

Learning Generative Models of Graphs

Lin Han, Richard C. Wilson, Edwin R. Hancock
Department of Computer Science
The University of York
YO10 5DD, UK

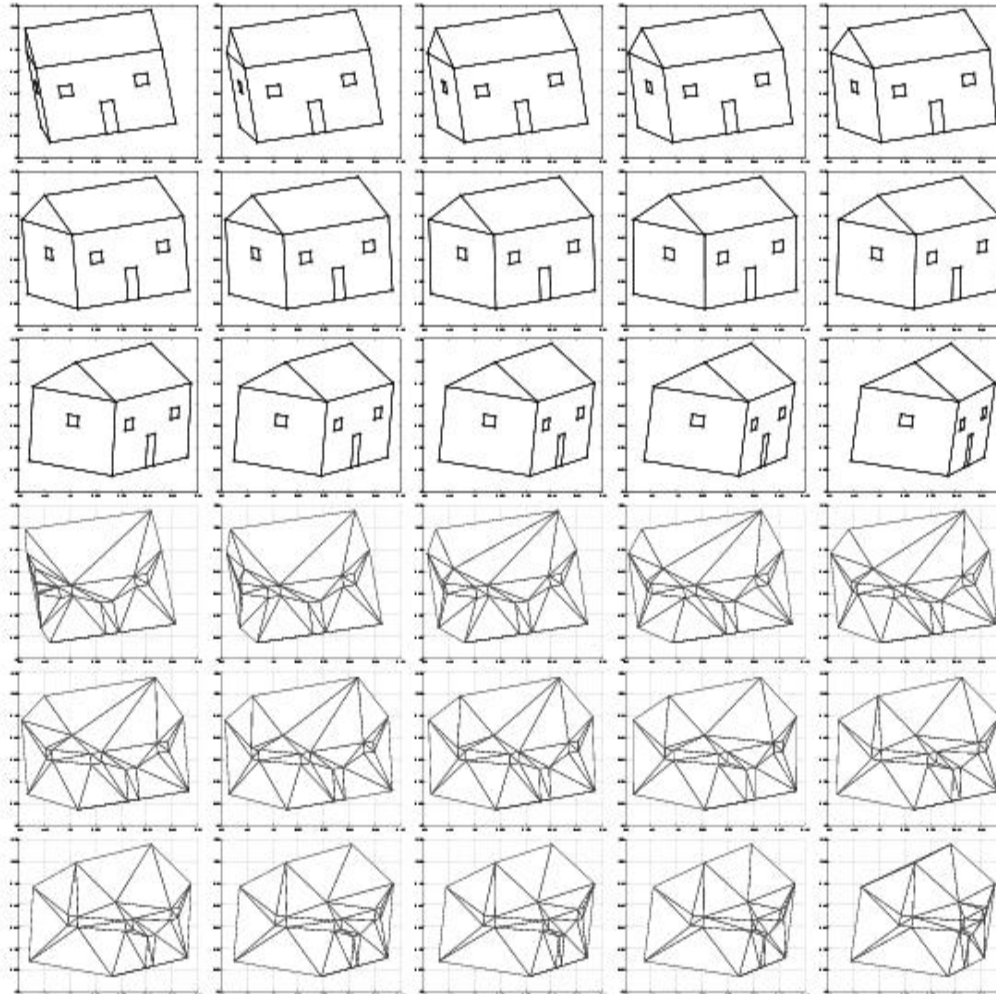
Graph representations

- Advantages: can capture object or scene structure in a manner that is invariant to changes in viewpoint. Abstract scene contents in a efficient way.
- Disadvantages: Can be fragile (sensitive to noise and segmentation error). Available pattern recognition/machine learning methodology limited.

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.

Synthetic Sequence



PCA on Adjacency matrices

Convert adjacency matrices into long-vectors and perform PCA on sequence.

Left column:

Eigenspaces

Right column:

Graph distances in eigenspace

Top row:

