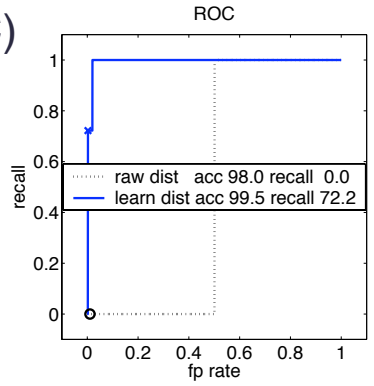
- How does it work in practice?

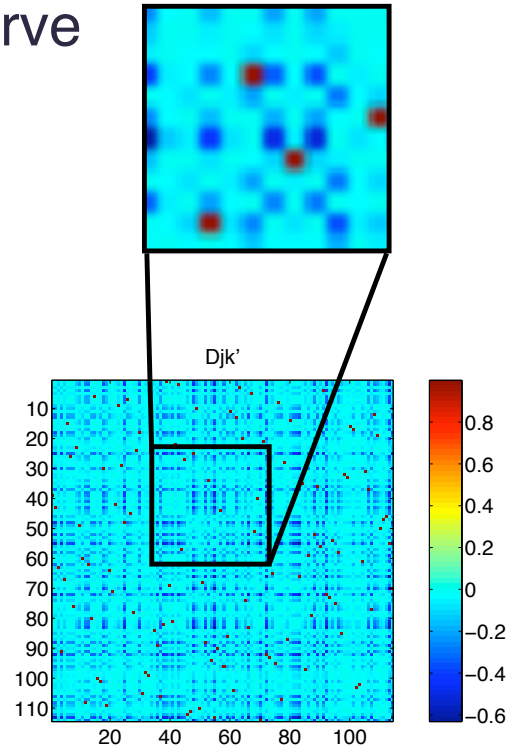
Error metrics for SNP

- Recall & hamming loss (#FP + #FN)
  - Reward the correct structure, but not the distance metric
- We construct a structure-sensitive ROC curve
  - Structure predictions are blended with distances

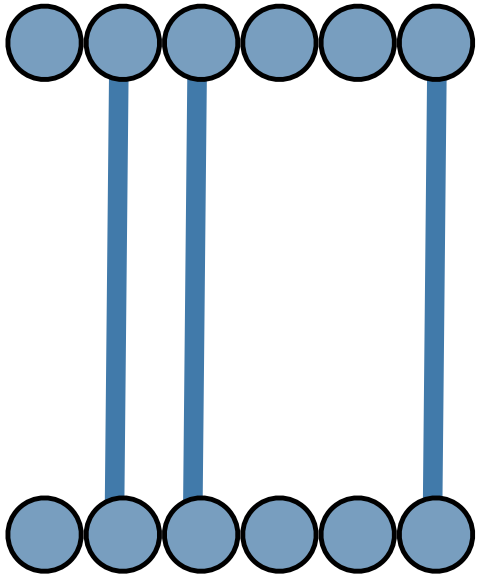
$$\tilde{y}_{j,k} = y_{j,k} + \epsilon \exp(-d_{j,k})$$

