Information Theoretic Methods for Learning Generative Models of Relational Structure

Lin Han, Edwin R. Hancock and Richard C. Wilson
Department of Computer Science
The University of York

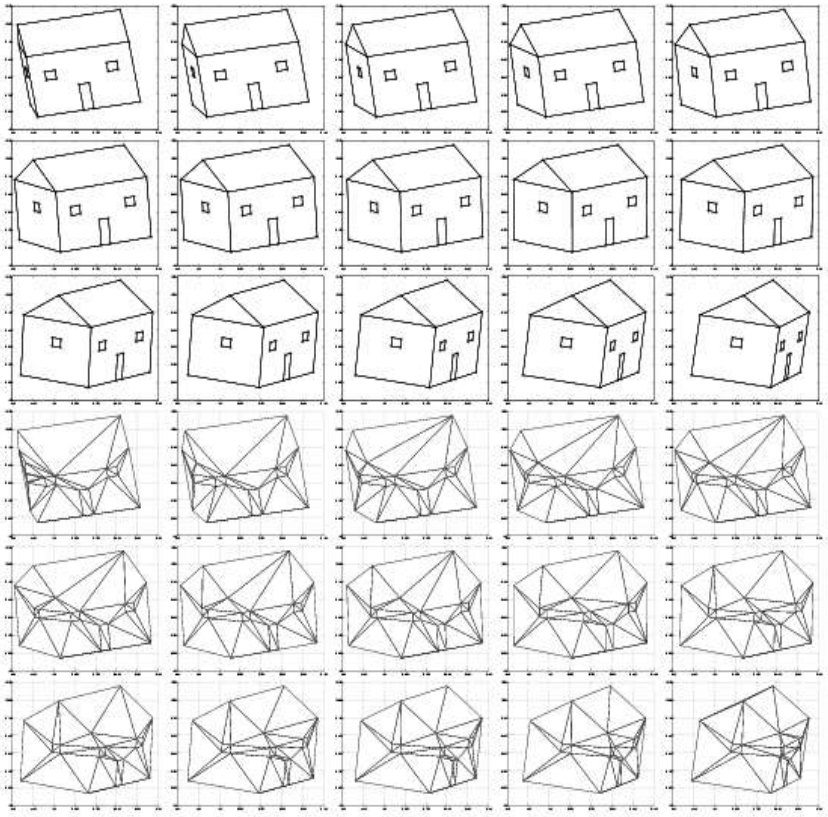
THE ROYAL SOCIETY

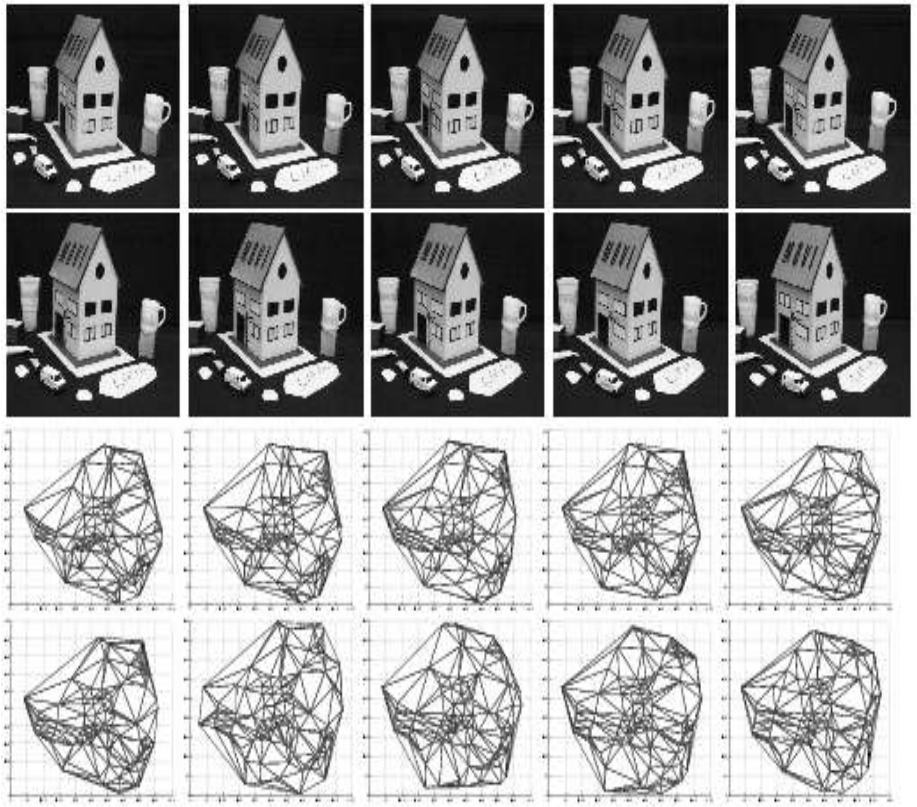
Graph representations

- Advantages: can capture object or scene structure in a manner that is invariant to changes in viewpoint. Abstract scene contents in an efficient way.

