An Importance Sampling Approach to Learning Structural Representations of Shape

Andrea Torsello
Dipartimento di informatica
Università “Ca’Foscari” di Venezia
torsello@dsi.unive.it

Abstract

This paper addresses the problem of learning archetypal structural models from examples. This is done by providing a generative model for graphs where the distribution of observed nodes and edges is governed by a set of independent Bernoulli trials with parameters to be estimated, however, the correspondences between sample node and model nodes is not known and must be estimated from local structure. The parameters are estimated maximizing the likelihood of the observed graphs, marginalizing it over all possible node correspondences. This is done adopting an importance sampling approach to limit the exponential explosion of the set of correspondences. The approach is used to summarize the variation in two different structural abstraction of shape: Delaunay graph over a set of image features and shock graphs. The experiments show that the approach can be used to recognize structures belonging to a same class.

1. Introduction

Graph-based representations [1] have been used with considerable success in computer vision in the abstraction and recognition of object shape and scene structure. Concrete examples include the use of shock graphs to represent shape-skeletons [14], the use of trees to represent articulated objects [15, 12, 20] and the use of aspect graphs for 3D object representation [5]. The attractive feature of structural representations is that they concisely capture the relational arrangement of object primitives, in a manner which can be invariant to changes in object viewpoint. However, despite their many advantages, the methodology available for learning structural representations from sets of training examples is relatively limited. As a result, the process of constructing archetypal representations which capture the modes of structural variation for sets of graphs has proved to be elusive. For this reason geometric representations of shape such have proved to be more amenable when variable sets of shapes must be analyzed.

There has been considerable interest in learning structural representations from samples of training data in the context of Bayesian networks [11, 7], or general relational models [8]. The idea is to associate random variables with the nodes of the structure and to use a structural learning process infer the stochastic dependency between these variables. Although these approaches provide a powerful way to infer the relations between the observable quantities of the model under examination, they rely on the availability of correspondence information for the nodes of the different structures used in learning. The problem that we wish to address here is complementary to that of learning a graphical model. In the case of a graphical model, the training data is accompanied with complete correspondence information, but the structural information is absent and must be inferred from the data. When learning structural archetypes, on the other hand, the data has structural organization, but correspondence information is lacking and must be estimated using graph matching techniques. Additionally, in the latter problem, the structural information may also be incomplete and noisy.

Recently, however, there has been some effort aimed at learning structural archetypes and clustering data abstracted in terms of graphs. Jain and Wysotzki adopt a geometric approach which aims to embed graphs in a high-dimensional space by means of the Schur-Hadamard inner product [13], while Hagenbuchner et al. [9] use Recursive Neural Networks to perform unsupervised learning of graph structures. While these approaches preserve the structural information present, they do not provide a means of characterizing the modes of structural variation encountered and this renders them of limited use for the analysis of shape. Bonev et al. [3], and Bunke et al. [4] summarize the data by creating super-graph representation from the available samples, while White and Wilson [19] use a probabilistic model over the spectral decomposition of the graphs to produce a generative model of their structure. While these techniques provide a structural model of the samples, the way in which the

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

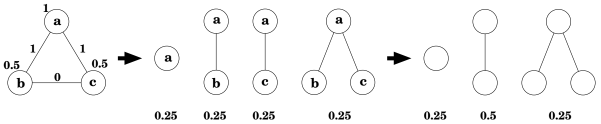


Figure 1. A structural model and the generated graphs. When the correspondence information is lost, the second and third graph become indistinguishable.

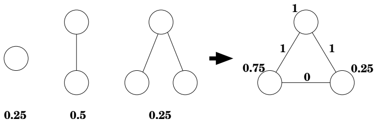


Figure 2. Model estimation bias. If a single node correspondence is taken into account the estimated model will exhibit a bias torwards one of multiple possible correspondences.

supergraph is learned or estimated is largely heuristic in nature and are not rooted in a statistical learning framework. In [18] we proposed an approach to learn trees by defining a superstructure called tree-union that captures the relations and observation probabilities of all nodes of all the trees in the training set. The structure is obtained by merging the corresponding nodes of the structures and is critically dependent on both the extracted correspondence and the order in which trees are merged. Todorovic and Ahuja [17] applied the approach to object recognition based on a hierarchical segmentation of image patches and lifted the order dependence by repeating the merger procedure several times and picking the best model according to an entropic measure. The problem with these approaches is that they are all reliant of the extraction of node correspondences which, as we will show later, may induce a bias in the estimation.