- We can now measure
  - Area under the ROC curve (AUC)
  - Recall



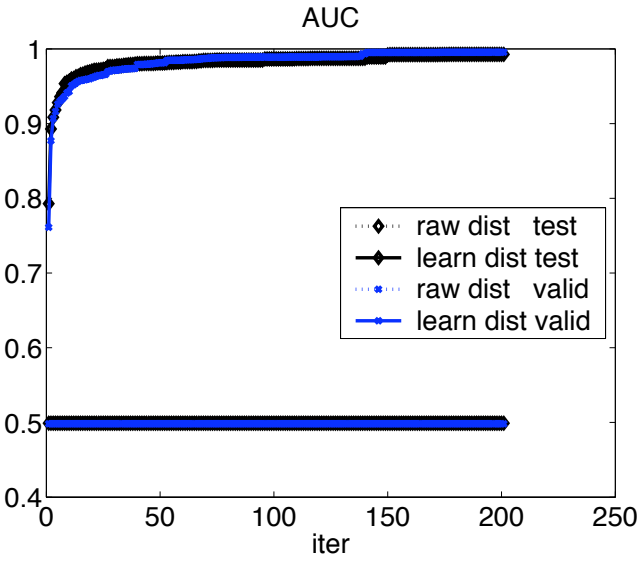


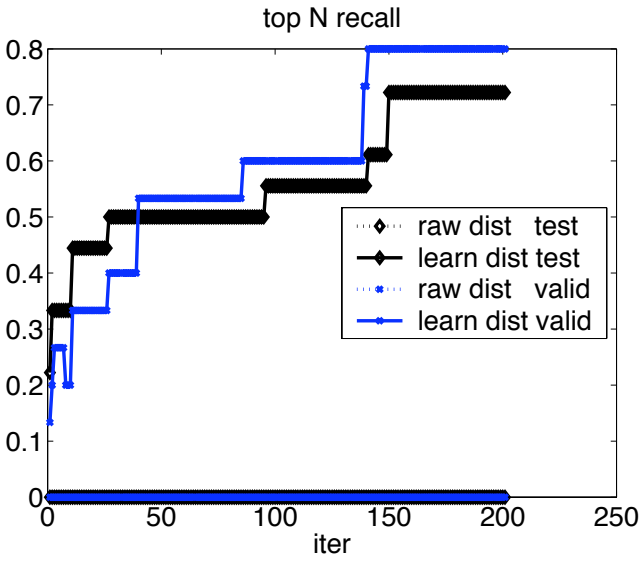
Example

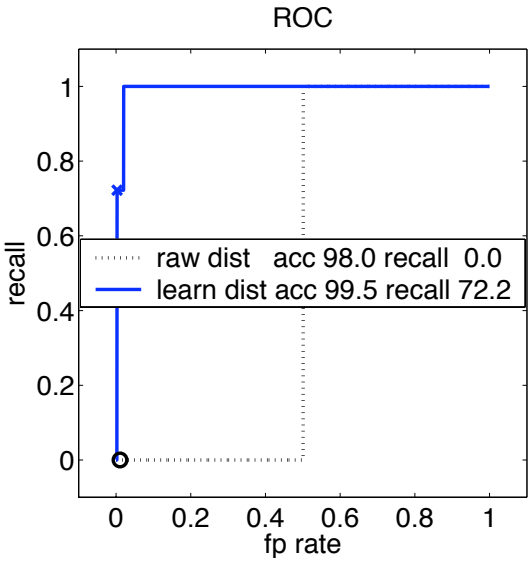


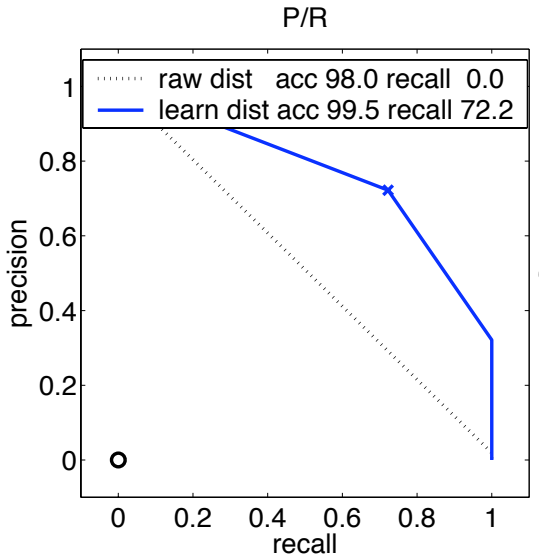
300 nodes in 2D
1-matching structure
X,Y features

Example

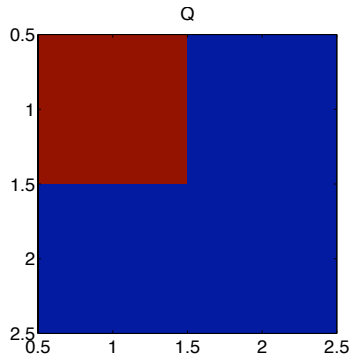

AUC
raw dist test
learn dist test
raw dist valid
learn dist valid
iter


top N recall
raw dist test
learn dist test
raw dist valid
learn dist valid
iter


ROC
raw dist acc 98.0 recall 0.0
learn dist acc 99.5 recall 72.2
recall
fp rate

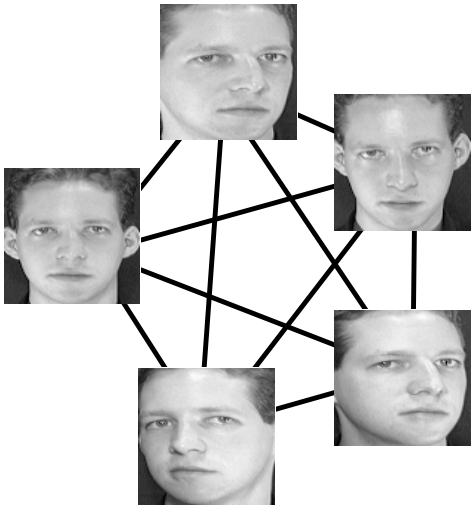

P/R
raw dist acc 98.0 recall 0.0
learn dist acc 99.5 recall 72.2
precision
recall