- Disadvantages: can be fragile (sensitive to noise and segmentation error). Available pattern recognition/machine learning methodology limited.

Graph Representations from images





Learning with graphs

- Cluster similar objects, and represent them using a class prototype (e.g. median) using (dis) similarities.
- Extract features and perform central clustering.
- Construct a generative model to capture distribution of structural variations using probability distributions.

….. is difficult because

- Graphs are not vectors: There is no natural ordering of nodes and edges. Correspondences must be used to establish order.

- Structural variations: Numbers of nodes and edges are not fixed. They can vary due to segmentation error.

- Not easily summarized: Since they do not reside in a vector space, mean and covariance hard to characterise.

Generative Models

- Structural domain: define probability distribution over prototype structure. Prototype together with parameters of distribution minimise description length (Torsello and Hancock, PAMI 2007) .

- Spectral domain: embed nodes of graphs into vector-space using spectral decomposition. Construct point distribution model over embedded positions of nodes (Bai, Wilson and Hancock, CVIU 2009).

Deep learning

- Deep belief networks (Hinton 2006, Bengio 2007).
- Compositional networks (Amit+Geman 1999, Fergus 2010).
- Markov models (Leonardis 200
- Stochastic image grammars (Zhu, Mumford, Yuille)
- Taxonomy/category learning (Todorovic+Ahuja, 2006-2008).

Aim

- Combine spectral and structural methods.
- Use description length criterion.
- Apply to graphs rather than trees.

Prior work

- IJCV 2007 (Torsello, Robles-Kelly, Hancock) –shape classes from edit distance using pairwise clustering.
- PAMI 06 and Pattern Recognition 05 (Wilson, Luo and Hancock) – graph clustering using spectral features and polynomials.
- PAMI 07 (Torsello and Hancock) – generative model for variations in tree structure using description length.
- CVIU09 (Xiao, Wilson and Hancock) – generative model from heat-kernel embedding of graphs.

Structural learning

Using description length

Description length

- Wallace+Freeman: minimum message length.

- Rissanen: minimum description length.

  Use log-posterior probability to locate model that is optimal with respect to code-length.

Similarities/differences

- MDL: selection of model is aim; model parameters are simply a means to this end. Parameters usually maximum likelihood. Prior on parameters is flat.

- MML: Recovery of model parameters is central. Parameter prior may be more complex.

Coding scheme

- Usually assumed to follow an exponential distribution.
- Alternatives are universal codes and predictive codes.
- MML has two part codes (model+parameters). In MDL the codes may be one or two-part.

Method

- Model is supergraph (i.e. Graph prototypes) formed by graph union.
- Sample data observation model: Bernoulli distribution over nodes and edges.
- Mode: complexity: Von-Neumann entropy of supergraphs.
- Fitting criterion:

  MDL-like-make ML estimates of the Bernoulli parameters

  MML-like: two-part code for data-model fit + supergraph complexity.

Model overview

- Description length criterion

$$L(G,\Gamma) = LL(G\,|\,\Gamma) + H(\Gamma)$$

code-length=negative log-likelihood + model code-length (entropy)

Data-set: set of graphs G

Model: prototype graph+correspondences with it

Updates by expectation maximisation:

Model graph adjacency matrix (M-step)

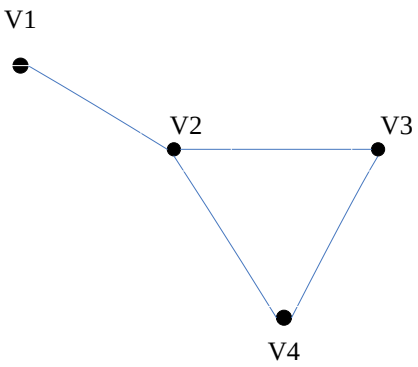
+ correspondence indicators (E-step).

Learn supergraph using MDL

- Follow Torsello and Hancock and pose the problem of learning generative model for graphs as that of learning a supergraph representation.
- Required probability distributions is an extension of model developed by Luo and Hancock.
- Use von Neumann entropy to control supergraph's complexity.
- Develop an EM algorithm in which the node correspondences and the supergraph edge probability matrix are treated as missing data.

