Learning a generative model for structural representations

Andrea Torsello[1] and David L. Dowe[2]

[1] Dipartimento di Informatica
Università Ca' Foscari, Venezia, Italy
[2] Clayton School of Information Technology
Monash University, Clayton, Vic. 3800, Australia

Abstract. Graph-based representations have been used with considerable success in computer vision in the abstraction and recognition of object shape and scene structure. Despite this, the methodology available for learning structural representations from sets of training examples is relatively limited. This paper addresses the problem of learning archetypal structural models from examples. To this end we define 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 from data in a situation where the correspondences between the nodes in the data graphs and the nodes in the model are not known *ab initio* and must be estimated from local structure. This results in an EM-like approach where we alternate the estimation of the node correspondences with the estimation of the model parameters. The former estimation is cast as an instance of graph matching, while the latter estimation, together with model order selection, is addressed within a Minimum Message Length (MML) framework. Experiments on a shape recognition task show the effectiveness of the proposed learning approach.

1 Introduction

Graph-based representations have been used with considerable success in computer vision in the abstraction and recognition of object shape and scene structure. Specific examples include the use of shock graphs to represent shape-skeletons [13], the use of trees to represent articulated objects [12, 22] and the use of aspect graphs for 3D object representation [7]. 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. Despite the many advantages of graph representations, the methodology available for learning structural representations from sets of training examples is relatively limited, and the process of capturing the modes of structural variation for sets of graphs has proved to be elusive. For this reason feature-based geometric representations have been preferred when analyzing variable sets of shapes. There are two reasons why pattern spaces are more easily constructed for feature-based representations than for graphs. First, there is no canonical ordering for the nodes or edges of a graph. Hence, before a vector-space can be constructed, the correspondences between nodes must be established. Second, structural variations in graphs manifest themselves as differences in the numbers of nodes and edges. As a result, even if a vector mapping can be established then the vectors will be of variable length.

There has been considerable interest in learning structural representations from samples of training data in the context of Bayesian networks [8], generalized Bayesian networks [3, 4], or general relational models [9, 19, 5]. However, these models rely on the availability of correspondence information. In many situations, however, the identity of the nodes and their correspondences across samples of training data are not known but must be recovered from the structure typically using graph matching techniques during the learning process. This leads to a chicken and egg problem in structural learning: the correspondences must be available to learn the model and yet the model itself must be known to locate correspondences.

Recently, there has been some effort aimed at learning structural archetypes and clustering data abstracted in terms of graphs even when the correspondences are not known *ab initio*. Hagenbuchner et al. [11] use Recursive Neural Networks to perform unsupervised learning of graph structures. While this approach preserves the structural information present, it does not provide a means of characterizing the modes of structural variation encountered. Bonev et al. [1] and Bunke et al. [2] summarize the data by creating super-graph representation from the available samples, while White and Wilson [21] use a probabilistic model over the spectral decomposition of the graphs to produce a generative model of their structure. These techniques provide a structural model of the samples - however, the way in which the supergraph is learned or estimated is largely heuristic in nature and is not rooted in a statistical learning framework. Torsello and Hancock [16] 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 [14] 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. While these approaches do capture the structural variation present in the data in a way solidly rooted in statistical learning, there are two major problems in the way the model is constructed. First, the model structure and model parameters are tightly coupled, which forces the learning process to be approximated as a series of model merges. Second, all the observed nodes must have a counterpart in the model, which must then account for both the underlying structure as well as the random structural noise observed.

The aim in this paper is to develop an information-theoretic framework for learning of generative models of graph-structures from sets of examples. The

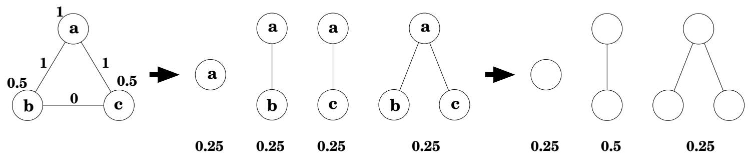


Fig. 1. A core structural model and the generated graphs. When the correspondence information is lost, the second and third graph become indistinguishable.

major characteristics of the model are the fact that the model structure and parameters are decoupled, and that we have two components to the model: one which describes the *core* part, or the proper set of structural variations, and one which defines an isotropic random structural noise.