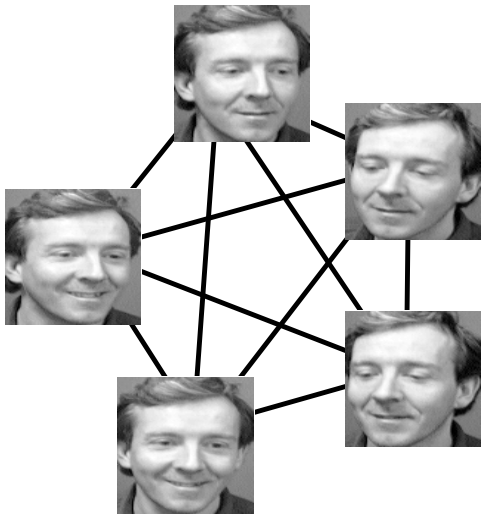
$Q =$

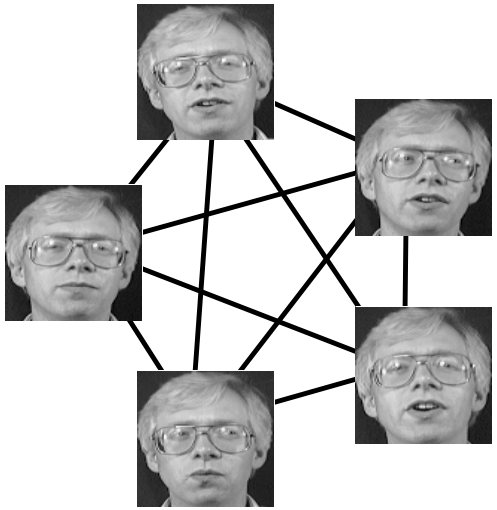

Q

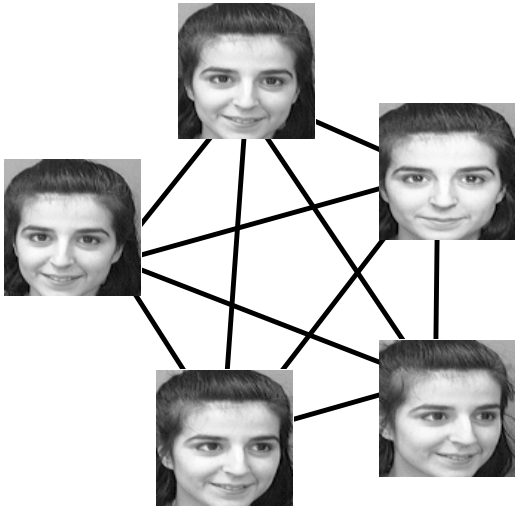
Equivalence networks

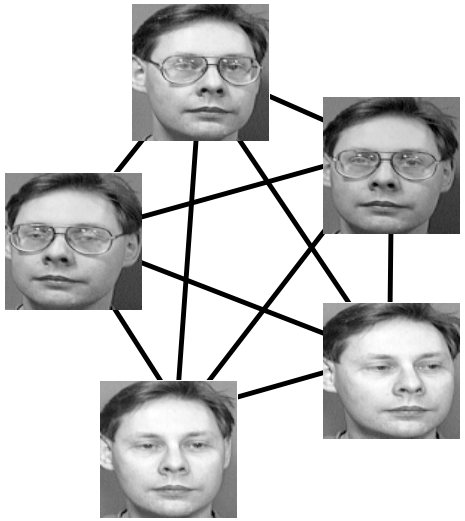
- Olivetti face images





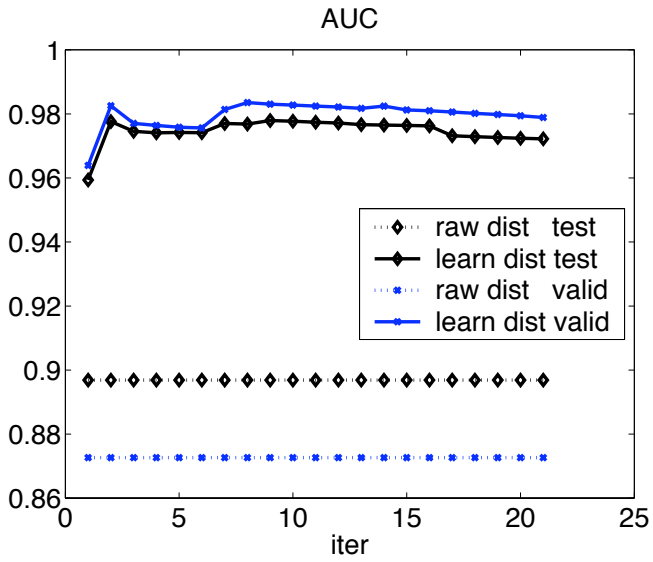