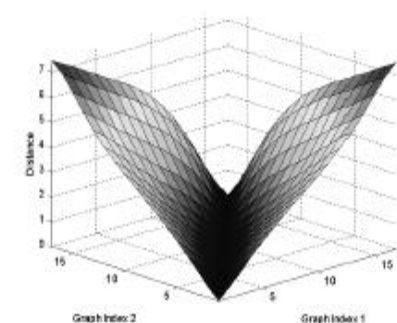
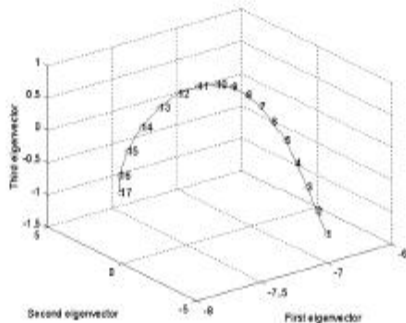
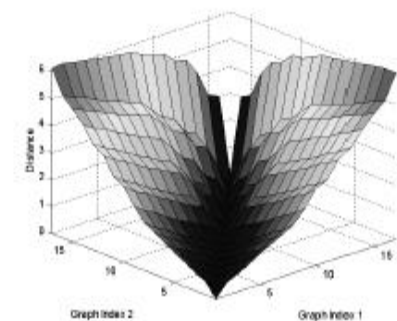
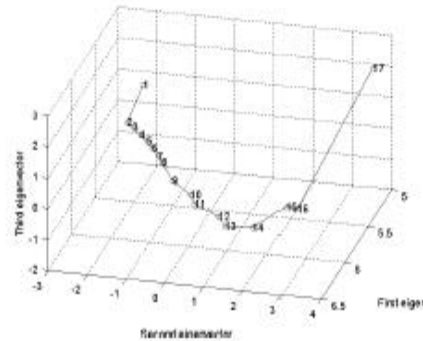
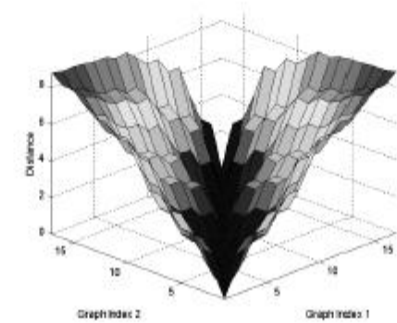
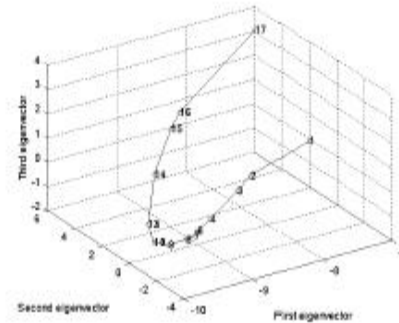
Unweighted graph

Middle row:

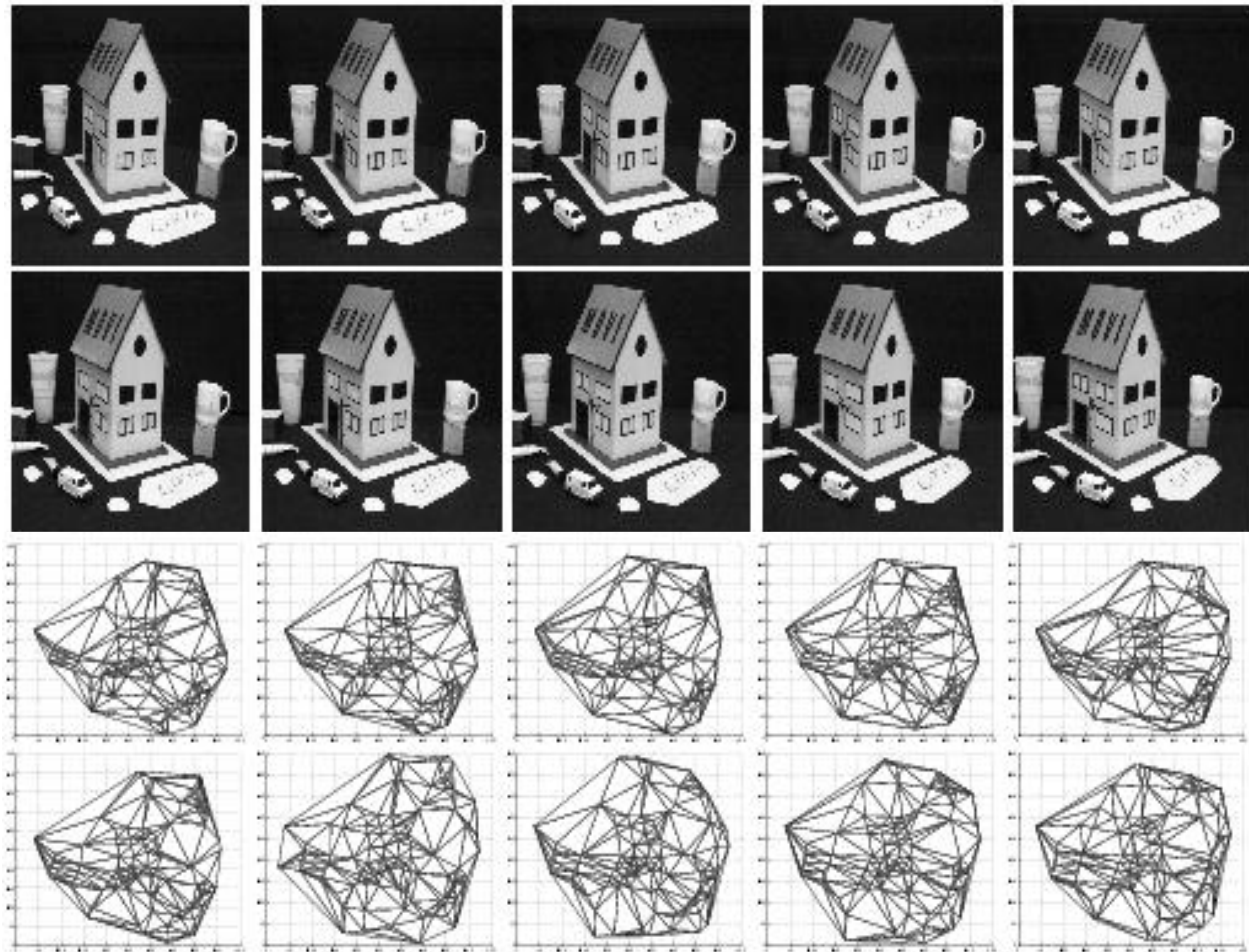
Weighted graph – proximity weights

Bottom row:

Fully connected weighted graph (point proximity matrix)



Structural Variations



Problem

In computer vision graph-structures are used to abstract image structure. However, the algorithms used to segment the image primitives are not reliable.

As a result there are both additional and missing nodes (due to segmentation error) and variations in edge-structure.

Hence image matching and recognition can not be reduced to a graph isomorphism or even a subgraph isomorphism problem.

Instead inexact graph matching methods are needed.

Problem

In computer vision graph-structures are used to abstract image structure. However, the algorithms used to segment the image primitives are not reliable.

As a result there are both additional and missing nodes (due to segmentation error) and variations in edge-structure.

Hence image matching and recognition can not be reduced to a graph isomorphism or even a subgraph isomorphism problem.

Instead inexact graph matching methods are needed.

Problem

In computer vision graph-structures are used to abstract image structure. However, the algorithms used to segment the image primitives are not reliable.

As a result there are both additional and missing nodes (due to segmentation error) and variations in edge-structure.

Hence image matching and recognition can not be reduced to a graph isomorphism or even a subgraph isomorphism problem.

Instead inexact graph matching methods are needed.

..... 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 summarised:** Since they do not reside in a vector space, mean and covariance hard to characterise.

Learning with graphs

Learn class structure: Assign graphs to classes. Need a distance measure or vector of graph characteristics. Central clustering is possible with characteristics but difficult when number of nodes and edges varies and correspondences are not known. Easier to perform pairwise clustering. (Bunke, Buhman).

Embed graphs in a low dimensional space: Correspondences are again needed, but spectral methods may offer a solution. Can apply standard statistical and geometric learning methods to graph-vectors.

Learn modes of structural variation: Understand how edge (connectivity) structure varies for graphs belonging to the same class. (Dickinson, Williams)

Construct generative model: Borrow ideas from graphical models (Langley, Friedman, Koller).

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.

Generative Models

- Structural domain: define probability distribution over prototype structure. Prototype together with parameters of distribution minimise description length.
- Spectral domain: embed nodes of graphs into vector-space using spectral decomposition. Construct point distribution model over embedded positions of nodes.

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.