2 Generative Graph Model

Consider the set of undirected graphs $S = (g_1, \dots, g_l)$. 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 in 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 $V = \{1, \dots, n\}$ is the set of nodes, $E \subseteq V \times V$ is the set of edges and Θ is a set of observation probabilities. In an observation, or sample, from this model, node $i \in V$ is present with probability θ_i, i.e., the existence of each node in a sample graph is modelled as a Bernoulli trial of parameter θ_i. Further, edge (i, j) is present with probability τ_{ij}, conditioned on the fact that both nodes i and j are present.

After the graph has been generated from the model, we lose track of the correspondences between the observation's nodes and the model's nodes 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_i and τ_{ij}. Note that, when the correspondence information (letters in Figure 1) is dropped, we cannot distinguish between the second and third graphs anymore, yielding the final distribution.

This definition applies to unweighted graphs, but it can be generalized to graphs with node or edge attributes by adding a generative model for node- and edge-attributes. Let us assume we have a set of node attributes $\mathcal{A}$ and a set of edge attributes $\mathcal{B}$, an attributed graph is a tuple (g, α, β) where g is a graph, α is a function from the nodes of g to the set of node-attributes $\mathcal{A}$, and β is a function from the set of edges of g to $\mathcal{B}$. A generative model for attributed graphs will then be characterized by the node and edge observation probabilities θ_i and τ_{ij} as well as the node attribute densities $f_i^{\mathcal{A}}$ and the edge attribute density $f_{ij}^{\mathcal{B}}$, so that the probability of observing node i with attribute α_i is $\theta_i f_i^{\mathcal{A}}(\alpha_i)$ and the probability of observing edge (i,j) with attribute β_{ij}, conditioned on the observation of i and j, is $\tau_{ij} f_i^{\mathcal{B}}(\beta_{ij})$.

With this generative model, since every node in the generated graphs must originate from a node in the model, the only structural operation we can perform to generate a new graph is the removal of nodes and edges. This implies that the model must describe every possible structural variation encountered in the data, be it central to the distribution, or simply structural noise that is encountered with very low probability. To avoid this we allow for nodes to be added to the model by saying that, with a certain probability, the model generates nodes that do not correspond to any one represented by the structural part of the model, and that have identical probability $\bar{\tau}$ of being connected to any other node, where we force this probability to be equal to the average density of the core part of the structural model, i.e.,

$$\bar{\tau} = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \theta_i \theta_j \tau_{ij}}{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \theta_i \theta_j} \,.$$

Hence, external nodes model isotropic (or spherical) noise. In general, a generative model will generate a graph with k external nodes according to a geometric distribution $P_k = (1-\bar{\theta})\bar{\theta}^k \prod_{i=1}^{k} f^{\mathcal{A}}(\alpha_i)$, where $\bar{\theta} \in [0,1]$ is a model parameter that quantifies the tendency of the model to generate external nodes and $f^{\mathcal{A}}(\alpha_i)$ is the density of the observed attributes of the external nodes.

Let us 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.

Let g be a graph and $\sigma : (1,\ldots,n) \to (1,\ldots,m+1)$ be a set of correspondences from the model nodes to the nodes in g where $\sigma(i) = m+1$ if model node i has no corresponding node in g, that is, if model node i is not observed in graph g. Further, let $\pi : (1,\ldots,m) \to (1,\ldots,n+1)$ be the inverse set of correspondences, where $\pi(h) = n+1$ if h is an external node, otherwise $\sigma(\pi(h)) = h$, and $\pi(\sigma(i)) = i$ if $\sigma(i) \neq m+1$. With this notation, the probability that a graph g was sampled from a model $\mathcal{G}$ given the correspondences σ and π is

$$P(g|\mathcal{G},\sigma) = (1-\bar{\theta}) \prod_{i=1}^{n} \prod_{j=i}^{n} \Theta_{ij}^{\sigma(i)\sigma(j)} \prod_{h=1}^{m} \prod_{k=h}^{m} \overline{\Theta}_{\pi(h)\pi(k)}^{hk} \,, \qquad (1)$$