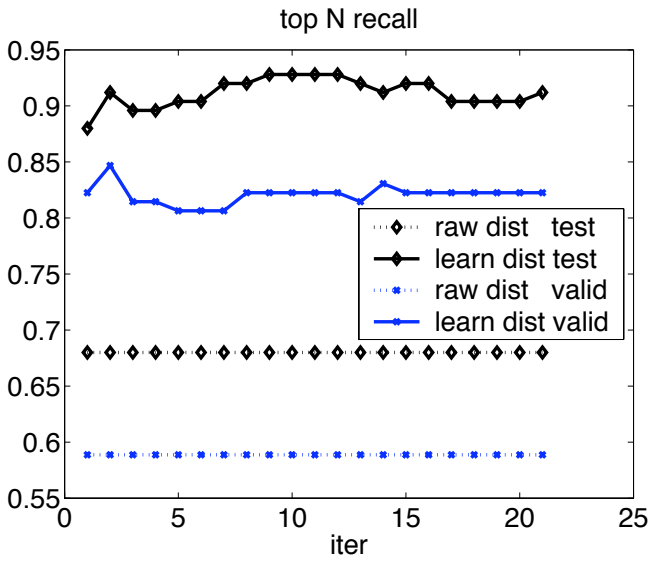


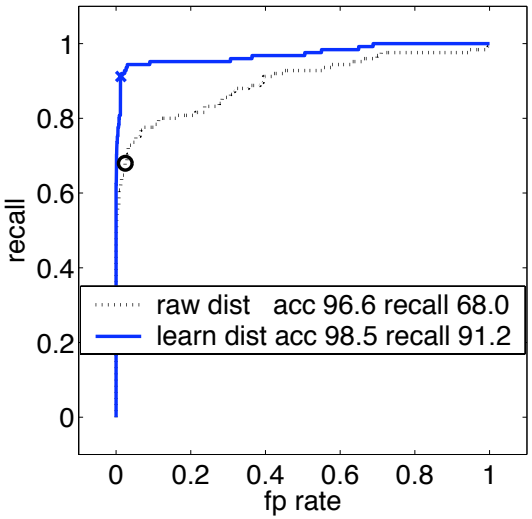


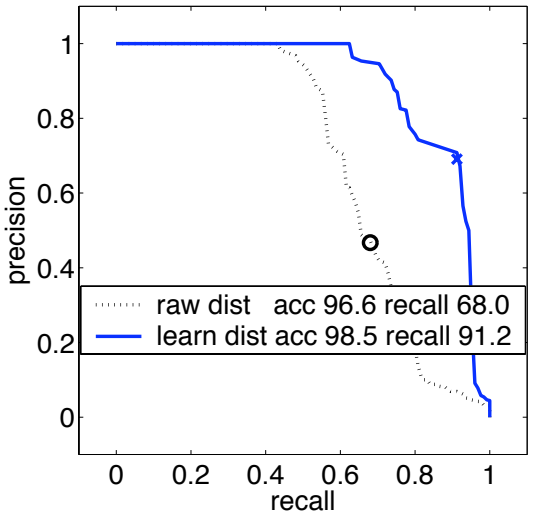
300 images
10 per person
30 PCA features

Olivetti face images

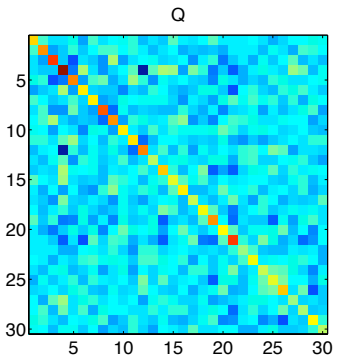








$Q =$



Olivetti face images

Reconstructions of rows of sqrt(Q)