Probabilistic Framework

Here the structure of the sample graphs and the supergraph are represented by their Adjacency matrices



$$A = \begin{array}{c} \begin{matrix} V1 & V2 & V3 & V4 \end{matrix} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} V1 \\ V2 \\ V3 \\ V4 \end{matrix}$$

Observation model

Given a sample graph $G_i = (V_i, E_i)$ and a supergraph $\Gamma = (V_\Gamma, E_\Gamma)$

$$D^i_{ab} = \begin{cases} 1 & \text{if } (a,b) \in E_i \\ 0 & \text{otherwise} \end{cases}, \quad M_{\alpha\beta} = \begin{cases} 1 & \text{if } (\alpha,\beta) \in E_\Gamma \\ 0 & \text{otherwise} \end{cases}$$

along with their assignment matrix,

$$s^i_{a\alpha} = \begin{cases} 1 \text{ if } a \to \alpha \\ 0 \text{ otherwise} \end{cases}$$

the *a posteriori* probabilities of the sample graphs given the structure of the supergraph and the node correspondences is defined as

$$P(G_i|\Gamma, S^i) = \prod_{a \in V_i} \sum_{\alpha \in V_\Gamma} K^i_a \exp[\mu \sum_{b \in V_i} \sum_{\beta \in V_\Gamma} D^i_{ab} M_{\alpha\beta} s^i_{b\beta}]$$

Data code-length

- For the sample graph-set $\mathcal{G} = \{ G_1, ...G_i, ...G_N \}$ and the supergraph Γ, the set of assignment is $\mathcal{S} = \{S^1, ...S^i, ...S^N\}$. Under the assumption that the graphs in $\mathcal{G}$ are independent samples from the distribution, the likelihood of the sample graphs can be written

$$P(\mathcal{G}|\Gamma, \mathcal{S}) = \prod_{G_i \in \mathcal{G}} P(G_i|\Gamma, S^i) = \prod_{G_i \in \mathcal{G}} \prod_{a \in V_i} \sum_{\alpha \in V_\Gamma} K_a^i \exp[\mu \sum_{b \in V_i} \sum_{\beta \in V_\Gamma} D_{ab}^i M_{\alpha\beta} s_{b\beta}^i]$$

- Code length of observed data

$$LL(\mathcal{G}|\Gamma) = -\frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} \log P(G_i|\Gamma, S^i)$$

Information theory

- Entropic measures of complexity: Shannon , Erdos-Renyi, Von-Neumann.
- Description length: fitting of models to data, entropy (model cost) tensioned against log-likelihood (goodness of fit).
- Kernels: Use entropy to computeJensen-Shannon divergence

Von-Neumann Entropy

- Derived from normalised Laplacian spectrum

$$H_{VN} = -\sum_{i=1}^{|V|} \frac{\hat{\lambda}_i}{2} \ln \frac{\hat{\lambda}_i}{2}$$

$$\hat{L} = T^{-1/2}(D - A)T^{-1/2} = \hat{\Phi}\hat{\Lambda}\hat{\Phi}^T$$

- Comes from quantum mechanics and is entropy associated with density matrix.

Approximation

- Quadratic entropy

$$H_{VN} = \sum_{i=1}^{|V|} \frac{\hat{\lambda}_i}{2}\left\{1 - \frac{\hat{\lambda}_i}{2}\right\} = \frac{1}{2}\sum_{i=1}^{|V|} \hat{\lambda}_i - \frac{1}{4}\sum_{i=1}^{|V|} \hat{\lambda}_i^2$$

- In terms of matrix traces

$$H_{VN} = \frac{1}{2} Tr[\hat{L}] - \frac{1}{4} Tr[\hat{L}^2]$$

Computing Traces

- Normalised Laplacian

$$Tr[\hat{L}] = |V|$$

- Normalised Laplacian squared

$$Tr[\hat{L}^2] = |V| + \sum_{(u,v)\in E} \frac{1}{4T_u T_v}$$

Simplified entropy

Collect terms together, von Neumann entropy reduces to

$$H_{VN} = \frac{1}{4}|V| - \sum_{(u,v)\in E} \frac{1}{4T_u T_v}$$

Uses

- Complexity-based clustering (especially protein-protein interaction networks).
- Defining information theoretic (Jensen-Shannon) kernels.
- Controlling complexity of generative models of graphs.

Overall code-length

- According to Rissanen and Grunwald's minimum description length criterion, we encode and transmit the sample graphs and the supergraph structure. This leads to a two-part message whose total length is given

$$\mathcal{L}(\mathcal{G},\Gamma) = LL(\mathcal{G}|\Gamma) + H_{VN} =$$
$$-\frac{1}{|\mathcal{G}|}\sum_{G_i\in\mathcal{G}}\sum_{a\in V_i}\log\{\sum_{\alpha\in V_\Gamma}K_a^i\exp[\mu\sum_{b\in V_i}\sum_{\beta\in V_\Gamma}D_{ab}^i M_{ab}s_{b\beta}^i]\}+\frac{|V_\Gamma|}{4}-\sum_{(\alpha,\beta)\in E_\Gamma}\frac{1}{4\,T_\alpha T_\beta}$$

- We consider both the node correspondence information between graphs S and the structure of the supergraph M as missing data and locate M by minimizing the overall code-length using EM algorithm.

