Supervised Learning of Graph Structure

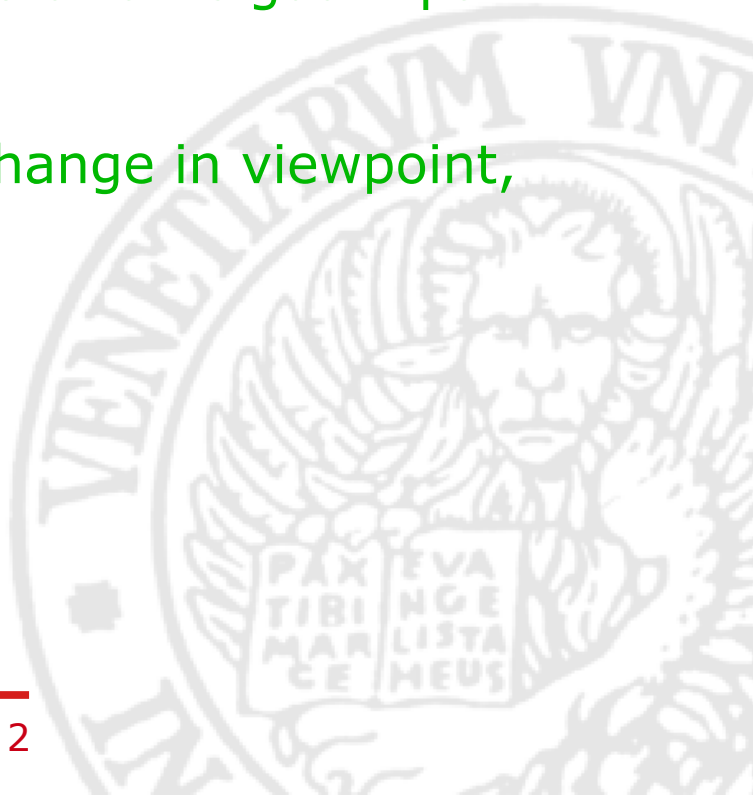
Andrea Torsello, Luca Rossi
Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari Venezia, Italy

Graph-Based Representations

The use of graph representations naturally stems from a number of problems of chemo informatics, proteomics, data mining, computer vision and complex systems

The main advantages over feature-based representations are:

- Capture relational arrangements
- Provide contextual information needed to disambiguate part-identification
- Invariant to transformations (rotation, change in viewpoint, change of scale…)



Difficulties in Graph Learning

Since it is not clear how to convert a graph into a vectorial representation, applying standard pattern recognition techniques is not straightforward

- There is no natural ordering of nodes and edges
  - Correspondences must be used to establish order
- The number of nodes and edges is not fixed
  - Due to noise, occlusion, segmentation errors
- Not easily summarized
  - Since they do not reside in a vector space, mean and covariance are hard to characterize

There have been some successful attempts at embedding graph into vector spaces, but they are not able to characterize the structural variation of the set

Generative Graph Model

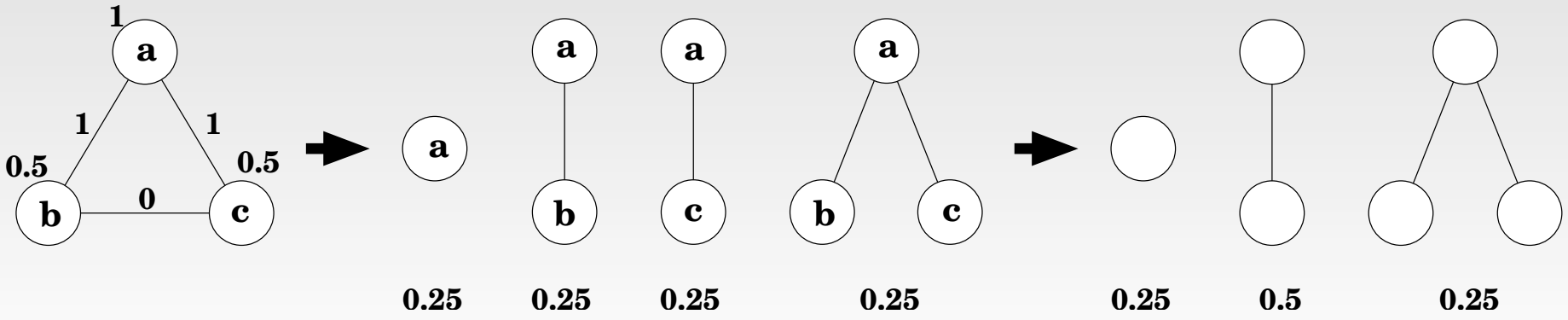
Given a set of undirected graphs **S**, our goal is to learn a generative graph model **G** that can be used to describe the distribution of structural data and characterize the structural variation of the set

We assume that the model is a mixture of naïve models where observations of nodes and edges are independent of the others

The naïve graph model is composed of

- A structural part G=(V,E)
  - Where V are all the nodes that can be generated by the graph and $E \subseteq V \times V$ is the set of possible edges
- A stochastic part that encodes the variability in the observed graphs

Generative Graph Model



θ_i probability of generating node I

T_{ij} probability of generating node (i,j), conditioned on the generation of both i and j

W^n_i and W^e_{ij} generative models for the nodes and edges attributes

Note that θ_i and W^n_i need not to be independent (nor do T_{ij} and W^e_{ij})

External Nodes Generation

To allow the addition of nodes we must change the model a bit

Assume

- there is a core model that we describe fully as previously
- nodes can be generated outside of the core model
- the probability of generating a graph with k external nodes is $(1-\bar{\theta})\,\bar{\theta}^{k}$
- the probability of observing an edge connecting an external node is $\bar{\tau}$
- the attribute models $\overline{W}^{n}$ and $\overline{W}^{e}$ for the nodes and edges respectively

External nodes represent isotropic background noise

Probability of Observing a Graph

After we sample a graph *g* from G, we lose track of the correspondences between the sample's nodes and the nodes of the model

- A random permutation is applied to the nodes of the sample

Hence the observation probability of a graph depends on this unknown set of correspondences.

In particular, given the set of correspondences σ between nodes in G and nodes in *g*, the probability of observing *g* from G is

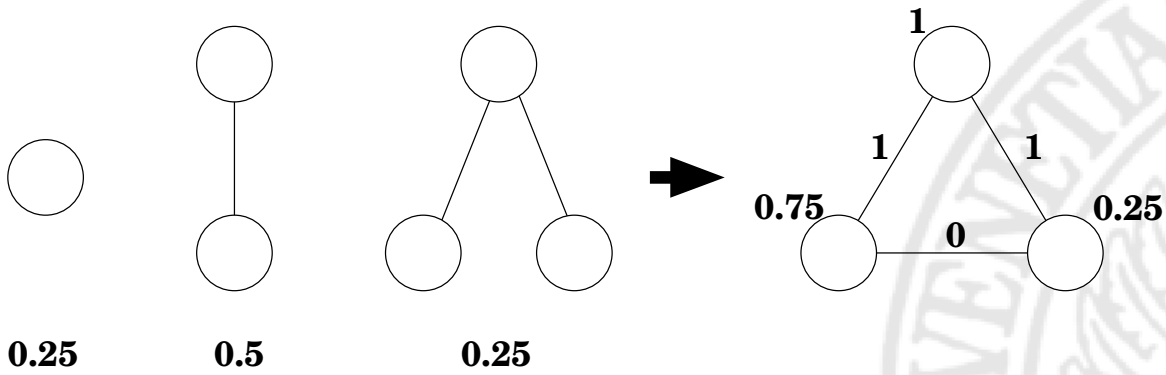
$$P(g|\mathcal{G},\sigma_g) = (1-\bar{\theta}) \prod_{i\in V} P(g_{\sigma_g^{-1}(i)}|\theta_i,\omega_i^n) \cdot \prod_{(i,j)\in E} P(g_{\sigma_g^{-1}(i),\sigma_g^{-1}(j)}|\tau_{i,j},\omega_{i,j}^e) \cdot$$
$$\cdot \prod_{i\notin V} P(g_{\sigma_g^{-1}(i)}|\bar{\theta},\bar{\omega^n}) \cdot \prod_{(i,j)\notin E} P(g_{\sigma_g^{-1}(i),\sigma_g^{-1}(j)}|\bar{\tau},\bar{\omega^e})$$