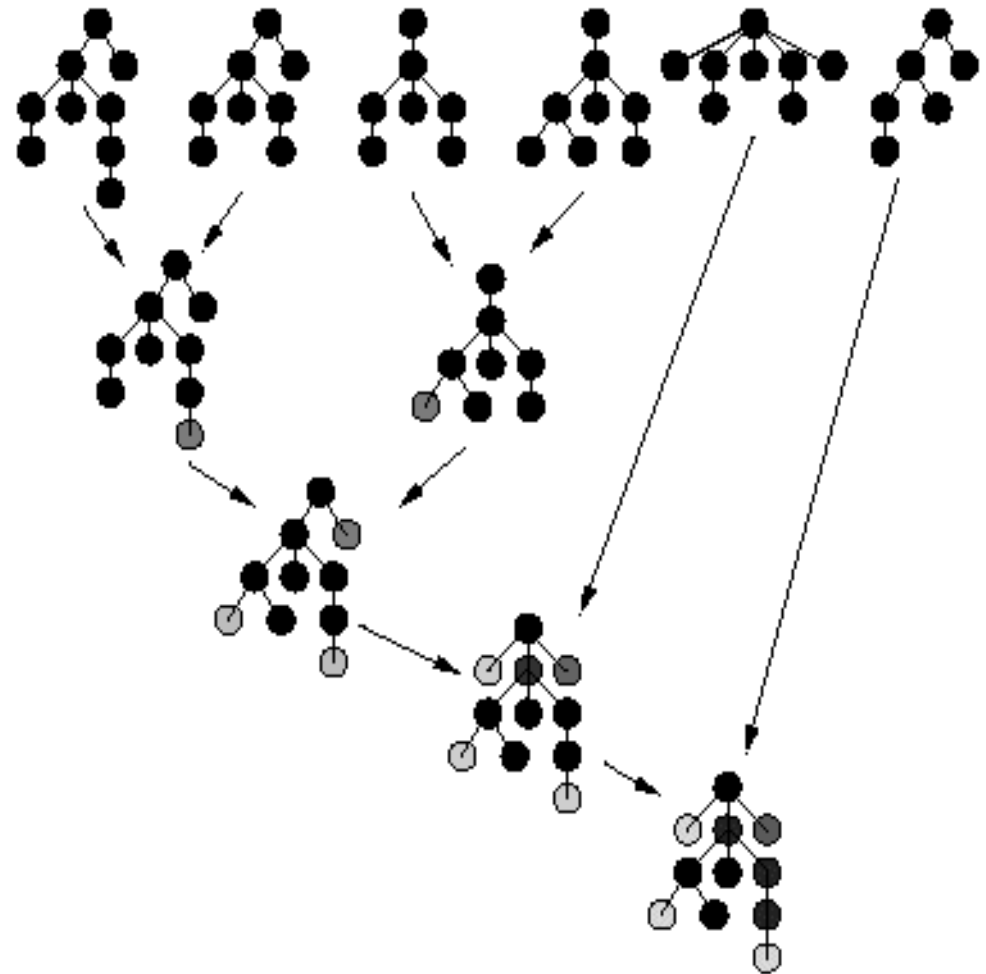
Method

- Mixture of tree unions with each modelled using a Bernoulli distribution over nodes.
- MDL-like: make ML estimates of the Bernoulli parameters.
- MML-like: three-part code for data-model match, mixture model parameters and tree complexity.

Algorithm

- Assign each sample trees to a tree union.
- Union trees formed by merging sample trees, and this requires node correspondences.
- Each sample tree can be obtained from a tree union through node removal operations.
- Control tree merging using a minimum code length criterion.
- Node correspondence frequencies gives ML estimates of node observation probabilities.

Illustration

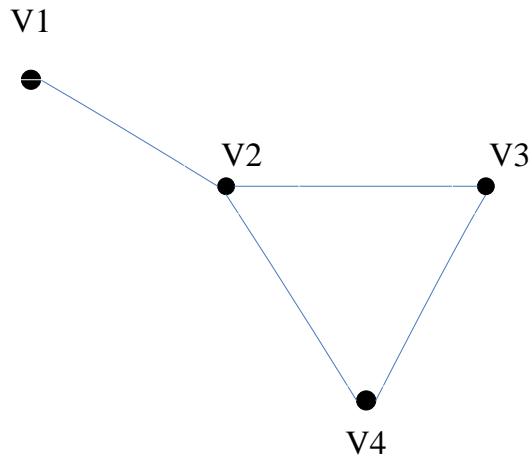


Extension to graphs

- Aim to learn a generative model that can capture variations in edge structure when correspondences are not available and must be inferred from the data.
- Follow Torsello and Hancock and pose the problem as that of learning a supergraph representation from which each sample graph can be obtained by edit operations.
- Required probability distributions extension of model developed by Luo and Hancock .
- Develop an EM algorithm in which the node correspondences and the supergraph edge probability matrix are treated as missing data.
- Using von-Neuman entropy developed method to minimize code-length.

Probabilistic Framework

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



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

Probabilistic Framework

Given a sample graph $G_D = (V_D, E_D)$ and a supergraph $G_M = (V_M, E_M)$

$$D_{ab} = \begin{cases} 1 & \text{if } (a, b) \in E_D \\ 0 & \text{otherwise} \end{cases}, M_{\alpha\beta} = \begin{cases} 1 & \text{if } (\alpha, \beta) \in E_M \\ 0 & \text{otherwise} \end{cases}$$

along with their assignment matrix S ,

$$s_{a\alpha} = \begin{cases} 1 & \text{if } f(a) = \alpha \\ 0 & \text{otherwise} \end{cases}$$

the *a posteriori* probabilities of the sample graphs given the structure of the supergraph and the node correspondences is defined as [5].

$$P(G_D|G_M, S) = \prod_{a \in V_D} \sum_{\alpha \in V_M} K_a \exp[\mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} s_{b\beta}]$$

Data Log-likelihood

- Let the graphs in the sample set be $\{G_{D^i}\} = \{G_{D^1}, \dots, G_{D^i}, \dots, G_{D^N}\}$. The assignment matrices $\{S^i\}$ represent the correspondences between the nodes of the sample graphs and those of the supergraph. Under the assumption that the graphs are independent samples from the distribution, the likelihood of the sample graphs can be written

$$P(\{G_{D^i}\} | G_M, \{S^i\}) = \prod_{i \in N} \prod_{a \in V_{D^i}} \sum_{\alpha \in V_M} K_a^i \exp[\mu \sum_{b \in V_{D^i}} \sum_{\beta \in V_M} D_{ab}^i M_{\alpha\beta} S_{b\beta}^i]$$

- We aim to locate the supergraph that maximizes the above likelihood function. To deal with the missing node correspondence matrices and the structure of the supergraph, we choose to use expectation maximization algorithm to locate the solution.