EM – code-length criterion

$$\Lambda^{(n+1)}(\mathcal{G}|\Gamma,\mathcal{S}^{(n+1)}) = \frac{1}{|\mathcal{G}|}\sum_{G_i\in\mathcal{G}}\sum_{a\in V_i}\sum_{\alpha\in V_\Gamma} Q_{a\alpha}^{i,(n)}\{\ln K_a^i + \mu\sum_{b\in V_i}\sum_{\beta\in V_\Gamma} D_{ab}^i M_{\alpha\beta}^{(n)} s_{b\beta}^{i,(n+1)}\}$$

$$-\frac{|V_\Gamma|}{4} + \sum_{(\alpha,\beta)\in E_\Gamma}\frac{1}{4\,T_\alpha^{(n)}T_\beta^{(n)}} \quad . \qquad (16)$$

Expectation + Maximization

- M-step :

  Recover correspondence matrices: Take partial derivative of the weighted log-likelihood function and soft assign.

$$\frac{\partial \hat{\Lambda}^{(n+1)}}{\partial s_{b\beta}^{i,(n+1)}} = \frac{1}{|\mathcal{G}|} \sum_{a \in V_i} \sum_{\alpha \in V_\Gamma} Q_{a\alpha}^{i,(n)} D_{ab}^{i} M_{\alpha\beta}^{(n)}$$

$$s_{a\alpha}^{i,(n+1)} \leftarrow \exp[\frac{1}{\tau} \frac{\partial \hat{\Lambda}^{(n+1)}}{\partial s_{a\alpha}^{i,(n+1)}}] / \sum_{\alpha' \in V_\Gamma} \exp[\frac{1}{\tau} \frac{\partial \hat{\Lambda}^{(n+1)}}{\partial s_{a\alpha'}^{i,(n+1)}}]$$

  Modify supergraph structure :

$$M_{\alpha\beta}^{(n+1)} \leftarrow \exp[\frac{1}{\tau} \frac{\partial \hat{\Lambda}^{(n+1)}}{\partial M_{\alpha\beta}^{(n)}}] \; / \sum_{(\alpha',\beta') \in E_\Gamma} \exp[\frac{1}{\tau} \frac{\partial \hat{\Lambda}^{(n+1)}}{\partial M_{\alpha'\beta'}^{(n)}}]$$

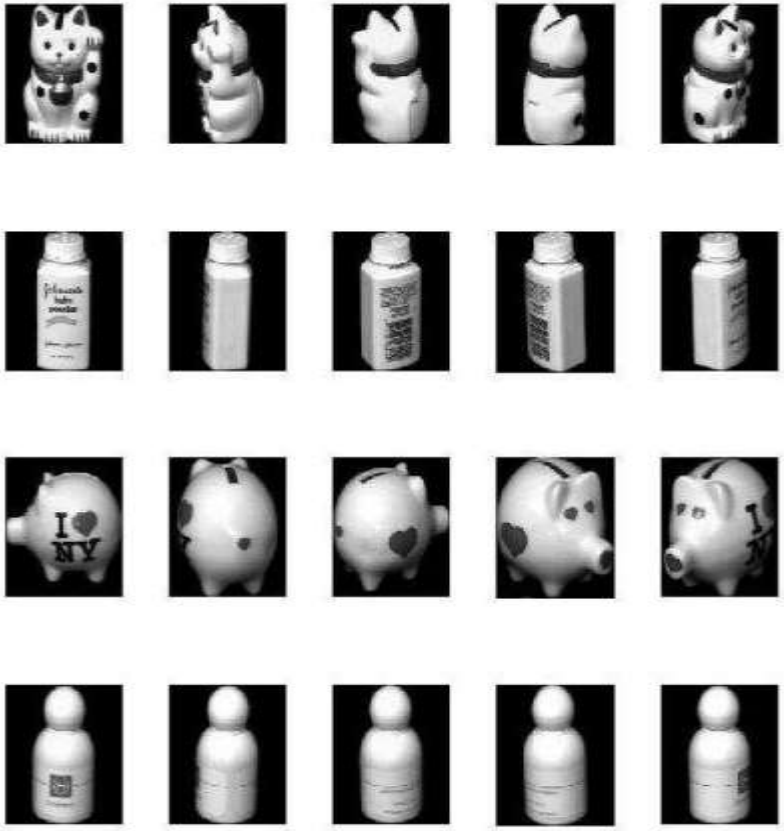
$$\frac{\partial \hat{\Lambda}^{(n+1)}}{\partial M_{\alpha\beta}^{(n)}} = \frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} \sum_{a \in V_i} \sum_{b \in V_i} Q_{a\alpha}^{i,(n)} D_{ab}^{i} s_{b\beta}^{i,(n+1)} - \frac{1}{(T_\alpha^{(n)})^2} \sum_{(\alpha,\beta') \in E_\Gamma} \frac{1}{4\, T_{\beta'}^{(n)}}$$

- E-step: Compute the *a posteriori* probability of the nodes in the sample graphs being matching to those of the supergraph.