Estimating the Correspondences

Typically the correspondences are estimated using graph matching techniques. This is equivalent to say that

$$P(g|\mathcal{G}) = \max_{\sigma \in \Sigma_n} P(g|\mathcal{G}, \sigma)$$

However, maximum likelihood estimation of correspondences introduces **bias** in the model estimation



Overcoming the Bias

An alternative approach is to take the expectation over all the possible correspondences

$$P(\hat{g}|\mathcal{G}) = \sum_{\sigma \in \Sigma_n^m} P(g|\mathcal{G}, \sigma) P(\sigma) = \frac{1}{|\Sigma_g|} \sum_{\sigma \in \Sigma_n^m} P(g|\mathcal{G}, \sigma)$$

Averaging over all possible correspondences is not possible due to the super-exponential growth of the space

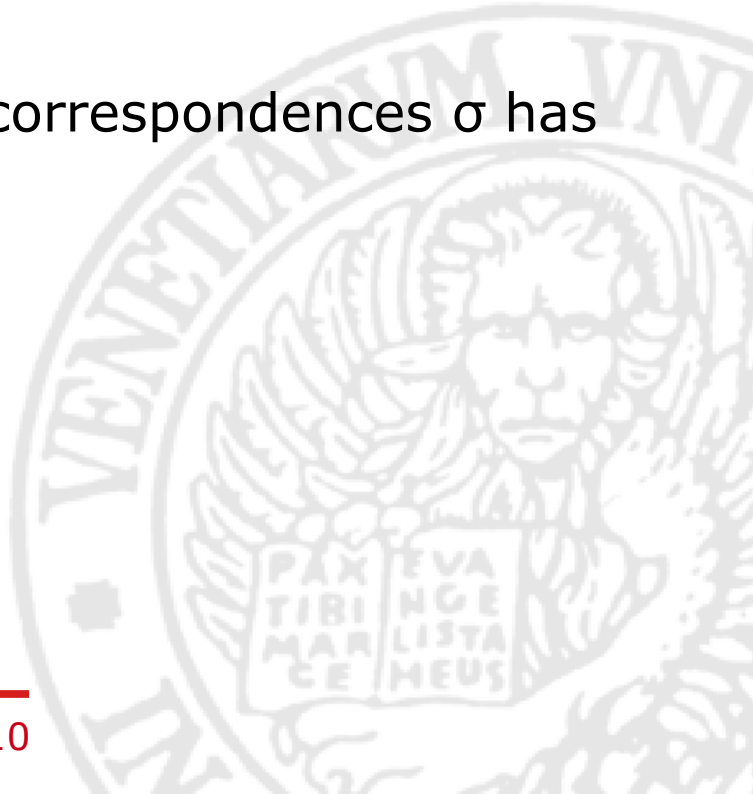
We have to resort to an estimation approach: importance sampling approach to compute a fast-converging estimate of P(g|G) [Torsello 2008]

Correspondence Sampler

We can sample a correspondence as follows

- We start from an initial guess of the correspondences matrix M
- We sample the correspondence for model node i picking a node j with probability m_{ij}
- Then, we condition M to the current match by taking into account the structural information between the sampled node and all the others

The process is iterated until a complete set of correspondences σ has been sampled



Updating the Model

Once correspondences are to hand, we can use maximum likelihood to estimate the parameters of the node, edge and attributes models

- By construction we need only to maximize the node and edge models independently, ignoring what is going on in the rest of the graph

Performance of the sampler critically dependent on M

- m_{ih} is equal to the probability that model node i generates the attributes of graph node h

- After first round of sampling, update M

$$\bar{M}' = \frac{1}{\sum_\sigma \frac{P(\sigma|g,\mathcal{G})}{P(\sigma)}} \sum_\sigma \frac{P(\sigma|g,\mathcal{G})}{P(\sigma)} M_\sigma$$

Learning the Model

We can estimate the model as follows

- Initialize the model randomly
- Repeat n_{iter} times
  - Repeat n_{σ} times)
    - Sample correspondences σ
    - Add nodes and edges observation
  - Update the parameters

Further, single models are too restrictive, so we want to learn **mixtures** of simple graph models. Starting from an initially oversized model, we prune the number and size of the mixture components until we get an optimal model

We follow [Torsello and Dowe 2008] in adopting a MML approach to guide the model pruning

Minimum Message Length Criterion

Simplicity is formalized as the joint cost of

- describing a probabilistic model for the data
- describing the data given the model

We choose the model that corresponds to the shortest two-part message

$$I_1 = \frac{D}{2}\log\left(\frac{|S|}{2\pi}\right) + \frac{1}{2}\log(\pi D) - 1 - \sum_{g\in S}\log\left(P(g|\mathcal{G},\sigma_g)\right)$$

We greedily choose the pruning action (if any) that maximizes the reduction in message length

In order to compute the reduction in message length incurred by removing a node, while sampling the correspondences we compute the matching probability also of the models obtained from the current one with any singe node removal

Modeling the Attributes

For the node-attributed graphs, we adopted the rectified Gaussian model used in [Torsello and Hancock 2006]