The aim in this paper is to develop a framework for the unsupervised learning of generative models of graph-structures from sets of examples. The model can then be used to perform supervised or unsupervised classification of structural abstractions of shape. Here we present an approach for unattributed graphs, but attributes and weights can be added to the model with an approach similar to [18].

2. Generative Graph model

Consider the set of undirected graphs $S = (g_1, \ldots, g_l)$, our goal is to learn a generative graph model $\mathcal{G}$ that can be used to describe the distribution of structural data and characterize the structural variations present the set. To develop this probabilistic model, we make an important simplifying assumption: We assume that the observation of each node and each edge is independent of the others. Hence, the proposed structural model is a complete graph $\mathcal{G} = (V, E, \Theta)$, where $V = \{1, \ldots, n\}$ is the set of nodes, $E \subseteq V \times V$ is the set of edges and $\Theta = (\theta_{ij})$ is a set of observation probabilities. In an observation, or sample, from this model, node $i \in V$ is present with probability θ_{ii}, i.e., the existence of each node in a sample graph is modelled as a Bernoulli trial of parameter θ_i. Futher, edge (i, j) is present with probability θ_{ij}, conditioned to the fact that both nodes i and j are present. Here we will focus on this unattributed model, but, in order to deal with weighted graphs, the Bernoulli trials can be substituted with more complex node and edge observation probabilities.

After the graph has been sampled from the generative model, we lose track of the correspondences between the sample's nodes and the nodes of the model that generated them. We can model this by saying that an unknown random permutation is applied to the nodes of the sample. For this reason, the observation probability of a sample graph depends on the unknown correspondences between sample and model nodes. Figure 1 shows a graph model and the graphs that can be generated from it with the corresponding probabilities. Here the numbers next to the nodes and edges of the model represent the values of θ_{ij}. Note that, when the correspondence information (letters in the Figure) is dropped, we cannot distinguish between the second and third graph anymore, yielding the final distribution.

Let assume that we have a model $\mathcal{G}$ with n nodes and that we want to compute the probability that graph g with m nodes was sampled from it, then clearly $m \leq n$ since $\mathcal{G}$ can only generate graphs of up to n nodes. Let A_g be the adjacency matrix of g, I_m the identity matrix of dimension m and $\mathrm{ext}_n(A)$ an operator that extends matrix A to a $n \times n$ matrix by adding columns and rows of zeroes at the end of A. We will represent graph g with the $n \times n$ matrix $G = \mathrm{ext}_n(A_g + I_m)$. This way, a graph node i is present if $G_{ii} = 1$ and an edge (i, j) is present if $G_{ij} = 1$. Further, correspondences from the nodes of extended graph representation to the nodes of the model are in correspondence with the group Σ_n of permutations over n elements.

With this notation, the probability that a graph g was sampled from a model $\mathcal{G}$ given the correspondences $\sigma \in \Sigma_n$ is $P(g|\mathcal{G}, \sigma) = \prod_{i=1}^{n} \prod_{j=i}^{n} \Theta_{ij}^{\sigma(i)\sigma(j)}$, where Θ_{ij}^{hk} is the probability that model edge (i, j) generated graph edge (h, k), where pairs with the same index represent a node instead of and edge. This probability is defined as follows:

$$\Theta_{ij}^{hk} = \begin{cases} 0 & \text{if } i = j \wedge h \neq k \text{ or } i \neq j \wedge h = k\,, \\ \theta_{ii} & \text{if } i = j \wedge h = k \wedge G_{hh} = 1\,, \\ 1 - \theta_{ii} & \text{if } i = j \wedge h = k \wedge G_{hh} = 0\,, \\ \theta_{ij} & \text{if } i \neq j \wedge h \neq k \wedge G_{hk} = 1\,, \\ 1 - \theta_{ij} & \text{if } i \neq j \wedge h \neq k \wedge G_{hh} = 1 \wedge \\ & G_{kk} = 1 \wedge G_{hk} = 0\,, \\ 1 & \text{otherwise.} \end{cases}$$

Almost invariably, the graph learning approaches in the literature have used some graph matching technique to estimate the correspondences and use them in learning the model parameters. This is equivalent to defining the sampling probability for node g as $P(g|\mathcal{G}) = \max_{\sigma\in\Sigma_n} P(g|\mathcal{G},\sigma)$. However, assuming the maximum likelihood estimation, or simply a single estimation, for the correspondences yields a bias in the estimation as shown in Figure 2. Here, the graph distribution obtained from the model in Figure 1 is used to infer a model, however, since each node of the second sample graphs is always mapped to the same model node, the resulting inferred model is different from the original one and it does not generate the same sample distribution.