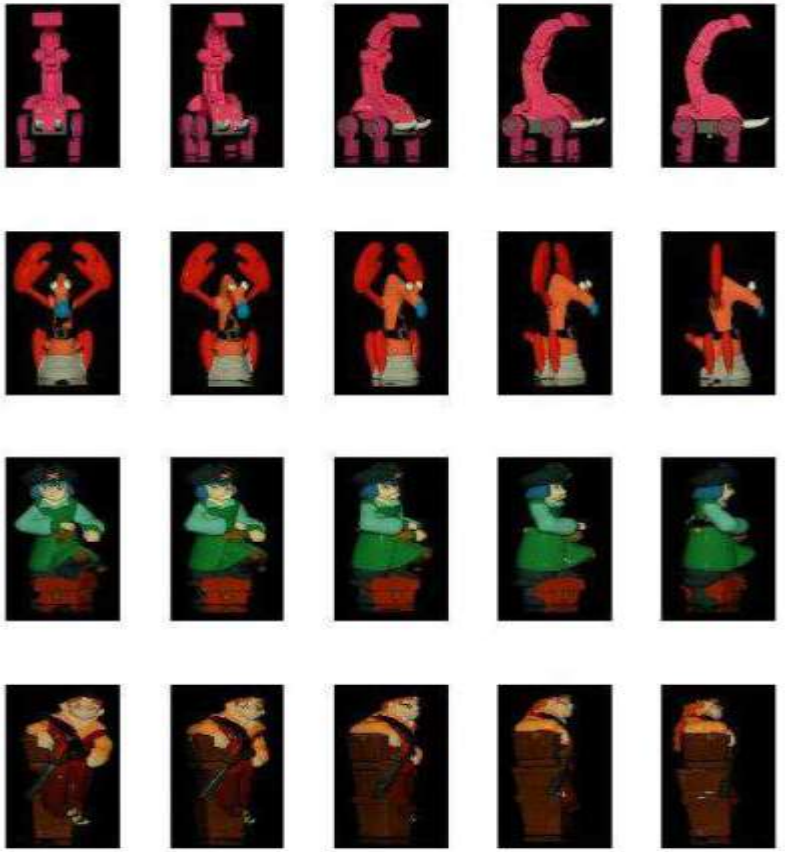
$$Q_{a\alpha}^{i,(n+1)} = \frac{\exp[\sum_{b \in V_i} \sum_{\beta \in V_\Gamma} D_{ab}^{i} M_{\alpha\beta}^{(n)} s_{b\beta}^{i,(n)}] \pi_\alpha^{i,(n)}}{\sum_{\alpha' \in V_\Gamma} \exp[\sum_{b \in V_i} \sum_{\beta \in V_\Gamma} D_{ab}^{i} M_{\alpha'\beta}^{(n)} s_{b\beta}^{i,(n)}] \pi_{\alpha'}^{i,(n)}}$$

Experiments

Delaunay graphs from images of different objects.