where Θ_{ij}^{hk} is the probability that model edge (i,j) generated graph edge (h,k), $\overline{\Theta}_{ij}^{hk}$ with $i = n+1$ or $j = n+1$ is the probability that edge (h,k) is external to the model. Here pairs with the same index represent a node instead of an edge. Letting $G = (g_{hk})$ be the adjacency matrix of graph g, we define Θ_{ij}^{hk} and $\overline{\Theta}_{ij}^{hk}$ as follows:

$$\mathit{\Theta}_{ij}^{hk} = \begin{cases} 0 & \text{if } i = j \wedge h \neq k \text{ or } i \neq j \wedge h = k \\ \theta_i f_i^{\mathcal{A}}(\alpha_h) & \text{if } i = j \wedge h = k \wedge G_{hh} = 1 \\ 1 - \theta_i & \text{if } i = j \wedge h = k \wedge G_{hh} = 0 \\ \tau_{ij} f_{ij}^{\mathcal{B}}(\beta_{hk}) & \text{if } i \neq j \wedge h \neq k \wedge G_{hk} = 1 \\ 1 - \tau_{ij} & \text{if } i \neq j \wedge h \neq k \wedge G_{hh} = 1 \wedge \\ & G_{kk} = 1 \wedge G_{hk} = 0\,, \\ 1 & \text{otherwise.} \end{cases}$$

$$\overline{\mathit{\Theta}}_{ij}^{hk} = \begin{cases} 0 & \text{if } i = j \wedge h \neq k \text{ or } i \neq j \wedge h = k \\ \bar{\theta} f^{\mathcal{A}}(\alpha_h) & \text{if } h = k \wedge i = j = n+1 \\ \bar{\tau} f^{\mathcal{B}}(\beta_{hk}) & \text{if } (i = n+1 \vee j = n+1) \wedge G_{hk} = 1 \\ 1 - \bar{\tau} & \text{if } (i = n+1 \vee j = n+1) \wedge G_{hk} = 0 \\ 1 & \text{otherwise.} \end{cases}$$

3 Model Estimation

Key to the estimation of the structural model is the realization that, conditioned on a given set of correspondences between every node of every graph in S and the nodes of the model $\mathcal{G}$, the node observation processes are independent from one another. Hence, since the structural component of the model is always a complete graph and node/edge observation is dictated by the model parameters, knowing the set of correspondences would effectively decouple parameters and structure.

Here we make the simplifying assumption that the likelihood of the set of correspondences σ_g between graph g and model $\mathcal{G}$ is strongly peaked, i.e., we have $P(g|\mathcal{G}) \approx \max_{\sigma_g} P(g|\mathcal{G}, \sigma_g)$. With this assumption the estimation of the structural model can be achieved with an EM-like process by alternating the estimation of the correspondences σ_g of every graph $g \in S$ with a fixed set of model parameters $\mathit{\Theta}$, and the estimation of $\mathit{\Theta}$ given the correspondences.

While this EM-like approach solves the problem of estimating the structural model of a given size, the problem of model order selection remains open. We have chosen to use Minimum Message Length (MML) criterion [18, 17], which allows us to address parameter estimation and model order selection within a single framework, solidly basing it on information-theoretic principles.

3.1 Correspondence Estimation

The estimation of the set of correspondences σ is an instance of a graph matching problem, where, for each graph g, we are looking for the set of correspondences that maximizes $P(g|\mathcal{G}, \sigma)$. To do this we relax the space of partial correspondences, where a relaxed state is represented by a matrix $P = (p_{ih} \in [0, 1])$ where $i = 1 \dots n+1$ iterates over the model nodes, with $i = n+1$ representing external nodes, and $h = 1 \dots m+1$ iterates over the nodes of g, with $j = m+1$ representing non-observed nodes. The matrix P satisfies the constraints

$$\begin{array}{l} x_{ih} \geq 0 \text{ for all } i = 1 \dots n \text{ and } h = 1 \dots m \\ \sum_{h=1}^{m+1} p_{ih} = 1 \text{ for all } i = 1 \dots n \\ \sum_{i=1}^{n+1} p_{ih} = 1 \text{ for all } h = 1 \dots m\,. \end{array}$$