Expected Log-likelihood

- According to [5], the expected log-likelihood function for observing sample graph G_D , i.e. for it to have been generated by the supergraph G_M is

$$\Lambda(S^{(n+1)}|S^{(n)}) = \sum_{a \in V_D} \sum_{\alpha \in V_M} Q_{a\alpha}^{(n)} \{ \ln K_a + \mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} s_{b\beta}^{(n+1)} \}$$

- To develop the expected log-likelihood function for our supergraph model, since we do not know the supergraph adjacency matrix M , we work with its expectation value P and obtain

$$\Lambda(\{S^i\}^{(n+1)}|\{S^i\}^{(n)}) = \sum_{i \in N} \sum_{a \in V_{Di}} \sum_{\alpha \in V_M} Q_{a\alpha}^{i,(n)} \{ \ln K_a^i + \mu \sum_{b \in V_{Di}} \sum_{\beta \in V_M} D_{ab}^i P_{\alpha\beta}^{(n)} s_{b\beta}^{i,(n+1)} \}$$

Where $P_{\alpha\beta}^{(n)} = E[M_{\alpha\beta}] = P(M_{\alpha\beta} = 1 | \{G_{Di}\}, \{S^i\}^{(n)})$.

Learning the Supergraph

- We assume graph edges arise as independent samples under Bernoulli distribution and then the likelihood becomes

$$P(\{G_{D^i}\} | G_M, \{S^i\}) = \prod_{i \in N} \prod_{a, b \in V_{D^i}} \sum_{\alpha, \beta \in V_M} P_{\alpha\beta}^{s_{a\alpha}^i s_{b\beta}^i D_{ab}^i} (1 - P_{\alpha\beta})^{1 - s_{a\alpha}^i s_{b\beta}^i D_{ab}^i}$$

so

$$P_{\alpha\beta} = \frac{1}{N} \sum_{i \in N} \sum_{a, b \in V_{D^i}} s_{a\alpha}^i s_{b\beta}^i D_{ab}^i$$

- Because $\sum_{a \in V_D} \sum_{\alpha \in V_M} Q_{a\alpha}^{(n)} \ln K_a = \sum_{a \in V_D} \ln K_a$, the critical quantity in the weighted log-likelihood function becomes

$$\begin{aligned} \hat{\Lambda}(\{S^i\}^{(n+1)} | \{S^i\}^{(n)}) &= \sum_{i \in N} \sum_{a \in V_{D^i}} \sum_{\alpha \in V_M} \sum_{b \in V_{D^i}} \sum_{\beta \in V_M} Q_{a\alpha}^{i, (n)} D_{ab}^i P_{\alpha\beta}^{(n)} s_{b\beta}^{i, (n+1)} \\ &= \sum_{i \in N} \text{Tr}[(D^i)^T Q^{i, (n)} P^{(n)} (S^{i, (n+1)})^T] \end{aligned}$$

Expectation + Maximisation

Apply EM algorithm to recover node correspondences

- M-step: The maximization step involves recovering the elements in $\{S^i\}^{(n+1)}$ that satisfy the condition

$$S^{i,(n+1)} = \arg \max_{\hat{S}} \text{Tr}[(D^i)^T Q^{i,(n)} P^{(n)} \hat{S}^T]$$

Perform singular value decomposition on $(D^i)^T Q^{i,(n)} P^{(n)} = V \Delta U^T$. Get matrix E by making the diagonal elements in Δ unity and compute matrix $R = V E U^T$.

We recover the correspondence matrices by

$$S_{a\alpha}^{i,(n+1)} = \begin{cases} 1 & \text{if } R_{a\alpha} = \arg \max_{b\beta} R_{b\beta} \\ 0 & \text{otherwise} \end{cases}$$

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

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

Experiments

Dataset.