Olivetti face images

Reconstructions of rows of sqrt(Q) - using scaled rows (x8)

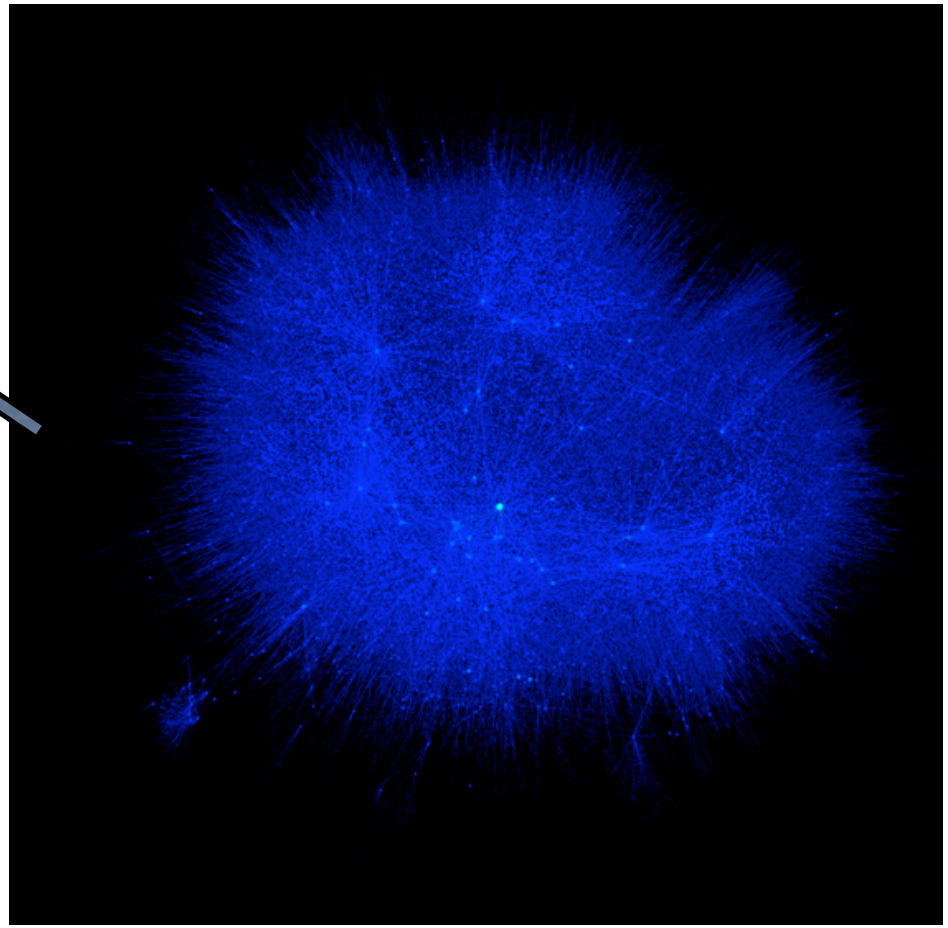
Olivetti face images

Reconstructions of rows of sqrt(Q) - using scaled rows (x1 1)

Olivetti face images

Reconstructions of rows of sqrt(Q) - using scaled rows (x14)