Note that, with the exception than the last row and column that are not normalized, the matrix P is almost doubly stochastic, i.e., the sum of the elements in each row and in each column is equal to one. The probability $P(g,\mathcal{G},\sigma)$ can be extended to the relaxed assignment space as the function

$$E(g,\mathcal{G},P) = (1-\bar{\theta})\Big(\prod_{i=1}^{n}\prod_{j=i}^{n}\sum_{h=1}^{m+1}\sum_{k=h}^{m+1} p_{ih}\Theta_{ij}^{hk}p_{jk}\Big)\Big(\prod_{h=1}^{m}\prod_{k=i}^{m}\sum_{i=1}^{n+1}\sum_{j=i}^{n+1} p_{ih}\bar{\Theta}_{ij}^{hk}p_{jk}\Big) .$$

In an approach similar to Graduated Assignment [10], we maximize the energy function E by iterating the recurrence $P^{t+1} = \mu\big(DE^{t}\big)$, where $DE^{t} = (de_{ih})$ is the differential of E with respect to P^{t} and satisfies

$$\frac{de_{ih}}{E(g,\mathcal{G},P^{t})} = \left(\sum_{j=1}^{n}\frac{\sum_{k=1}^{m+1}\Theta_{ij}^{hk}p_{jk}}{\sum_{l=1}^{m+1}\sum_{k=1}^{m+1}p_{il}\Theta_{ij}^{lk}p_{jk}}\right)\left(\sum_{k=1}^{m}\frac{\sum_{j=1}^{n+1}\overline{\Theta}_{ij}^{hk}p_{jk}}{\sum_{l=1}^{n+1}\sum_{j=1}^{n+1}p_{lh}\overline{\Theta}_{lj}^{hk}p_{jk}}\right),$$

and μ is a function projecting DE^{t} to the relaxed assignment space. The projection of a matrix P is obtained by searching for the relaxed partial assignments that minimizes the Frobenius distance $||P-P^{*}||_{F}$. The minimization can be performed by iteratively projecting P to the set Ω satisfying the equality constraints

$$\begin{array}{l}\sum_{h=1}^{m+1} p_{ih} = 1 \text{ for all } i = 1\dots n\\ \sum_{i=1}^{n+1} p_{ih} = 1 \text{ for all } h = 1\dots m\end{array}$$

and then projecting it on to the conic subspace $p_{ih} \geq 0$.

The projection to the conic subspace is done by setting to 0 all negative entries of P, while the projection to Ω will be of the form $P^{*} = P - \alpha\mathbf{e}_{m}^{T} - \mathbf{e}_{n}\beta^{T}$, where $\mathbf{e}_{k}$ is the $(k+1)$-dimensional vector with the first k entries equal to 1 and the last equal to 0, and α and β are defined as follows:

$$\begin{aligned}&\alpha_i = (P\mathbf{e}_m)_i + (p_{im+1}-1) -\\ &\frac{\mathbf{e}_n^{T}P\mathbf{e}_m + (m+1)\big(P_{n+1}\mathbf{e}_m - m\big) - m\big(\mathbf{e}_n^{T}P^{m+1} - n\big)}{m+1},\\ &\beta_h = (\mathbf{e}_n^{T}P)_i + (p_{n+1,h}-1) -\\ &\frac{\mathbf{e}_n^{T}P\mathbf{e}_m + (n+1)\big(\mathbf{e}_n^{T}P^{m+1} - n\big) - m\big(P_{n+1}\mathbf{e}_m - m\big)\big)}{n+1},\end{aligned}$$

where P_i and P^j refer to the ith row and jth column respectively.

Finally, once we have found the maximizer $P^{\infty} = \operatorname{argmax}_P E(g,\mathcal{G},P)$, we map it to the closest 0-1 matrix by solving a bipartite matching problem.

3.2 Parameter Estimation

The parameter estimation and model selection problem are tightly coupled. For this reason we have chosen to use Minimum Message Length (MML) [17], which has the ability to deal comfortably with hybrid discrete and continuous models, including model order selection. MML is a Bayesian method of point estimation based on an information-theoretic formalization of Occam's razor. Here, simplicity of an explanation is formalized as the joint cost of describing a probabilistic model for the data and describing the data given the model. Hence, to estimate a model class and the model parameters, MML constructs a two-part message. The first encodes the model class/order and the parameters, while the second assumes a Shannon-optimal encoding of the data given the model. According to the MML criterion, we choose the model class/order and the parameter estimate that correspond to the shortest two-part message. MML is closely related to the Kolmogorov complexity [19, 17], is invariant under 1-to-1 parameter transformations [20, 17], and has general statistical consistency properties [6, 5].