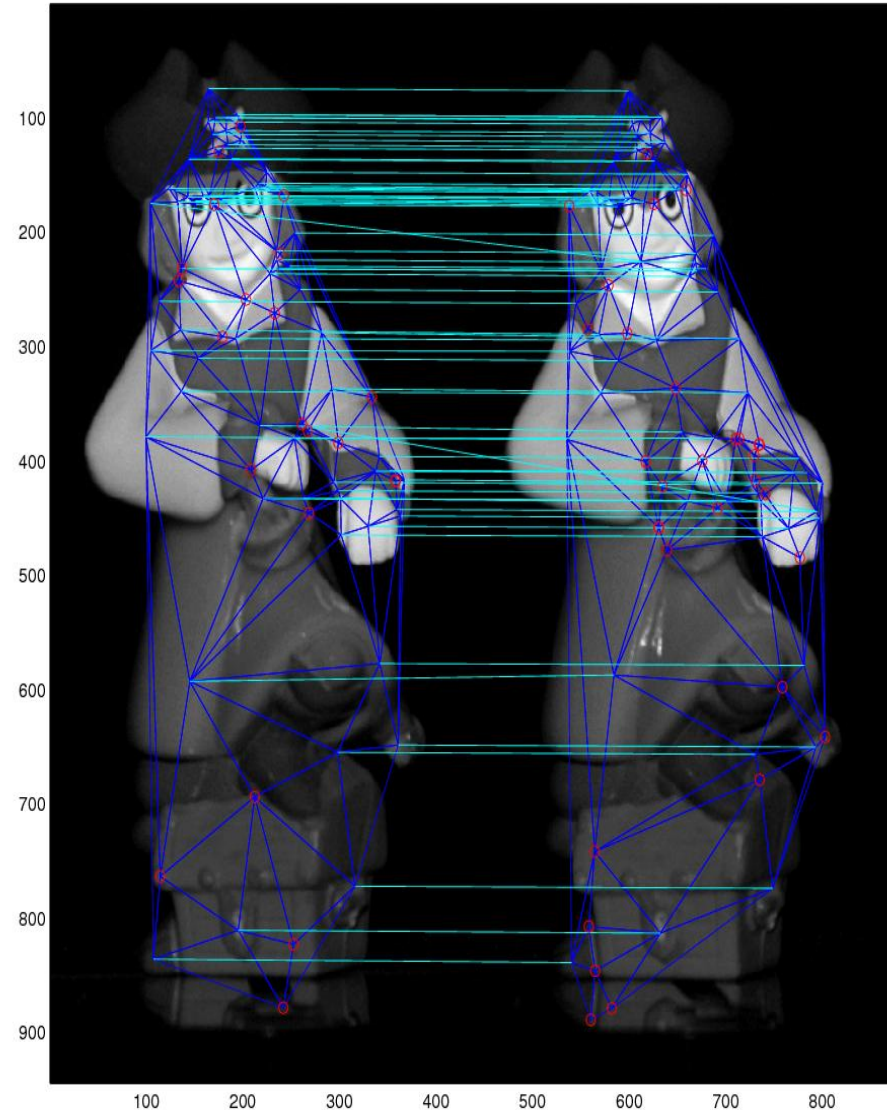
The dataset used consists of 4 objects, with 20 different images of each object. Sample graphs are constructed by first extracting feature keypoints in these object images using SIFT [7] detector and then connecting the detected points using Delaunay triangulation.



Experiments

Initialization

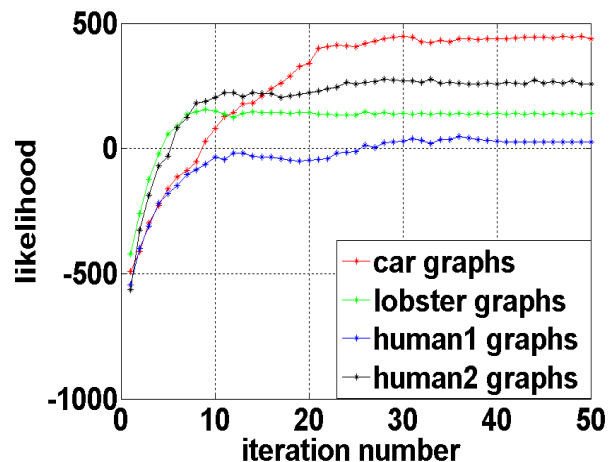
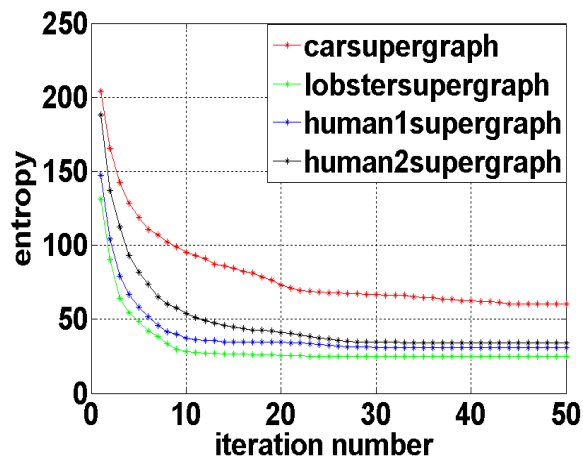
- Match pairs of the graphs from a same object using the discrete relaxation algorithm [10] and merge the common structures for pairs of graphs.
- The common structures over for the sample graphs from a same object are concatenated to form a supergraph, which is used to initialize the supergraph in our algorithm.



Experiment

- **Validation of our algorithm**

Measure the complexity of the supergraph by von-Neumann entropy $Entropy = - \sum_i \frac{\lambda_i}{2} \log \frac{\lambda_i}{2}$



Experiment result shows that the complexity of the supergraph, which is measure by the von-Neumann entropy, decreases as the iteration number increases. At the same time, the likelihood the sample graphs increases and gradually converges as the iteration number increases.

Experiment

- **Classification task**

We compare the performance of our learned supergraph on classification task with two alternative constructions of the supergraph, the initial supergraph and median graph.

The table below shows the average classification rates from 10-fold cross validation, which are followed by their standard errors.