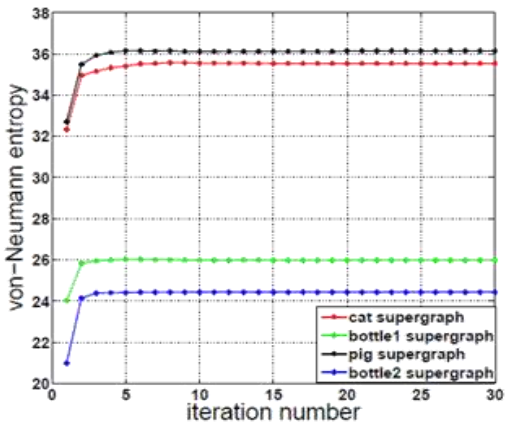
COIL dataset

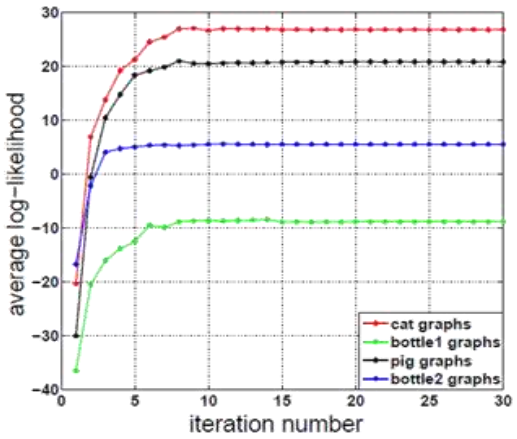


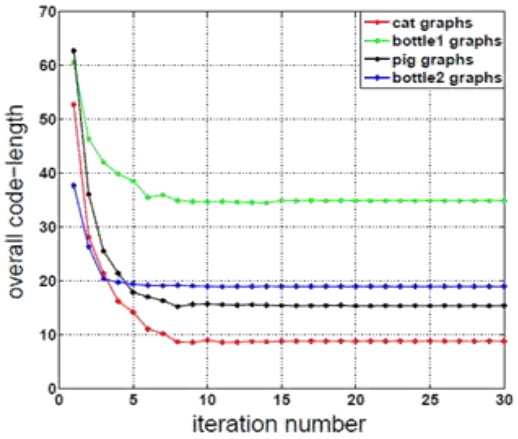
Toys dataset

Experiments---validation

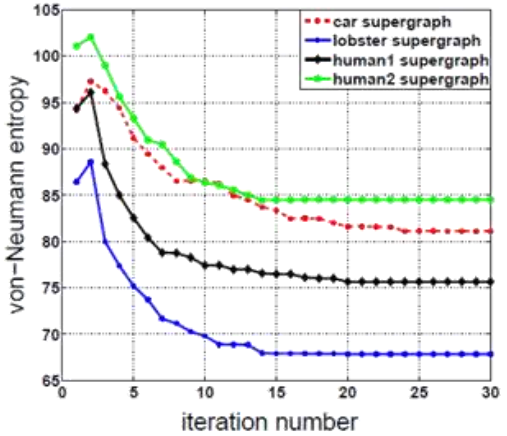
- **COIL dataset: model complexity increase, graph data log-likelihood increase, overall code length decrease during iterations.**

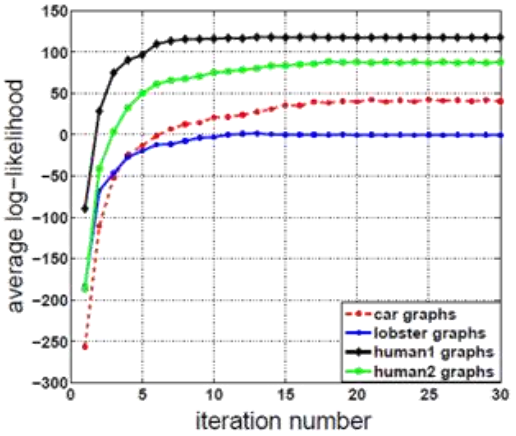


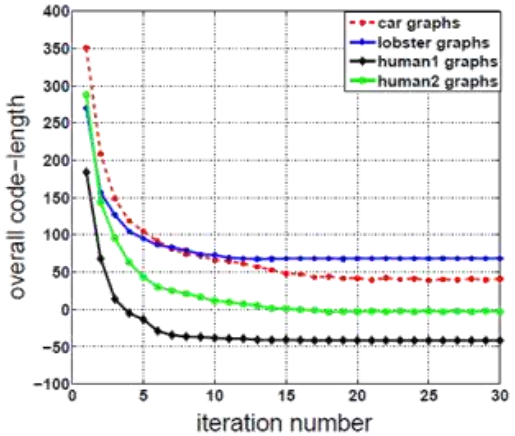




- **Toys dataset: model complexity decrease, graph data log-likelihood increase, overall code length decrease during iterations.**







Experiments---classification task

We compare the performance of our learned supergraph on classification task with two alternative constructions , the median graph and the supergraph learned without using MDL. The table below shows the average classification rates from 10-fold cross validation, which are followed by their standard errors.

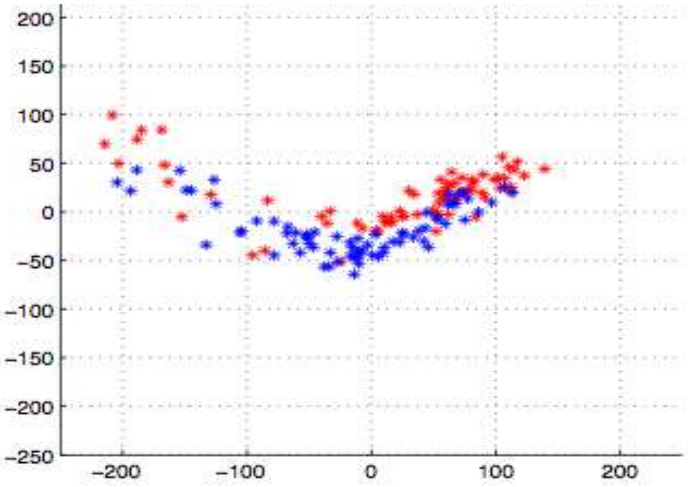
Classification Rate	cat & pig	bottle1 & bottle2	four objects (Toys)
learned supergraph(by MDL)	**0.824** $\pm$ 0.033	**0.780** $\pm$ 0.023	**0.763** $\pm$ 0.026
median graph/concatenated graph	**0.669** $\pm$ 0.052	**0.651** $\pm$ 0.023	**0.575** $\pm$ 0.020
learned supergraph	**0.807** $\pm$ 0.056	**0.699** $\pm$ 0.029	**0.725** $\pm$ 0.022