To solve this bias we propose to marginalize the sampling probability over all possible correspondences, hence obtaining the probability

$$P(g|\mathcal{G}) = \sum_{\sigma\in\Sigma_n} P(g|\mathcal{G},\sigma)P(\sigma) = \frac{1}{n!}\sum_{\sigma\in\Sigma_n} P(g|\mathcal{G},\sigma)\,. \tag{1}$$

Note that this probability takes into account the order in which the sample nodes are stored. The probability if we do not consider the order, or, equivalently, the probability of the quotient group modulo graph isomorphism, is

$$P(\hat{g}|\mathcal{G}) = \frac{1}{(n-m)!|\Sigma_g|}\sum_{\sigma\in\Sigma_n} P(g|\mathcal{G},\sigma)\,, \tag{2}$$

where $(n-m)!$ is the number mapping for the nodes not in g and Σ_g is the set of symmetries of g, i.e., the set of automorphisms of g onto itself. As it will be shown, the actual value of $|\Sigma_g|$ is irrelevant when learning the model and is only important when trying to estimate $P(\hat{g}|\mathcal{G})$. However, in the rest of the paper we will assume that $|\Sigma_g| = 1$ due to an important result of graph theory by Erdös and Rényi [6] that states that almost all graphs have no automorphism other than the identity.

Clearly, averaging over all possible correspondences is not possible due to the super-exponential growth of the size of Σ_n; hence, we have to resort to an estimation approach. We propose to use importance sampling to compute a fast-converging estimate of $P(\hat{g}|\mathcal{G})$.

3. Importance Sampling

Importance Sampling [10] is a Monte Carlo sampling technique designed to reduce the variance of the estimators for averages of the type: $E[h(x)] = \frac{1}{\|A\|}\int_A h(x)\,dx\,$, where $h(x)$ is any real function in the domain A. This is done by sampling A according to a distribution f not necessarily uniform in A. Let $\mathbf{x} = (x_1,\dots,x_k)^T$ be k samples extracted from the distribution f, we estimate $E[h(x)]$ as:

$$E[h(x)] \approx \frac{1}{k}\sum_{i=1}^{k} h(x_i)\frac{\frac{1}{\|A\|}}{f(x_i)}$$

where $\frac{\frac{1}{\|A\|}}{f(x_i)}$ is called the *importance factor* of x_i. It is easy to show that the estimation is unbiased, in fact

$$E_F[h(x)\frac{\frac{1}{\|A\|}}{f(x)}] = \int_A h(x)\frac{\frac{1}{\|A\|}}{f(x)}\,dF(x) = \frac{1}{\|A\|}\int_A h(x)\frac{1}{f(x)}\,f(x)dx = E[h(x)]\,.$$

The advantage of the approach comes from a judicious choice of f which should be chosen to be close to $\frac{h(x)}{\int_A h(x)\,dx}$. In the limit, if $f(x) = \frac{h(x)}{\int_A h(x)\,dx}$, the variance of the estimator is zero and, hence, a single sample is sufficient to estimate $E[h(x)]$.

3.1. Correspondence Sampler

In order to estimate $P(g|\mathcal{G})$, and to learn the graph model, we need to sample a permutation $\sigma\in\Sigma_N$ with probability close to $\frac{P(g|\mathcal{G},\sigma)}{\sum_{\sigma\in\Sigma_n} P(g|\mathcal{G},\sigma)} = P(\sigma|g,\mathcal{G})$, that is, close to the posterior of the correspondences. Assume that we know the node-correspondence matrix $M = (m_{ih})$, which gives us the probability that model node i corresponds to graph node h. Note that M is a doubly-stochastic matrix, i.e., its rows and column add up to one. We can sample the correspondence for model node 1 picking a node h_1 with probability m_{1h_1}. Then, we need to condition the node-correspondence matrix to the current match by taking into account the structural information between the sampled node and all the others. We do this by multiplying m_{jk} by $\Theta^{h_1k}_{1j}$, i.e., the probability that the edges/non-edges between k and h_1 map to the model edge $(1,j)$. The multiplied matrix $\bar{M} = (\bar{m}_{jk} = m_{jk}\Theta^{h_1k}_{1j})$ is then projected to a double-stochastic matrix using the the Sinkhorn projection [16]. We can then sample a correspondence for model node 2 according to the distribution of the second row of $M^{h_1}_1$ and compute the conditional matching probability $M^{h_1,h_2}_{1,2}$ in much the same way we computed $M^{h_1}_1$. In general, after selecting the correspondence h_i for model node i, the conditional node matching probability matrix $M^{h_1,\dots,h_i}_{1,\dots,i}$ can be computed as follows:

$$M^{h_1,\dots,h_i}_{1,\dots,i} = \pi\Big(M^{h_1,\dots,h_{i-1}}_{1,\dots,i-1}\odot\Theta^{h_i-}_{i-}\Big)\,,$$

where $\odot$ denotes the Hadamard or element-wise product, $\Theta^{h_i-}_{i-}$ is the matrix obtained from Θ^{hk}_{ij} fixing i and h_i, and $\pi(M)$ denotes the Sinkhorn projection of matrix M. Iterating the procedure until all the node correspondences $h_1,\dots,h_n$ have been selected, we have our sampled correspondence σ such that $\sigma(i) = h_i$.

The sampling procedure can be better described with the following example: Let g be a 4 node ring and $\mathcal{G}$ a deterministic model identical to g, i.e., the g and $\mathcal{G}$ be characterized by the adjacency matrix $A = \left(\begin{smallmatrix}0&1&0&1\\1&0&1&0\\0&1&0&1\\1&0&1&0\end{smallmatrix}\right)$

and the observation probabilities $\Theta = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ respectively. In this case there are only 8 permutation with non-zero probability: 4 rotations plus the 4 rotations followed an order inversion. However, each model node i is equally likely to be matched to any graph node h, hence, giving a matching probability matrix $M = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$ which gives us very little information about the correct correspondences. To simulate the behavior of the sampler, assume we have select the mapping $1 \rightarrow 1$, i.e., a correspondence that maps the first model node to the first node of the observed graph; this map has probability 0.25. By multiplying by Θ_{1-}^{1-} and projecting the result, we obtain the conditional matching probability matrix $M_1^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0.5 & 0 & 0.5 \end{pmatrix}$, and from that, selecting the mapping $2 \rightarrow 4$ with probability 0.5, we obtain $M_{1,2}^{1,4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$. The set of correspondences in σ is now completely determined, sampled with probability 0.125, which is exactly the same as the posterior of σ.

In [2] the authors proppposed a similar, importance sampling based, approach to estimating the permanent of a matrix. They did that by sampling over the set of permutation using a similar sampler. However, in their proposal the conditional matching probability was computed by simply projecting the $(n-1) \times (n-1)$ minor of M obtained by eliminating the chosen row and column, without multiplying it by $\Theta_{1j}^{h_1 k}$. Unfortunately, this much simpler sampler will not work in our case as it does not sample close to the posterior in several important cases. In particular, in the previous ring example it would sample uniformly from the set of permutations.

3.2. Node Matching Probability

Note that the performance of the sampler is critically dependent on the initial node matching probability matrix M, which, in theory, should be computed as $M = \sum_{\sigma \in \Sigma_n} P(g|\mathcal{G}, \sigma) M_\sigma$, where M_σ is the permutation matrix of σ. However, marginalizing over the set of correspondences in this case is not possible.Several graph matching approaches could be used to estimate M, but here, we propose a self-correcting approach which improves the estimate with each sample.

In order to obtain the initial estimate we assign the matching probability m_{ih} according only to local information from the neighborhoods of nodes i and h. Theoretically this would mean computing $\sum_{\substack{\sigma \in \Sigma_N \\ \sigma(i)=h}} \prod_{j=1}^n \Theta_{i,j}^{h,\sigma(j)}$, and then re-project the matrix using the Sinkhorn projection. However, here we approximate this by projecting $\hat{M} = (\hat{m}_{ih})$, where $\hat{m}_{ih} = min\left(\prod_{k=1}^n \sum_{j=1}^n \Theta_{ij}^{hk}, \prod_{j=1}^n \sum_{k=1}^n \Theta_{ij}^{hk} \right)$. Note that $\hat{m}_{ih} \geq \bar{m}_{ih}$ for all $i, h = 1 \dots n$. Clearly this is a very rough estimate and better ones can be devised, but experimentally we have seen that it is good enough for our purposes.