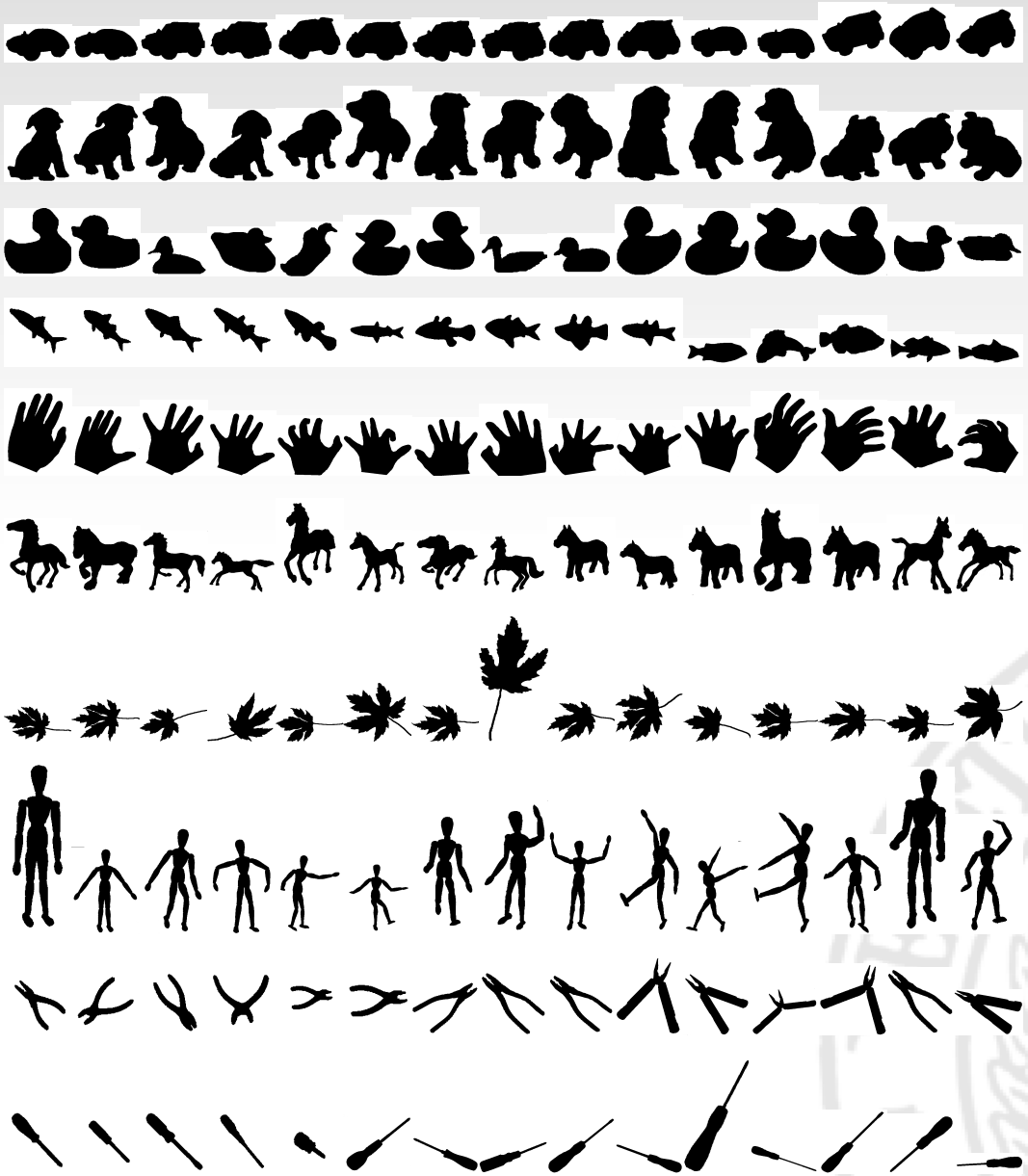
- single stochastic node observation model X_i (normally distributed)
- when sampling the i_{th} node, if $x_i \geq 0$ the node is observed with weight $w_i = x_i$, otherwise the node is not present in the sampled graph

Hence the node is observed with probability $\theta_i = 1\text{-erfc}(\mu_i/\sigma_i)$ with

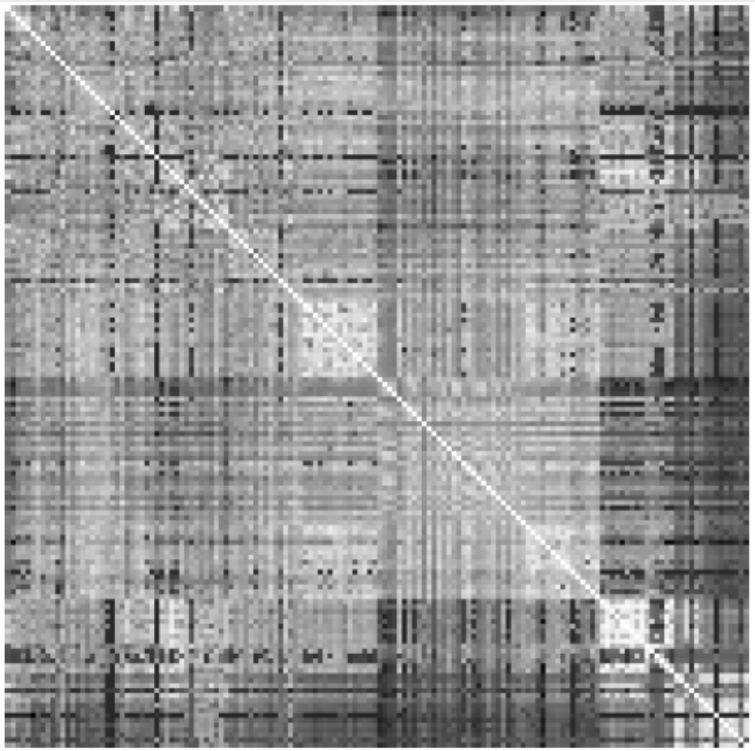
$$\operatorname{erfc} = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}s^2\right) ds$$

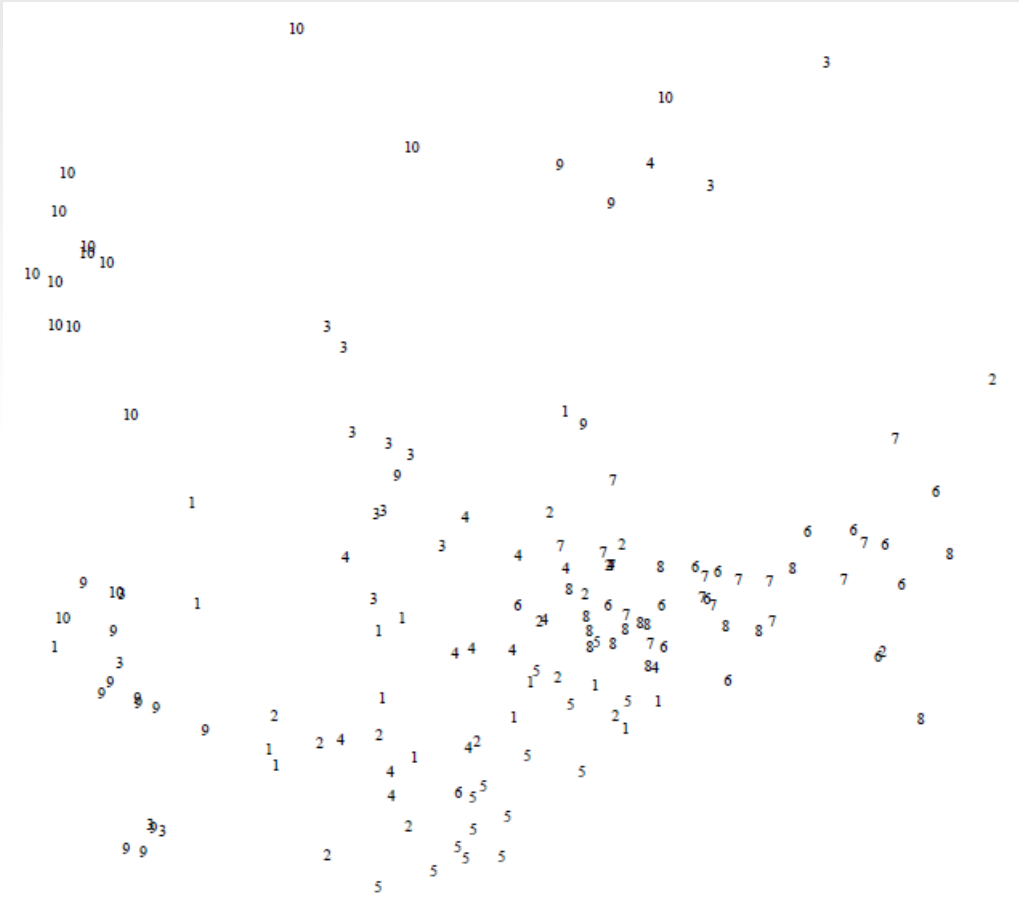
On the other hand, for the edges we use a combination of two independent Bernoulli and Gaussian models: a Bernoulli process establishes whether the edge is observed, and if it is the weight is drawn according to an independent Gaussian variable