The cost of describing a fully specified model (in the first part of the message) with a parameter vector $\theta_{\mathcal{G}}$ is approximately

$$-\log\left[\frac{h(\theta_{\mathcal{G}})}{\sqrt{k_D^D F(\theta_{\mathcal{G}})}}\right] = -\log\left[\frac{h(\theta_{\mathcal{G}})}{\sqrt{F(\theta_{\mathcal{G}})}}\right] + \frac{D}{2}\log k_D,$$

where D is the number of parameters of the model, k_D are the lattice constants specifying how tightly unit spheres can be packed in a D-dimensional space, $h(\theta)$ is the prior of the parameters θ, $F(\theta)$ is the Fisher information matrix and the term $1/(\sqrt{k_D^D F(\theta_{\mathcal{G}})})$ is the optimal round-off in the parameter estimates. It is this round-off which gives rise to the additional term of $D/2$ in the second part of the message below.

According to Shannon's theorem, the cost of encoding the data (in the second part of the message) has a tight lower bound in the negative log-likelihood function, to which - as immediately above - we add $D/2$.

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

If D is sufficiently large the logarithm of the lattice constants can be approximated as $\log(k_D) = \frac{\log(\pi D)-2}{D} - \log(2\pi) - 1$ [17].

In this work we have opted for a standard non-informative Jeffreys's prior for the model parameters which will push the parameters towards the edges of their range forcing each node/edge to be observed either very frequently or very rarely. A consequence of this choice is that the MML point estimates of the parameters are the same as the maximum likelihood estimates, leaving the MML criterion only for model-order selection. (A more general but more CPU-intensive alternative would be to generalise the Jeffreys prior by having a hyper-parametric prior of the form in [5, sec. 0.2.6].) In fact, the use of Jeffreys's prior implies $h(\theta) = \sqrt{F_1(\theta)}$, where $F_1(\theta)$ is the single datum Fisher information matrix and $F(\theta) = |S|^D F_1(\theta)$. Hence, the final message (or code) length, considering the approximation for $\log(k_D)$, is

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

where $|S|$ is the number of samples and the number of parameters for a n-node structural model is $D = \binom{n}{2} + n + 1$

Further, we have $\theta_i = \frac{a_i}{|S|}$, and $\tau_{ij} = \frac{|\{g\in S|(\sigma_g(i),\sigma_g(j))\in E_g\}|}{a_{ij}}$, where a_i is the number of graphs that observe model node i, a_{ij} is the number of graphs that observe both nodes i and j, and $\bar{\theta} = \frac{u}{u+|S|}$, where u is the set of external nodes that do not map to any node in the model. Similarly, all other per-node and per-edge parameters specifying the attribute models are estimated using maximum likelihood estimations.

Concluding, given a set of observation graphs S and a model dimension n, we jointly estimate node correspondences and model parameters by alternating the two estimation processes in an EM-like approach, and then we chose the model order that minimizes the message length, I_1.