<i>classification rate</i>	
initial supergraph	0.663 \pm 0.038
median graph	0.575 \pm 0.020
learned supergraph	0.838 \pm 0.053

Among the three constructions, our learned supergraph achieves an average classification rate of 83.8%, which is much higher than the initial supergraph's classification rate (66.3%) and the median graph(57.5%)



Description length

- Model complexity measured by Von Neuman entropy of adjacency matrix. Given in terms of Laplacian $L=D-M$ eigenvalues

$$H = \sum_k \lambda_k \ln \lambda_k$$

- Under quadratic approximation

$$H = \sum_{\alpha, \beta} M_{\alpha, \beta}^2$$



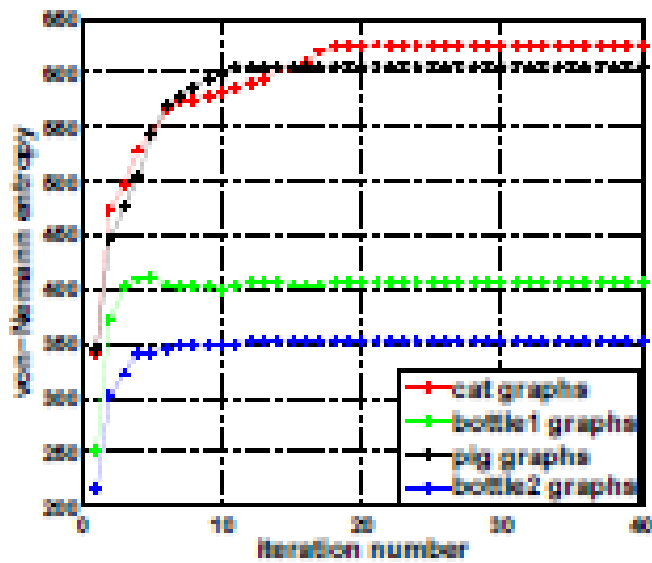
Overall code-length

Sum of expected log-likelihood and von-Neuman complexity

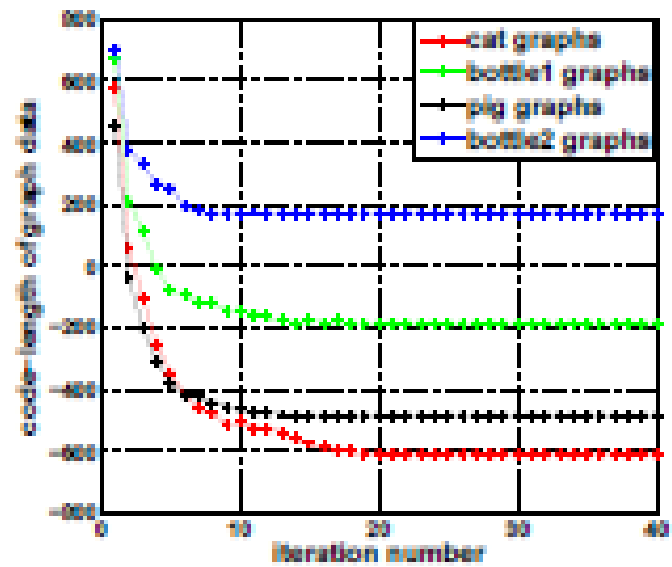
$$\mathcal{L}(\mathcal{G}, \Gamma) = -\frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} \sum_{a \in V_i} \log \left\{ \sum_{\alpha \in V_T} K_a^i \exp \left[\mu \sum_{b \in V_i} \sum_{\beta \in V_T} D_{ab}^i M_{ab} s_{b\beta}^i \right] \right\} + \sum_{\alpha \in V_T} \left(\sum_{\beta \in V_T} M_{\alpha\beta} \right)^2$$

EM reduces to optimising following with respect to M and S

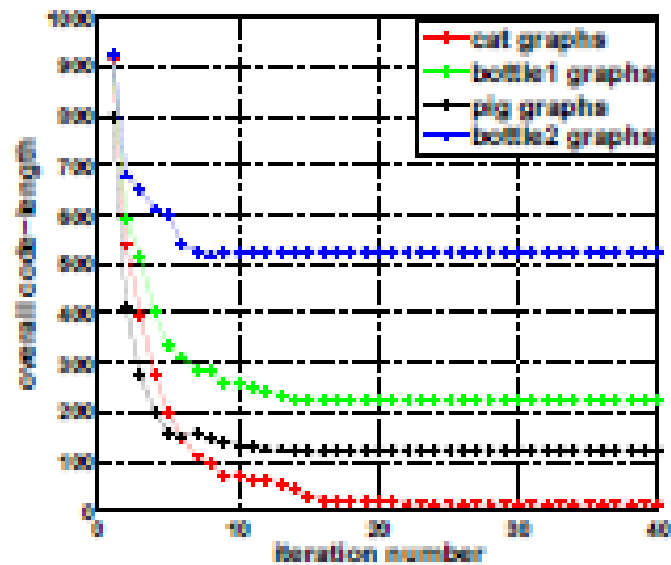
$$\hat{\Lambda}^{(n+1)} = \frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} \sum_{a \in V_i} \sum_{\alpha \in V_T} \sum_{b \in V_i} \sum_{\beta \in V_T} Q_{a\alpha}^{i,(n)} D_{ab}^i M_{\alpha\beta}^{(n)} s_{b\beta}^{i,(n+1)} - \sum_{\alpha \in V_T} \left(\sum_{\beta \in V_T} M_{\alpha\beta}^{(n)} \right)^2$$