Shock Graphs

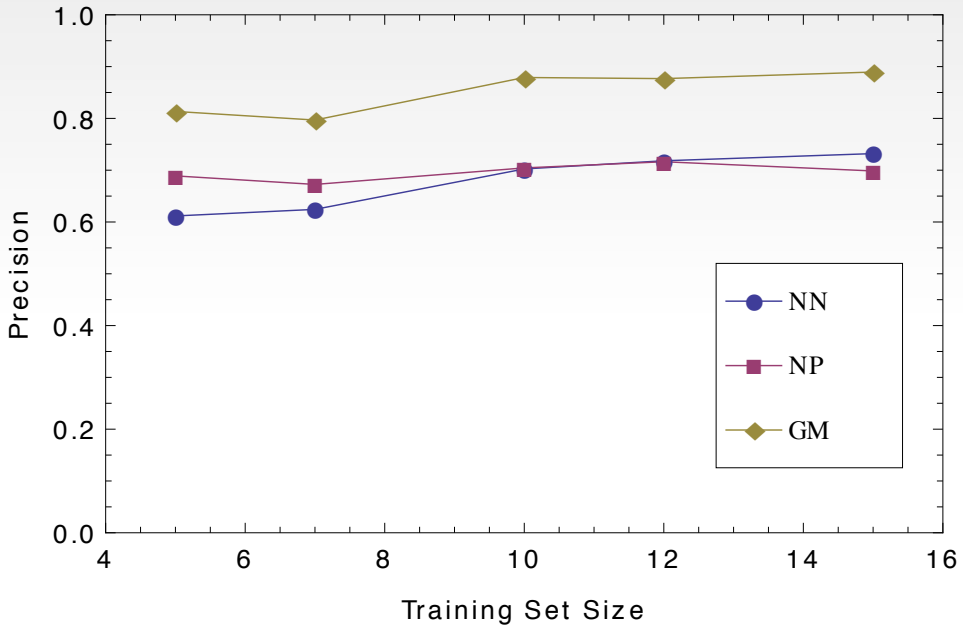


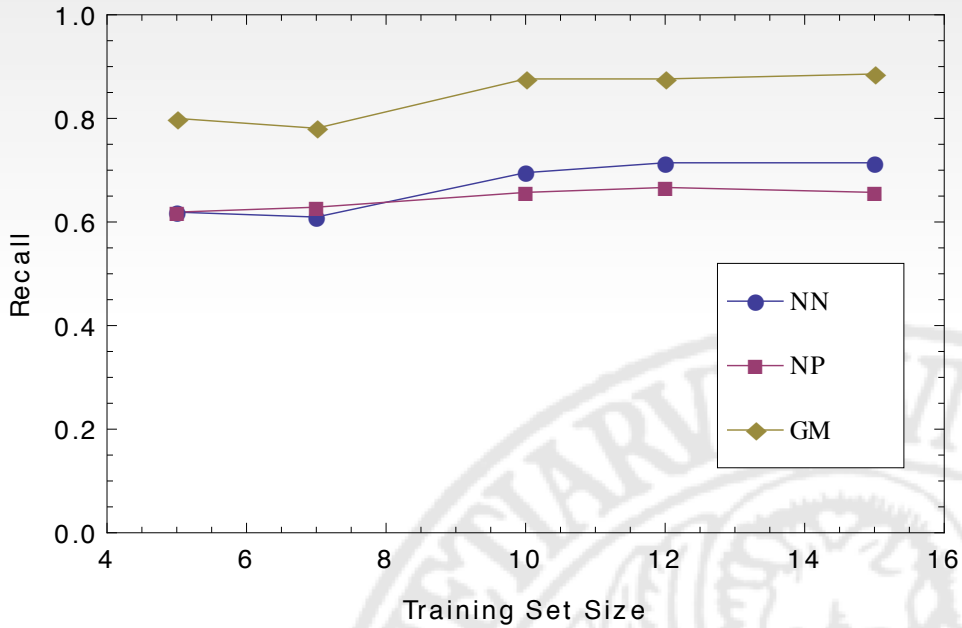
Shock Graphs: Distance Matrix





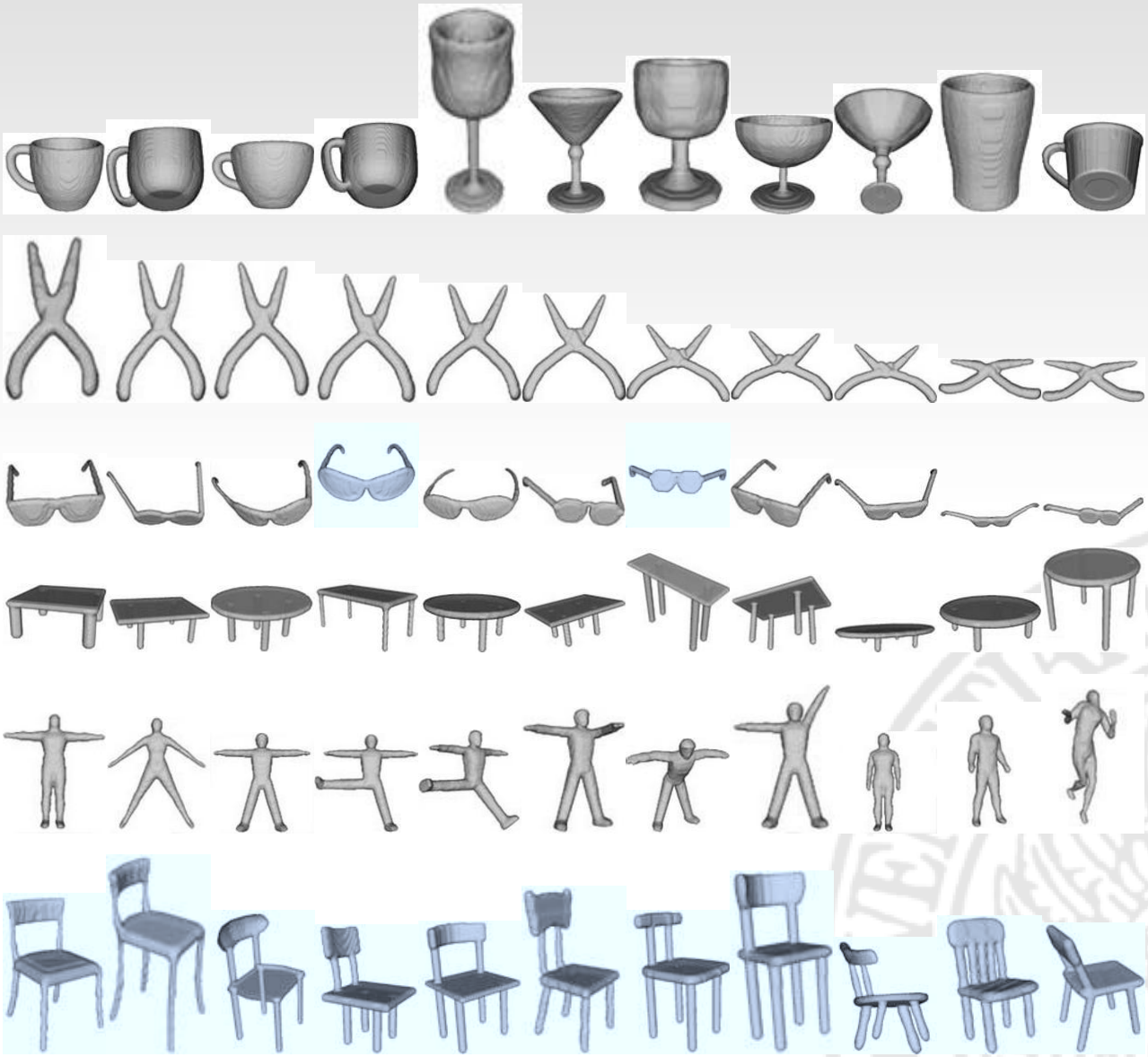
Shock Graphs: Precision and Recall



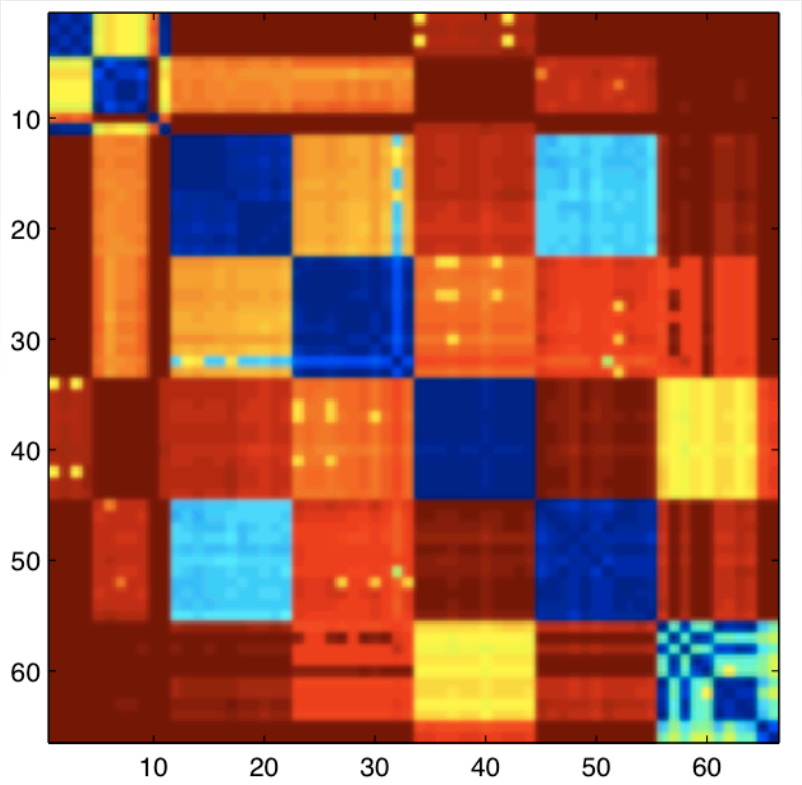


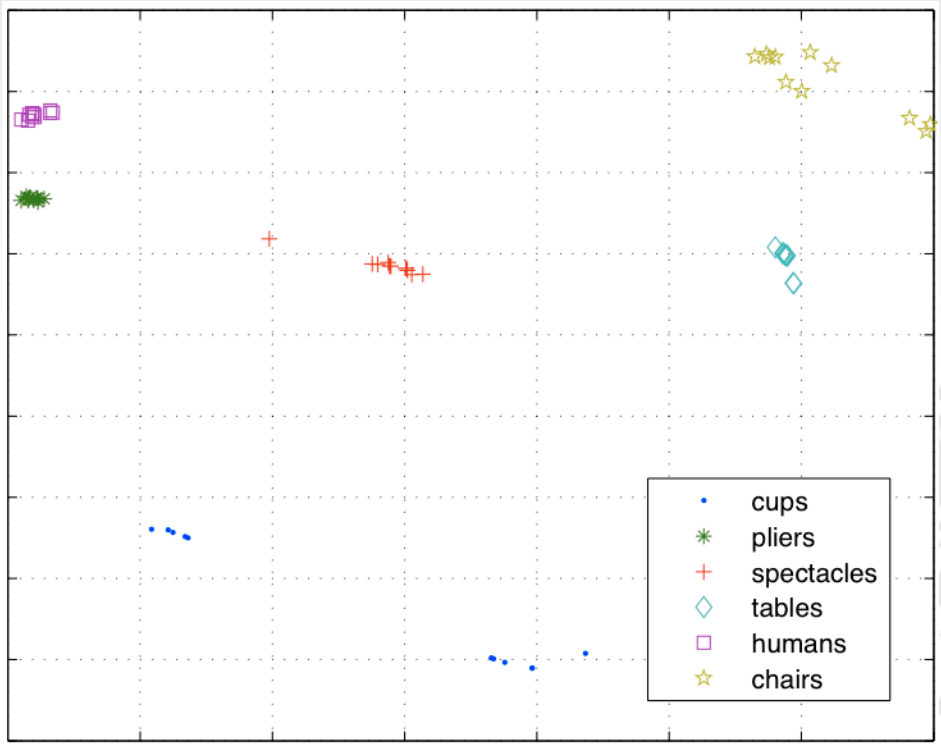
Training samples

3D Shapes

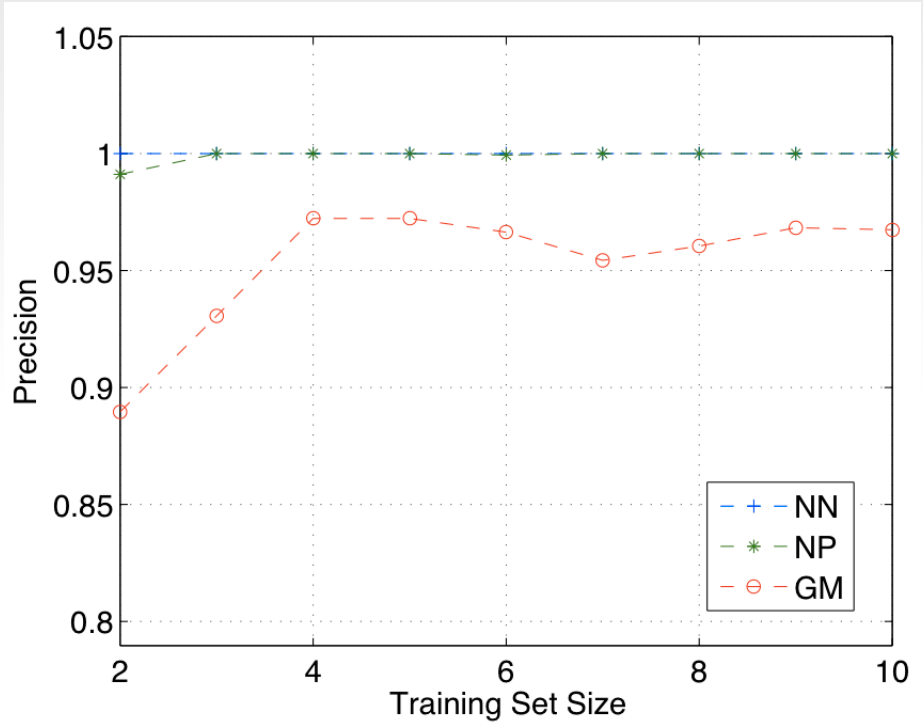


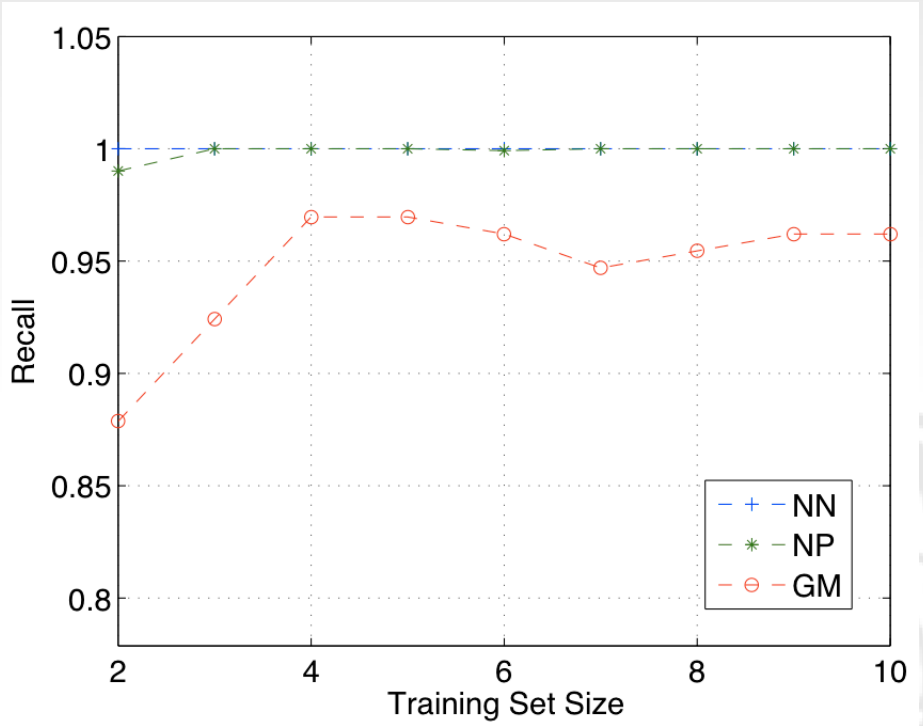
3D Shapes: Distance Matrix