In order to update the matching estimate, we assume that each row and column are the parameters of a multinomial distribution and perform Bayesian estimation of the matrix using a Dirichlet prior. Hence, the estimation of the matching probability matrix at time t will be

$$M^t = \frac{n_0}{n_0 + t} M^0 + \frac{t}{n_0 + t} \frac{\sum_{i=1}^t \frac{P(g|\mathcal{G}, \sigma^i)}{f(\sigma^i)} M_{\sigma^i}}{\sum_{i=1}^t \frac{P(g|\mathcal{G}, \sigma^i)}{f(\sigma^i)}}, \quad (3)$$

where n_0 is a prior parameter that tells us how confident we are in our initial estimate.

4. Model Estimation

With the correspondence sampler to hand, we can perform a maximum likelihood estimation of the model parameters. The likelihood of model $\mathcal{G}$ given a set of graphs examples S, is:

$$\mathcal{L}(\mathcal{G}) = \sum_{g \in S} \ln \left(\frac{1}{(n-m)!|\Sigma_g|} \sum_{\sigma \in \Sigma_n} P(g|\mathcal{G}, \sigma) \right) \approx$$

$$\approx \sum_{g \in S} \ln \left(\frac{n!}{k_g (n-m)!|\Sigma_g|} \sum_{\sigma \sim f}^{k_g} P(g|\mathcal{G}, \sigma) \frac{\frac{1}{n!}}{f(\sigma)} \right), \quad (4)$$

where with $\sigma \sim f$ we indicate that we sample the correspondence σ with distribution f, and k_g is the number of correspondences sampled for graph g. The derivative of $\mathcal{L}(\mathcal{G})$ with respect to the model parameters is, then:

$$\frac{\partial \mathcal{L}(S)}{\partial \theta_{ij}} = \sum_{g \in S} \frac{\sum_\sigma P(g|\mathcal{G}, \sigma) \frac{\frac{\partial}{\partial \theta_{ij}} \Theta_{ij}^{\sigma(i)\sigma(j)}}{\Theta_{ij}^{\sigma(i)\sigma(j)}}}{\sum_{\sigma \in \Sigma_n} P(g|\mathcal{G}, \sigma)} \approx$$

$$\approx \sum_{g \in S} \frac{\sum_{\sigma \sim f}^{k_g} \frac{P(g|\mathcal{G}, \sigma)}{f(\sigma)} \frac{\frac{\partial}{\partial \theta_{ij}} \Theta_{ij}^{\sigma(i)\sigma(j)}}{\Theta_{ij}^{\sigma(i)\sigma(j)}}}{\sum_{\sigma \sim f}^{k_g} \frac{P(g|\mathcal{G}, \sigma)}{f(\sigma)}}. \quad (5)$$

Starting from equation (5) we can use a gradient ascent method to maximize $\mathcal{L}(\mathcal{G})$ and, hence, learn the generative graph model.

5. Experimental Results

In order to assess the ability of the proposed approach to characterize the intrinsic distribution of structural representations of shape, we performed two sets of experiments with two widely-used graph abstraction of shape. The first abstraction is a Delaunay graph over a set of feature points

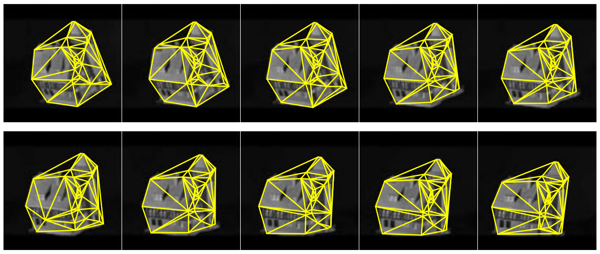


Figure 3. CMU house sequence with the feature points and the Delaunay graphs superimposed.

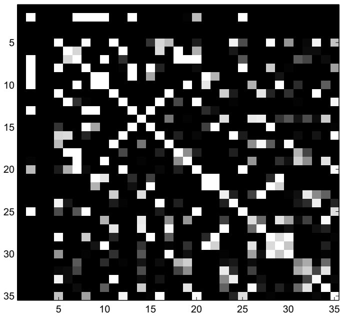
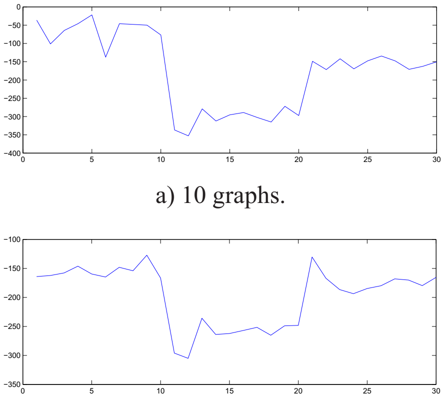


Figure 4. Observation probabilities Θ of the graph model trained on the CMU house sequence.

extracted from an image. Here we have ten images from the CMU house sequence, extract salient points using a corner detector, and connect them according to a Delaunay triangulation. Figure 3 shows the ten images with the feature points and the graphs superimposed.