(a)

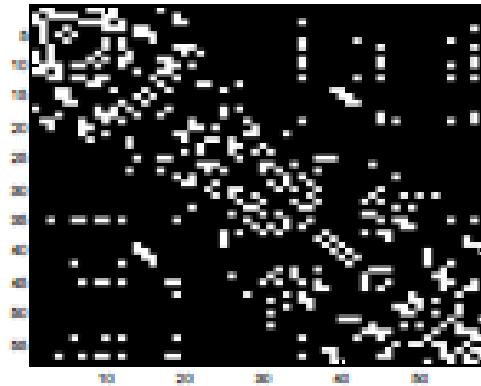


(b)

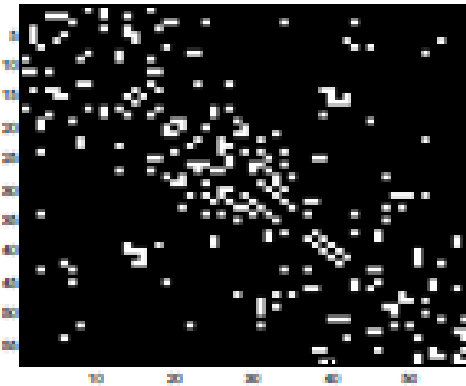


(c)

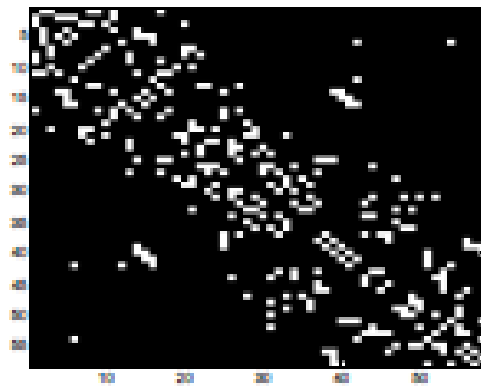
Learned models



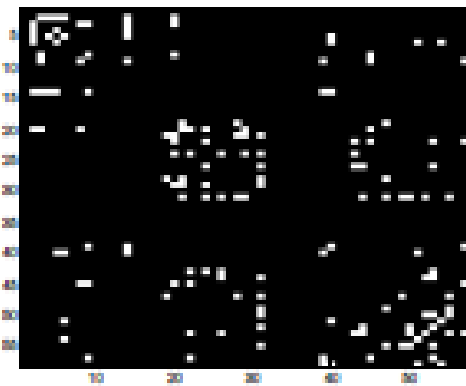
(a) Supergraph



(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 using a novel variant of the EM algorithm.
- In our experiments, we demonstrate that our supergraph learning method is valid and the supergraph learned is effective for classification.
- Currently , we carry on the research of how to generate new samples using Gibbs sampling and learn a supergraph under minimum description length criterion.

References

- [1] A.D.Bagdanov and M.Worring. First order gaussian graphs for efficient structure classification. *Pattern Recognition*, 36:1311-1324, 2003.
- [2] K.A.Shokoufandeh, S.Dickinson and S. Zucker. Indexing using a spectral encoding of topological structure. *CVPR*, pages 491–497, 1999.
- [3] A.Torsello. An importance sampling approach to learning structural representations of shape. *CVPR*, pages 1–7, 2008.
- [4] A.Torsello and E.R.Hancock. Learning shape-classes using a mixture of tree-unions. *PAMI*, 28(6):954–967, 2006.
- [5] B.Luo and E.R.Hancock. Structural graph matching using the em algorithm and singular value decomposition. *PAMI*, 23(10):1120–1136, 2001.
- [6] H.G. Barrow and R.M. Burstall, Subgraph isomorphism, matching relational structures and maximal cliques. *Inform. Process. Lett.* 4:83–84, 1996.
- [7] D.G.Lowe. Distinctive image features from scaleinvariant keypoints. *IJCV*, 99(2):91–110, 2004.
- [8] D.White and R.C.Wilson. Parts based generative models for graphs. *ICPR*, pages 1–4, 2008.
- [9] M. Fischler and R. Elschlager, The representation and matching of pictorial structures. *IEEE Trans. Comput.* 22: 67–92, 1973.
- [10] R.C.Wilson and E.R.Hancock. Structural matching by discrete relaxation. *PAMI*, 19(6):634–648, 1997.
- [11] S.Gold and A.Rangarajan. A graduated assignment algorithm for graph matching. *PAMI*, 18(4):377–388, 1996.
- [12] W.J.Christmas and M.Petrou. Structural matching in computer vision using probabilistic relaxation. *PAMI*, 17(8):749–764, 1995.

Thanks ! And Questions ?