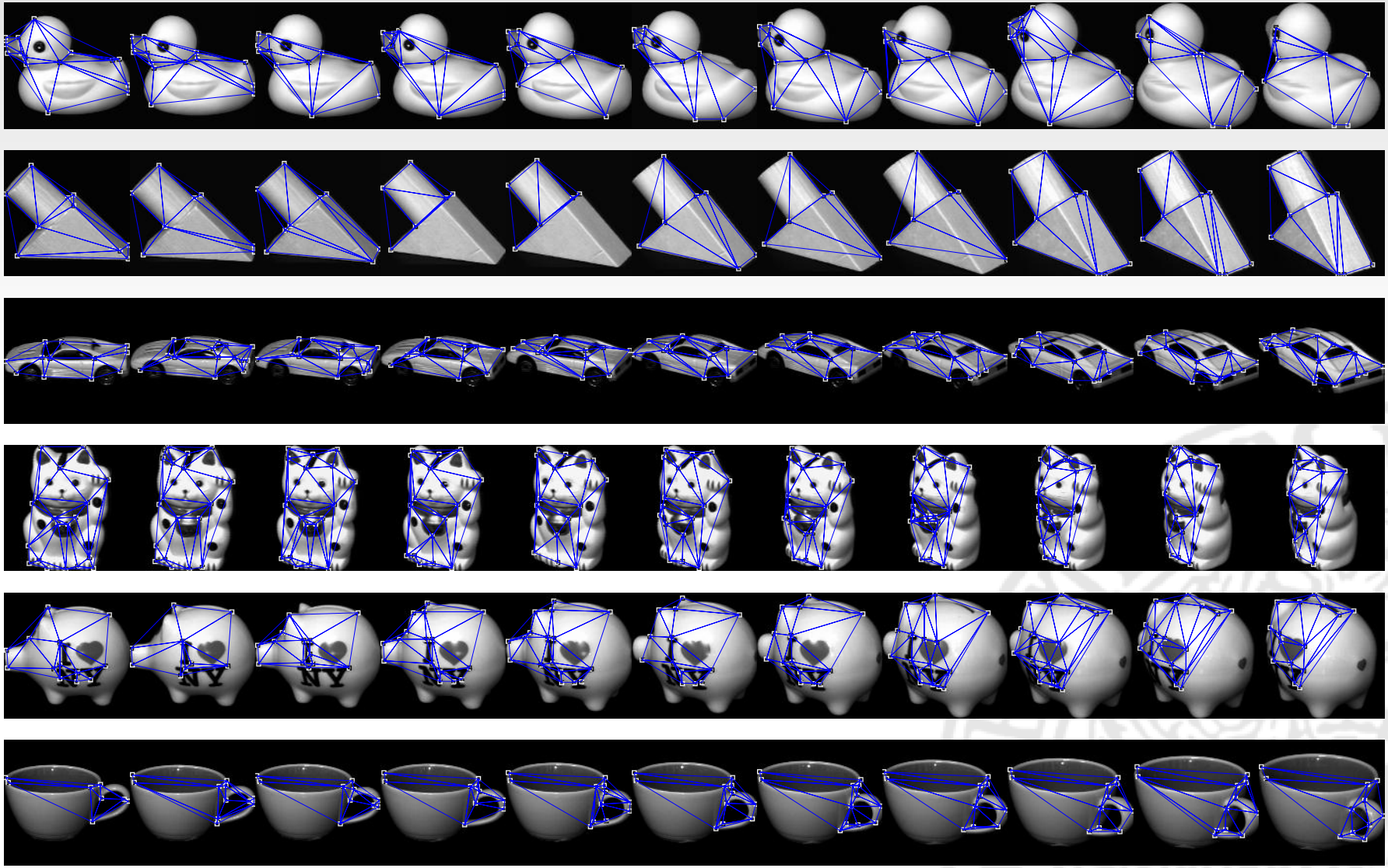


3D Shapes: Precision and Recall

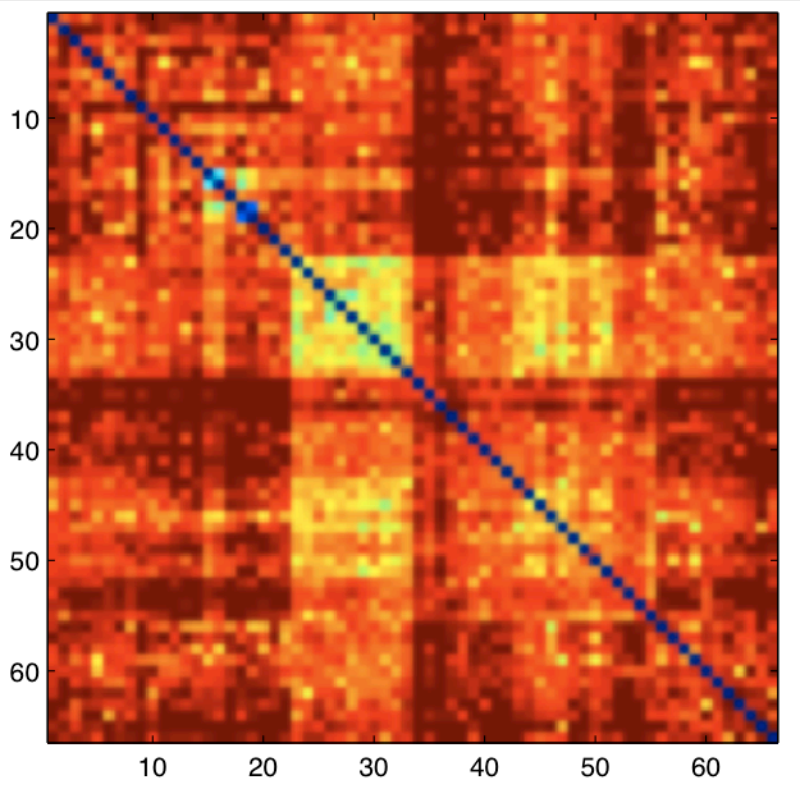


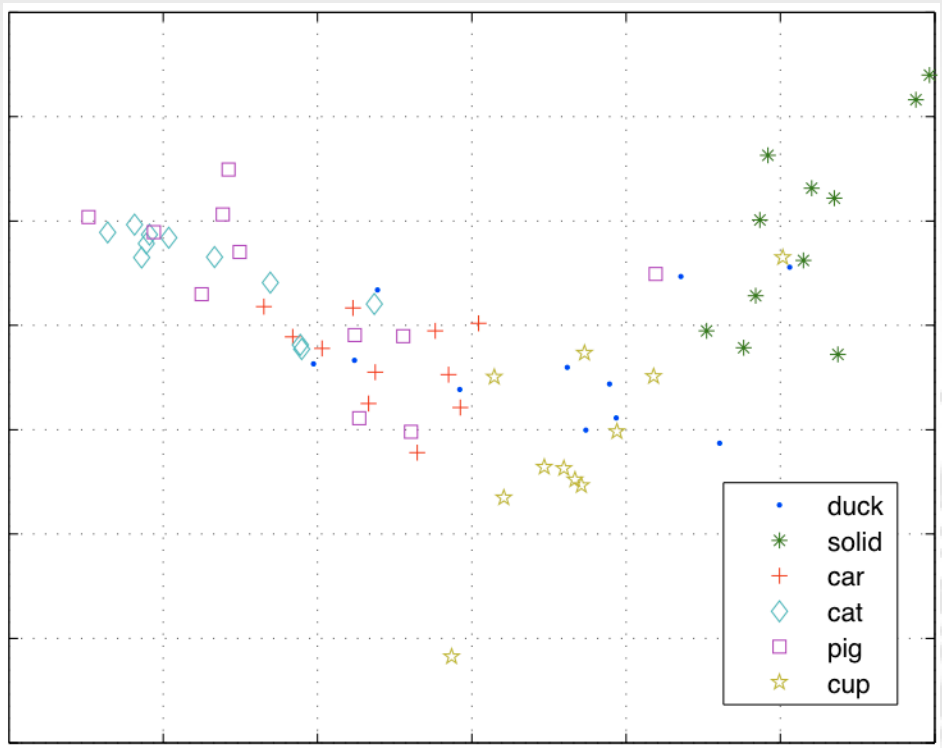


COIL-20

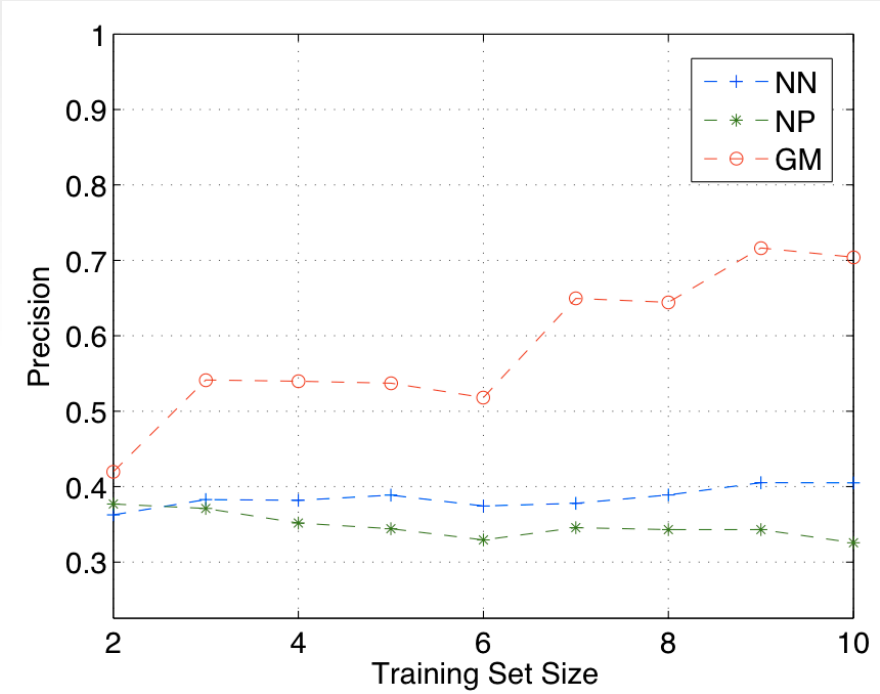


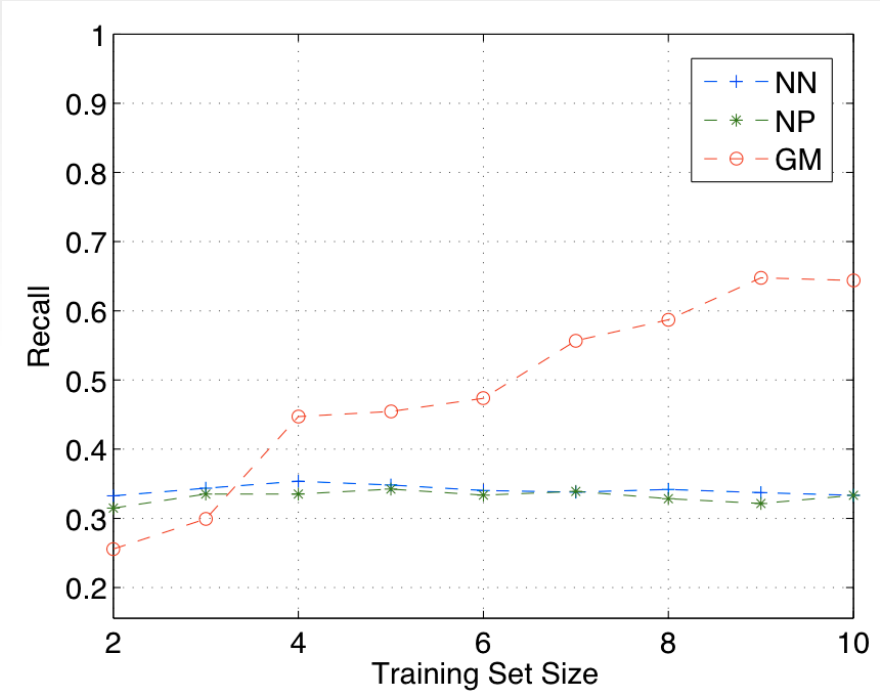
COIL-20: Distance Matrix



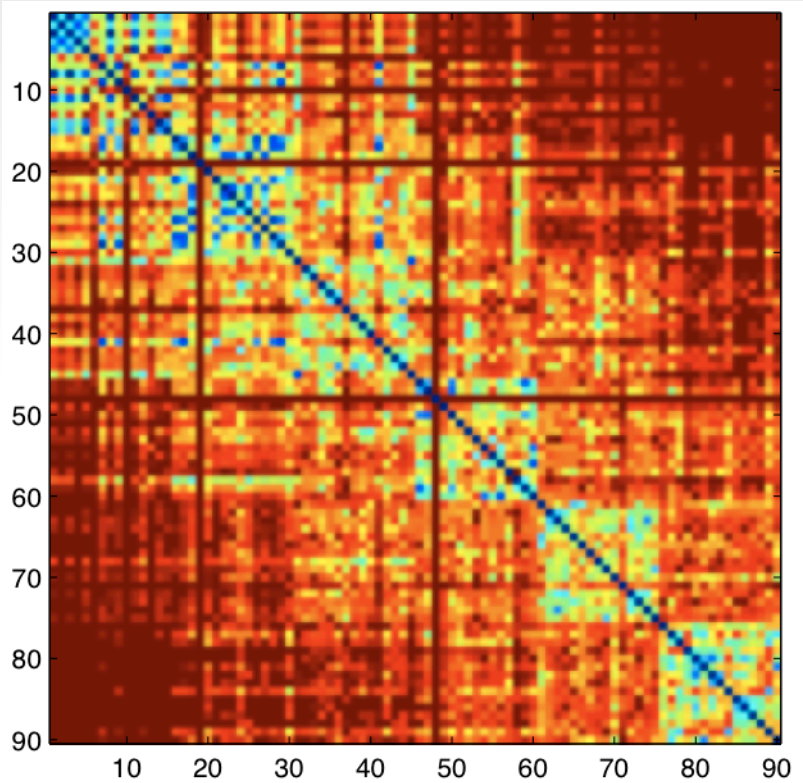


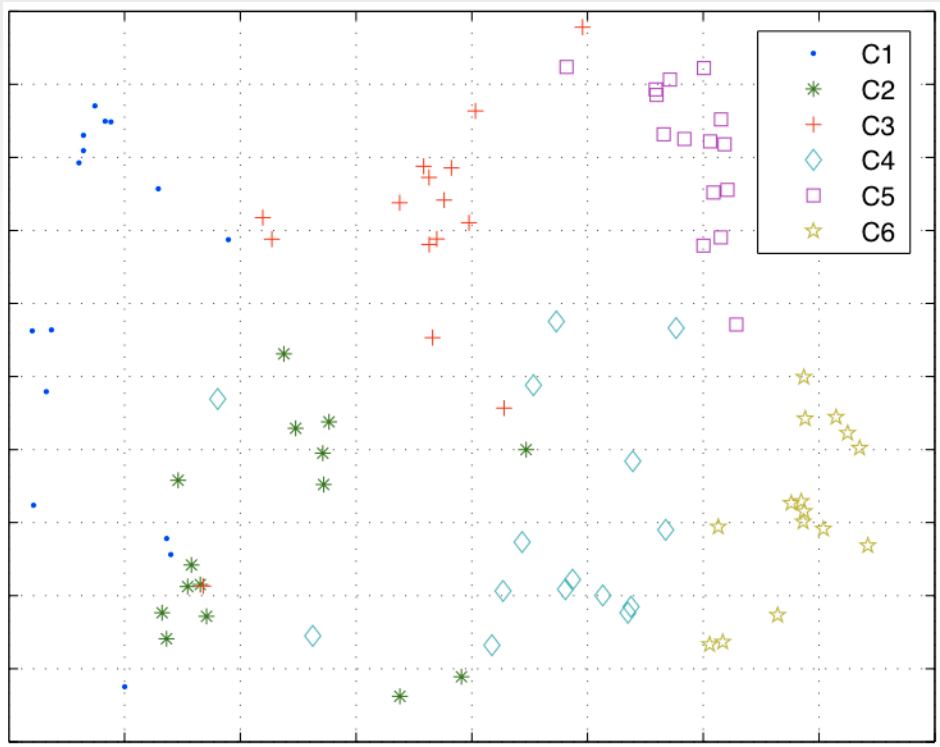
COIL-20: Precision and Recall



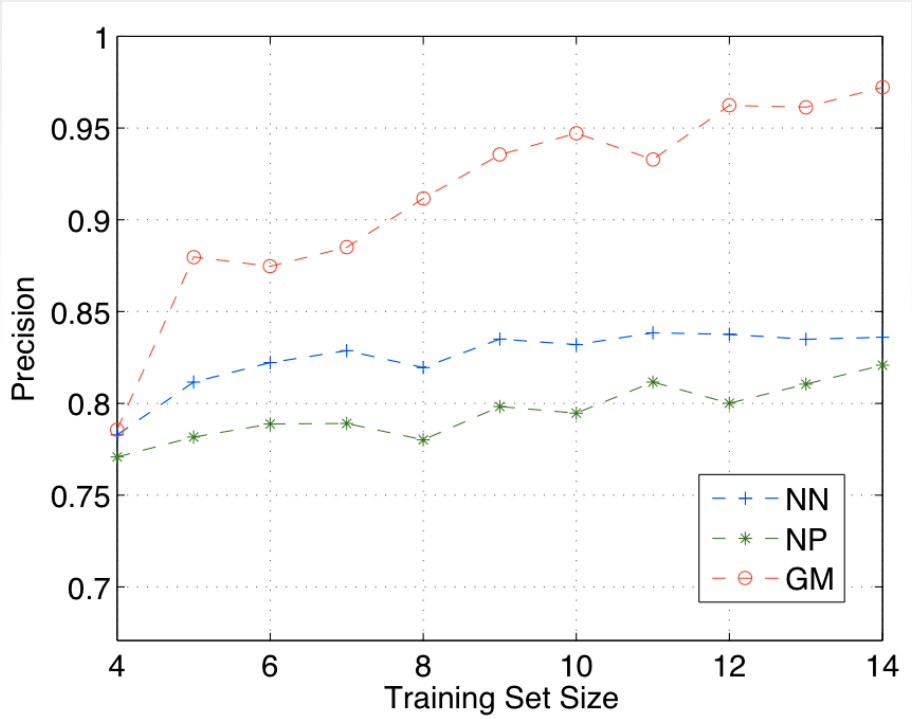