4 Experimental Evaluation

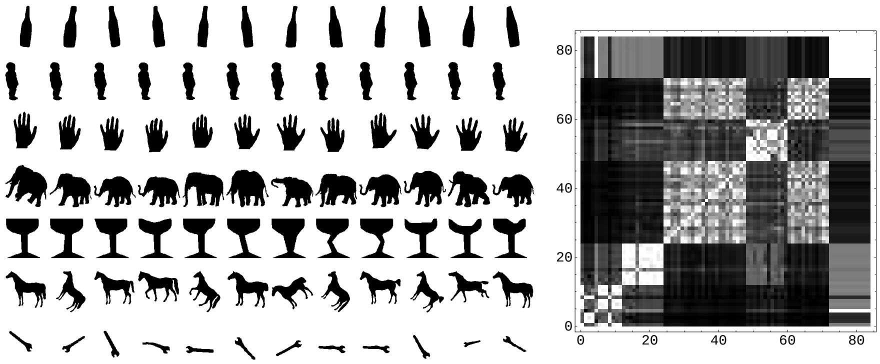


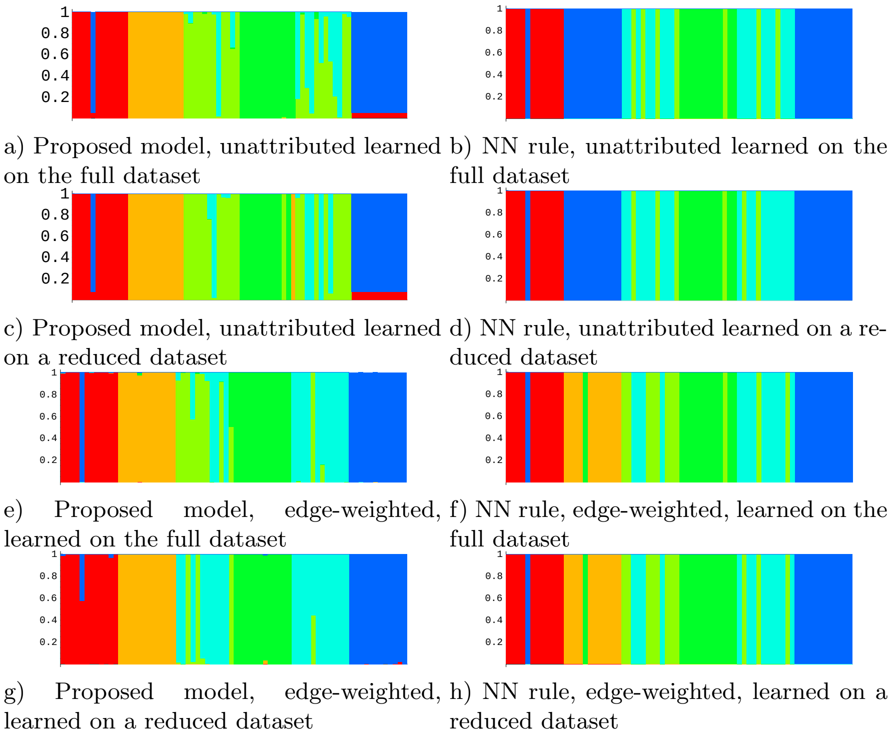
Fig. 2. The shape classes in the database.

We tested our structural learning approach on shock-graphs [13], a skeletal-based representation of the differential structure of the boundary of a 2D shape. We have used a database consisting of 72 shapes from the MPEG7 database, divided into 6 classes of 12 shapes each. The shape classes where composed of bottles, children, elephants, glasses, and tools. Figure 2 shows the shapes in the database (left) and their distance matrix computed using Graduated Assignment. The size of the resulting shock-graphs varies from 4 to 20 nodes. We have learned a model for each shape class, first using structural information only, and then adding attributes to the edges measuring the proportion of boundary linked to the skeletal branch [15]. In the latter case we assumed a Gaussian distribution for the edge attributes, and learned the attributes' means μ_{ij} and variances σ_{ij}^2 together with the observation probabilities τ_{ij}. For comparison, we also computed the structural similarities using Graduated Assignment [10].

In order to assess the ability of the approach to characterize the samples it was trained on, we computed the probability $P(g|\mathcal{G})$ for every shock-graph and every estimated model and assigned each graph to the model with maximum probability of generating it. Figure 3 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 the colour of the bars represent the classes, while their length is proportional to $\frac{P(g|\mathcal{G}_i)}{\sum_j P(g|\mathcal{G}_j)}$, the assignment probability of graph g to model G_i. Figure 3a shows the assignment of graphs to classes according to the proposed approach, while Figure 3b plots the assignments obtained using the nearest neighbour (NN) rule based on the distances obtained with Graduated Assignment. Here we can see that in most cases shock-graphs are predominantly assigned to the correct class,while NN has a slightly higher rate of misclassifications of 17% versus the 10% misclassification we obtained with our approach. Furthermore, it should be noted that NN classification is computationally more demanding than the classification using our structural models, as the computation of the similarity between two graphs using Graduated Assignment and the computation of $\max_\sigma \big(P(g,\mathcal{G},\sigma)\big)$ have the same computational complexity, but NN requires computing the similarity against each training graph, while our approach requires computing the probabilities only against the learned models. Clearly our approach requires the models to be learned ahead of time, but that can be performed offline.