All the graphs where composed of 30 to 32 nodes. Starting from these, and using only the structural information, we have learned several models ranging from 32 to 40 nodes each. Interestingly, every instance converged to a model with no more than 34 nodes, that is it dove to 0 all but at most 34 node sampling probabilities. Hence, the approach appears to be able to perform its own model-order selection since once the model is large enough to describe the whole data-set, there is no advantage in adding new nodes. Clearly, this is only half of the model selection problem. Note however, that our model nodes can only be removed to generate the observations which forces the learned model to have a representative for every node in every observed graph, be it central to the model or just noise. This forces a lower bound on the size of the model, limiting the available range for model order selection. For this reason in this paper we will not be concerned with model selection issues. However, work is underway to define a model where nodes can be added as well as eliminated, and in that context model selection becomes absolutely essential.

Figure 4 shows the model parameter matrix Θ for the model with the minimum estimated likelihood. The model estimated log-likelihood went from -2594 at initialization to -588 upon convergence, which took approximately 20 minutes in a PC with a 3GHz Pentium 4 CPU. This is a model with 35 nodes, but, as it can be seen, the first, third, and fourth node have converged to zero sampling probability, yielding a 32 node model. Note that the expected model density $D_{\mathcal{G}} = \frac{\sum_{i=1}^{35}\sum_{j=i+1}^{35}\theta_{ii}\theta_{jj}\theta_{ij}}{\sum_{i=1}^{35}\sum_{j=i+1}^{35}\theta_{ii}\theta_{jj}}$ is 1.73, very close to the mean density of the training graphs of 1.77.



a) 10 graphs.

b) First 5 graphs.

Figure 5. Estimated probability for the Delaunay graphs extracted CMU house sequence (graphs 1-10), p-random graphs with the same size and average density (graphs 11-20), and Delaunay graphs of randomly chosen points (graphs 21-30). a) Model trained on the 10 Delaunay graphs from CMU house sequence. b) Model trained only 5 graphs.

In order to assess the ability of the approach to characterize the samples it was trained on, we estimated $P(\hat{g}|\mathcal{G})$ for the ten sample graphs, for ten p-random graphs with similar sizes and expected density equal to the mean density of the graphs, and for ten Delaunay graphs obtained from a similar number of randomly chosen points. Figure 5a plots the logarithm of the estimated probabilities. Clearly the model can distinguish the triangulated structure of Delaunay graphs from random graphs. Further, the probability of the graphs it was trained on have, on average, a much higher probability than other Delaunay triangulations. Note, however, that there are a few cases in which the model is giving a relatively low probability to graphs it was trained on. This, however, happens with graphs with a number of nodes and edges different from the values observable in the majority of the other graphs and with a slightly different edge structure.

Next, in order to assess the generalization capabilities of the approach, we trained a new model using only 5 of the ten graphs and re-computed the estimates of $P(\hat{g}|\mathcal{G})$ for the same 30 graphs. Figure 5b plots the log-probabilities obtained using this model. We can clearly see that the approach generalizes fairly well, with the probabilities dis-