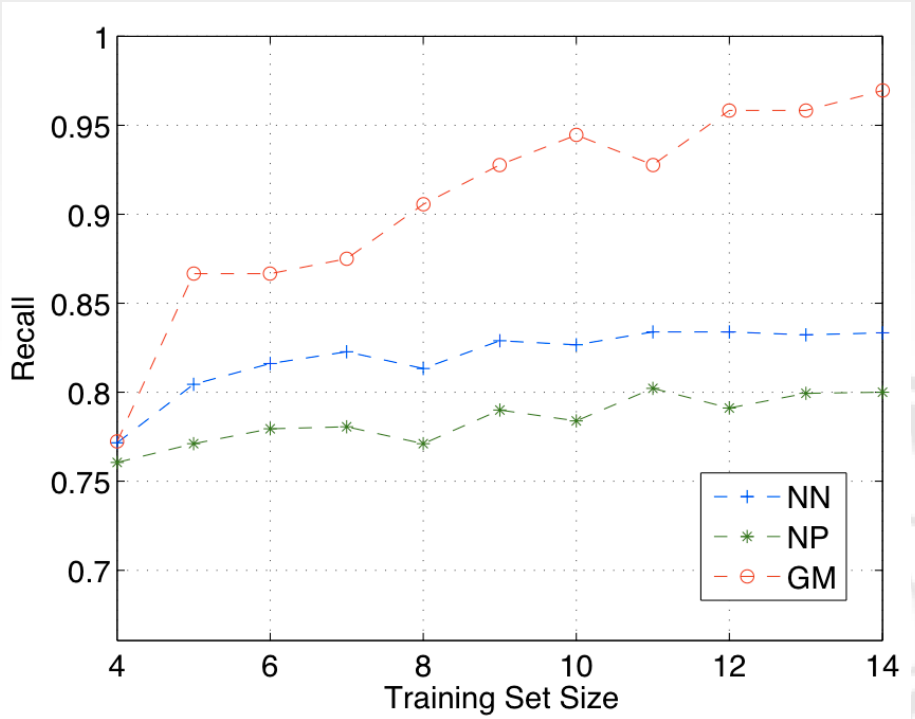
Synthetic Data: Distance Matrix





Synthetic Data: Precision and Recall





Conclusions

We have addressed the problem of learning a generative model for graphs from samples

The model is based on a naïve node independence assumptions, but mixes such simple models in order to capture node correlation

Fast sampling approach to overcome the bias of a single correspondence estimation, ML estimate of node/edge parameters and MML criterion for model selection

Experiments performed on a wide range of real world object recognition tasks as well as on synthetic data show that our approach clearly outperforms both the NN and NP rules regardless of the matching algorithm and the distance metric adopted

References

[Torsello 2008] A. Torsello. "An Importance Sampling Approach to Learning Structural Representations of Shape." In IEEE CVPR, 2008

[Torsello and Hancock 2006] A. Torsello and E. R. Hancock, "Learning Shape-Classes Using a Mixture of Tree-Unions." IEEE Trans. Pattern Anal. Machine Intell., 28(6):954-967, 2006

[Torsello and Dowe 2008] A. Torsello, D. Dowe, "Learning a generative model for structural representations." In 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence AI-08, LNAI 5360, pp. 573–583, 2008.