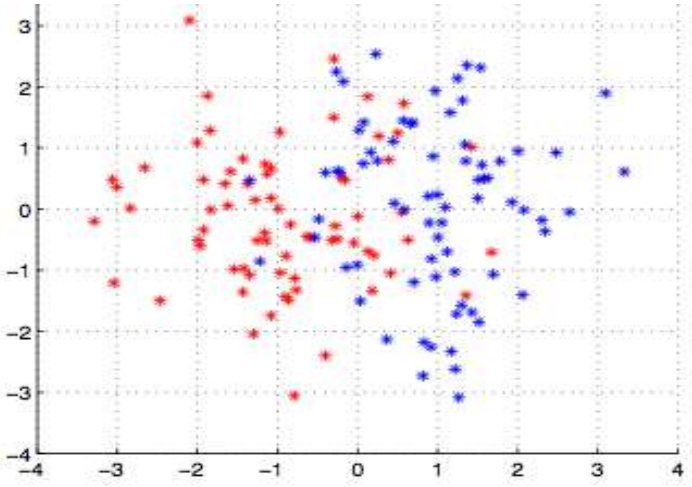
Experiments---graph embedding

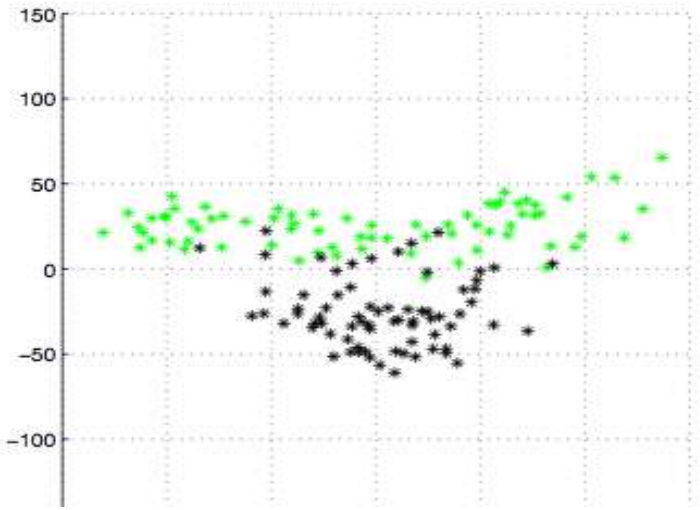
Pairwise graph distance based on the Jensen-Shannon divergence and the von Neumann entropy of graphs

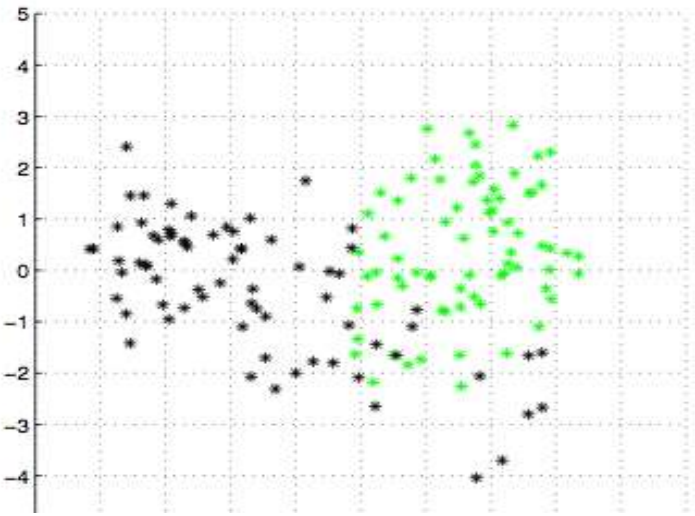
$$JSD(G_i, G_j) = H(G_i \otimes G_j) - \frac{H(G_i) + H(G_j)}{2}$$