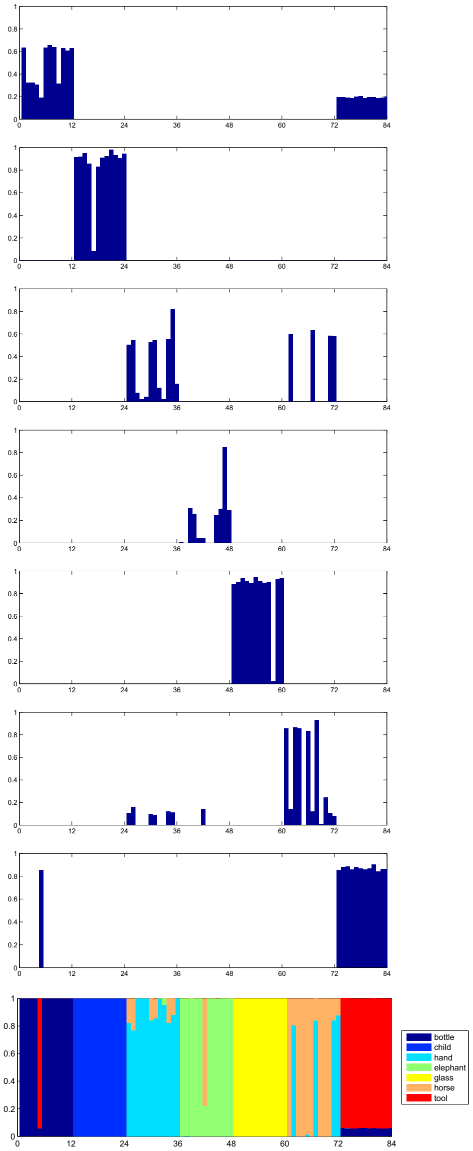


Figure 6. Sampling probability to models learned using all 12 shapes for each class. Top to bottom: bottle-model, child-model, hand-model, elephant-model, glass-model, horse-model, tool-model, and model-assignment probability.

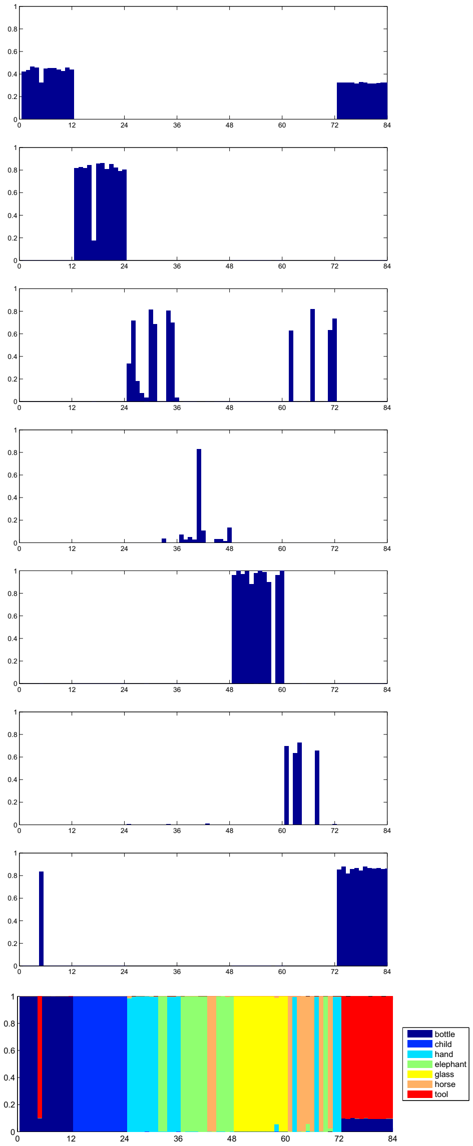


Figure 7. Sampling probability to models learned using only 6 shapes for each class. Top to bottom: bottle-model, child-model, hand-model, elephant-model, glass-model, horse-model, tool-model, and model-assignment probability.

tributing approximately in the same way as those obtained from the full model.

The second graph abstraction of shape we have tested our approach on is the shock-graph, a skeletal-based representation of the differential structure of the boundary of a 2D shape. We have used a database consisting of 84 shapes divided into 7 classes of 12 shapes each. The shape classes where composed of bottles, children, hands, elephants, glasses, horses, and tools. The size of the shock-graphs varied from 4 to 20 nodes. We have learned a model for each shape class, again using only structural information, and computed the sampling probability of each graph from each model. Figure 6 show the resulting distribution of sampling probabilities for each class. With few exceptions, the models shows high probabilities only on graph from the right class. This is especially evident looking at the bottom graph of Figure 6 which shows the model-assignment probability for each graph, i. e., a stacked histogram of the model probabilities normalized over the sum of all model probabilities associated with each graph. Here we can see that in all but 6 shock-graph are predominantly assigned to the correct class. Five of the six errors are due to the relative similarity of the structural part of the shock representations of hands, elephants, and horses: In each case the structure is dominated by 5 main branches, the legs and the head/trunk for the horses/elephants and the five fingers for the hands. The remaining error is due to the structural similarity of the bottle and tool models. Adding attributes to the nodes and edges will likely further improve the recognition rate.