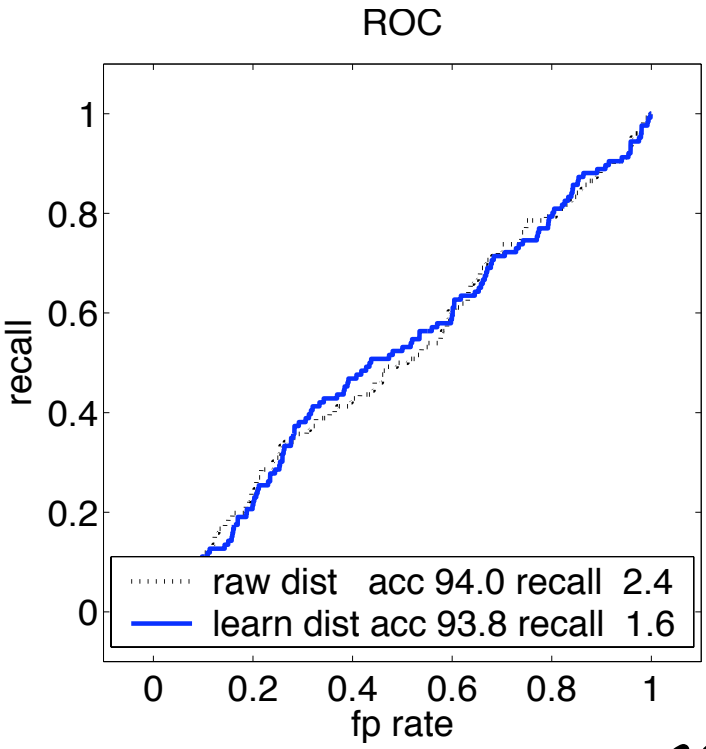
Social network ... and future work



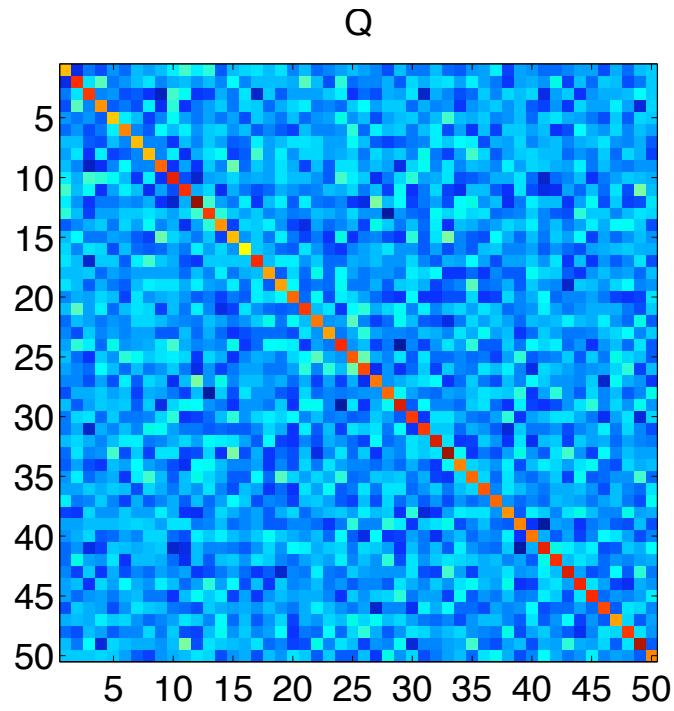
6848 users
"assume" b-matching structure
bag-of-words features
(favorite music, books, etc.)

Jeffrey Heer, Berkeley

Social network ... and future work



$Q =$



300 nodes in 2D
1-matching structure
X,Y features

Future work

- Selecting the parameter b
- Learning and matching to the true degree distribution
- Learning over alternate combinatorial structures such as trees, forests, cliques

Related Work

- Structured output models
  - B. Taskar, S. Lacoste-Julien, and M. I. Jordan “Structured prediction, dual extragradient and bregman projections” NIPS 2005
  - I. Tsochantaridis and T. Joachims and T. Hofmann and Y. Altun “Large Margin Methods for Structured and Interdependent Output Variables” JMLR
  - F. Sha, L. Saul “Large Margin Gaussian Mixture Models for Automatic Speech Recognition” NIPS 2006
- Network reconstruction
  - A. Culotta, R. Bekkerman, and A. McCallum “Extracting social networks and contact information from email and the web” AAAI 2004
  - M. Rabbat, M. Figueiredo, and R. Nowak. “Network inference from co-occurrences” University of Wisconsin 2006
- Network simulation
  - R. Albert and A. L. Barabasi “Statistical mechanics of complex networks”, Reviews of Modern Physics, and many others ...

Related Work

- Distance metric learning
  - J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov “Neighbourhood components analysis”, NIPS 2004
  - E. Xing, A. Ng, M. Jordan, and S. Russell “Distance metric learning, with application to clustering with side-information” NIPS 2003
  - S. Shalev-Shwartz, Y. Singer, and A. Ng “Online and batch learning of pseudometrics” ICML 2004, and many others ...

Conclusions

- We address a novel structured network prediction problem
- We developed a structured output model that uses a structural network priors to make predictions
- We parameterized the model using a Mahalanobis distance metric
- We demonstrated that it is possible to learn a distance suitable for structured network prediction
- The advantage of using a structured output model to predict edges is that we obtain a higher recall for comparable precision / FP rates

Thank you for your attention
Question & comments?