Experiments---graph embedding





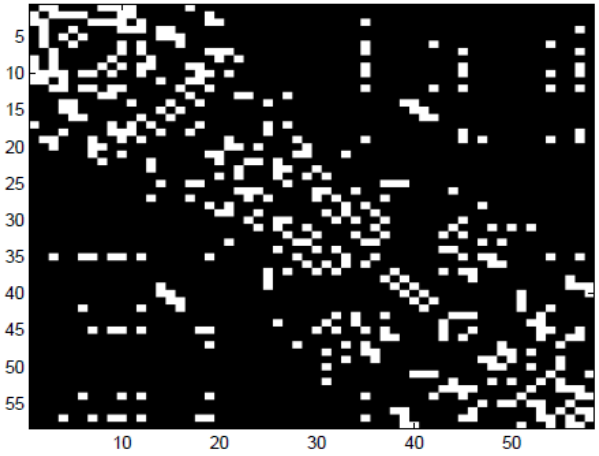




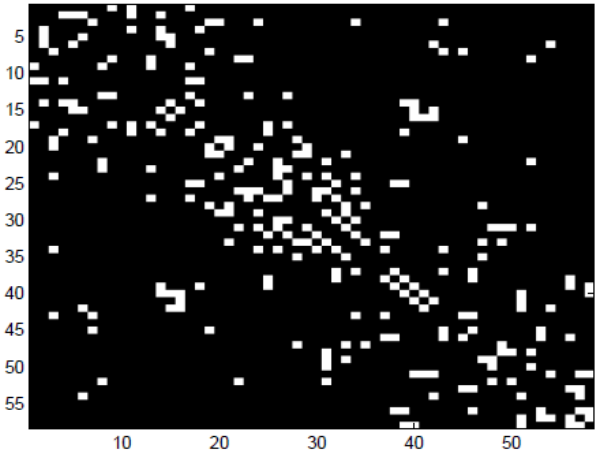
Edit distance

JSD distance

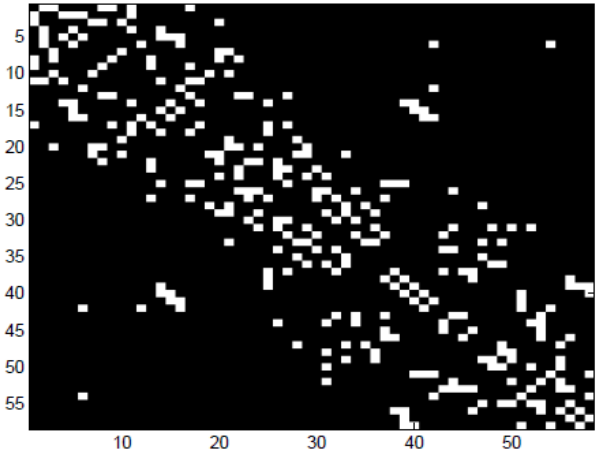
Experiments---generate new samples



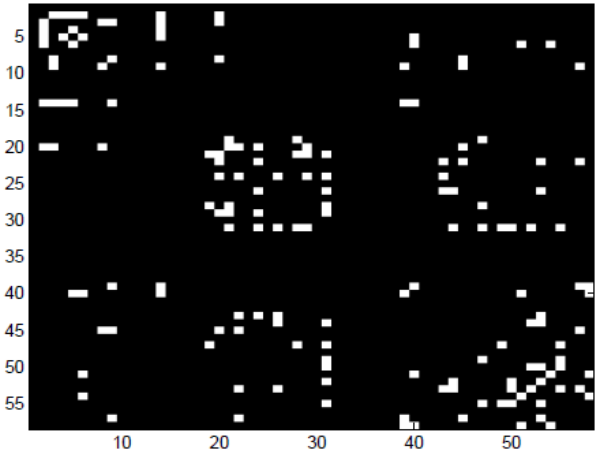
(a) Supergraph



(b) Generated sample graph with high likelihood.



(c) Median graph



(d) Generated sample graph that has low likelihood

Conclusion

- We have shown how a supergraph or generative model of graph structure can be learned under minimum description length.
- We propose a variant of EM algorithm to locate the structure of the supergraph .
- In our experiments, we demonstrate that our supergraph learning method is valid and the supergraph learned is effective for classification.

References

1. Friedman, N., Koller, D.:Being Bayesian about network structure.A Bayesian approach to structure discovery in Bayesian networks. Machine Learning,95–125(2003)
2. Christmas, W. J., Kittler, J., Petrou, M.: Modeling compatibility coefficient distribution. Image and Vsion Computing 14, 617–625 (1996)
3. Bagdanov, A.D., Worring, M.: First order Gaussian graphs for efficient structure classification. Pattern Recognition 36, 1311–1324 (2003)
4. Luo, B., Hancock, E.R.: A spectral approach to learning structural variations in graphs. Pattern Recognition 39, 1188–1198 (2006)
5. Torsello, A., Hancock, E.R.: Learning shape-classes using a mixture of tree-unions. IEEE PAMI 28, 954–967 (2006)
6. Luo, B., Hancock, E.R.: Structural graph matching using the EM alogrithm and singular value decomposition. IEEE PAMI 23, 1120–1136 (2001)
7. Rissanen, J.: Modelling by Shortest Data Description. Automatica, 465–471 (1978)
8. Passerini, F., Severini, S.: The von-neumann entropy of networks. arXiv:0812.2597 (2008)
9. Grunwald, P.: Minimum Description Length Tutorial. Advances in Minimum Description Length: Theory and Applications (2005)
10. Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. IEEE PAMI 18, 377–388 (1996)
11. Wilson, R.C., Zhu, P.: A study of graph spectra for comparing graphs and trees. Pattern Recognition 41, 2833–2841 (2008)
12. Wilson, R.C., Hancock, E.R.: Structural matching by discrete relaxation. IEEE PAMI 19, 634-648 (1997)
13. Han, L., Wilson, R.C., Hancock, E.H.: A Supergraph-based Generative Model. ICPR, pp. 1566-1569 (2010)
14. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE PAMI 24, 381–396 (2002)

Thanks ! And Questions ?