In order to assess the generalization capabilities of the approach we have repeated the experiment using only 6 shapes to learn the models. Figure 7 plots the results. We can see that the reduction in the size of the training sets did not impact the recognition rate by much.

6. Conclusions

In this paper we have proposed an approach to the problem of learning a generative model of structural representations from examples. the approach does not depend on a single estimate of the correspondences between sample graphs and the model, which would induce a bias in the estimate of the model, but rather marginalize the sampling probability over all possible node correspondences. In order to reduce the super-exponential explosion of the set of correspondences, an importance sampling approach is used to estimate the graph observation probability. Experimental results performed both on Delaunay graphs and shock graphs show that the approach is capable of capturing the modes of structural variation present in the data and that it can be used to recognize structures belonging to a same class.

This work can be extended in several directions. First, we should model attributes associated with nodes and edges. This can be done using node observation models like the one presented in [18]. Second, the approach can be used to perform unsupervised classification by fitting a mixture of graph models to the data. Finally, theoretical bonds for the variance of the estimation of the observation probability should be studied.

References

[1] H. G. Barrow and R. M. Burstall, “Subgraph isomorphism, matching relational structures and maximal cliques.” *Inf. Proc. Letters*, 4:83–84, 1976.

[2] I. Beichl and F. Sullivan, “Approximating the permanent via importance sampling with application to the dimer covering problem.” *J. Comput. Phys.* 149(1):128–147, 1999.

[3] B. Bonev et al., “Constellations and the Unsupervised Learning of Graphs.” In *Graph Based representations in Pattern Recognition*, Springer, LNCS Vol. 4538, 2007.

[4] H. Bunke et al., “Graph Clustering Using the Weighted Minimum Common Supergraph.” In *Graph Based Representations in Pattern Recognition*, Springer, pp. 235–246, 2003.

[5] S. J. Dickinson, A. P. Pentlan, and A. Rosenfeld, “3-D shape recovery using distributed aspect matching.” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

[6] P. Erdös and A. Rényi, “Asymmetric graphs.” *Acta Math. Acad. Sci. Hungar.* 14:295–315, 1963.

[7] N. Friedman and D. Koller, “Being Bayesian about Network Structure.” *Machine Learning*, 50(1–2):95–125, 2003.

[8] L. Getoor et al., “Learning Probabilistic models of relational structure.” In *8th Int. Conf. on Machine Learning*, 2001.

[9] M. Hagenbuchner, A. Sperduti, and A.C. Tsoi, “A Self-Organizing Map for Adaptive Processing of Structured Data.” IEEE Trans. Neural Networks, 14:491–505, 2003.

[10] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Wiley, New York, 1964.

[11] D. Heckerman, D. Geiger, and D. M. Chickering, “Learning Bayesian networks: the combination of knowledge and statistical data.” *Machine Learning*, 20(3):197–243, 1995.

[12] Ioffe, S. and Forsyth, D.A., “Human Tracking with Mixtures of Trees.” In *IEEE Comp. Soc. Int. Conf. Computer Vision*, Vol. I, pp. 690-695, 2001.

[13] B. J. Jain and F. Wysotzki, “Central Clustering of Attributed Graphs.” Machine Learning, 56:169–207, 2004.

[14] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, “Shapes, shocks, and deformations I.” *International Journal of Computer Vision*, 15:189–224, 1995.

[15] Liu, T. and Geiger, D., “Approximate Tree Matching and Shape Similarity.” In *IEEE Comp. Soc. Int. Conf. Computer Vision*, pp. 456–462, 1999.

[16] R. Sinkhorn, “A relationship between arbitrary positive matrices and double stochastic matrices.” *Ann. Math. Stat.* 35:876–879, 1964.

[17] S. Todorovic and N. Ahuja, “Extracting Subimages of an unknown category from a set of images.” In *IEEE Comp. Soc. conf. Computer Vision and Pattern Recognition*, Vol. 1, pp. 927–934, 2006.

[18] A. Torsello, E. R. Hancock, “Learning Shape-Classes Using a Mixture of Tree-Unions.” IEEE Trans. Pattern Analysis and Machine Intelligence, 28(6):954–967, 2006.

[19] D. White and R. C. Wilson, “Spectral Generative Models for Graphs.” In *Int. Conf. Image Analysis and Processing*, IEEE Computer Society, pp. 35-42, 2007.

[20] Zhu, S.C. and Yuille, A.L., “FORMS: A Flexible Object Recognition and Modelling System.” *International Journal of Computer Vision*, 20(3):187–212, 1996.