Further, to assess the generalization capabilities of the approach we have repeated the experiment using only 6 shapes to learn the models. Figures 3c and 3d plot the model assignments obtained using our approach and the NN rule respectively. We can clearly see that the approach generalizes fairly well in both cases, with the probabilities approximately distributed in the same way as those obtained from the full training set, resulting in a 15% misclassification for our approach and 18% for NN classification.

Figures 3e, 3f, 3g, and 3h plot the assignments obtained using edge-weighted models learned on the full and reduced training set respectively. Here we see that the additional information allows for a much improved recognition performance, with both approaches improving the recognition rate and with the proposed approach maintaining the marginal advantage over the NN classification. The misclassification rates were 7% for our approach on the full database versus 14% obtained using the NN rule. With the reduced training set we obtained 13% misclassification rate versus 14% for the NN rule.



a) Proposed model, unattributed learned on the full dataset

b) NN rule, unattributed learned on the full dataset

c) Proposed model, unattributed learned on a reduced dataset

d) NN rule, unattributed learned on a reduced dataset

e) Proposed model, edge-weighted, learned on the full dataset

f) NN rule, edge-weighted, learned on the full dataset

g) Proposed model, edge-weighted, learned on a reduced dataset

h) NN rule, edge-weighted, learned on a reduced dataset

Fig. 3. Assignment probability of the graphs to the learned models. The colors of the classes are as follows: Bottle (red), Child (orange), Hand (light green), Glass (dark green), Horse (light bue), and Tool (dark blue).

5 Conclusions

In this paper we have proposed an approach to the problem of learning a generative model of structural representations from examples in a situation where the correspondences must be estimated from local structure. To this end, we defined a structural model where the distribution of observed nodes and edges is governed by a set of independent Bernoulli trials. The model is learned using an EM-like approach where we alternate the estimation of the node correspondences using a graph matching approach, with the estimation of the model parameters which, together with model order selection, is addressed within a Minimum Message Length (MML) framework. Experiments on a shape recognition task show that the approach is effective in characterizing the modes of structural variation present in a set of graphs.

Given the merits of log-loss probabilistic scoring over right/wrong accuracy [5, footnote 175], it is our hope to later re-visit our experimental results from sec. 4 using log-loss probabilistic scoring.

References

1. B. Bonev et al. (2007). Constellations and the Unsupervised Learning of Graphs. In *Graph Based representations in Pattern Recognition*, Springer, LNCS Vol. 4538.
2. H. Bunke et al. (2003). Graph Clustering Using the Weighted Minimum Common Supergraph. In *Graph Based Representations in Pattern Recognition*, Springer, pp. 235–246.
3. J. W. Comley and D. L. Dowe (2003). General Bayesian networks and asymmetric languages. In *Proc. Hawaii International Conference on Statistics and Related Fields.*
4. J. W. Comley and D. L. Dowe (2005). Minimum message length and generalized Bayesian nets with asymmetric languages. In P. Grünwald, M. A. Pitt, and I. J. Myung, eds., *Advances in Minimum Description Length: Theory and Applications (MDL Handbook)*, pp. 265–294. M.I.T. Press.
5. D. L. Dowe (2008), Foreword re C. S. Wallace, *Computer Journal*, 51(5):523–560.
6. D. L. Dowe, S. Gardner, and G. R. Oppy (2007). Bayes not bust! Why simplicity is no problem for Bayesians. *British J. for the Philosophy of Science*, 58(4):709–754.
7. S. J. Dickinson, A. P. Pentland, and A. Rosenfeld (1992). 3-D shape recovery using distributed aspect matching. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):174–198.
8. N. Friedman and D. Koller (2003). Being Bayesian about Network Structure, *Machine Learning*, 50(1-2):95–125.
9. L. Getoor, N. Friedman, D. Koller, and B. Taskar (2001) Learning Probabilistic models of relational structure, in *8th Int. Conf. on Machine Learning*, pp. 170–177.
10. S. Gold and A. Rangarajan (1995). A graduated Assignment Algorithm for Graph Matching. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(4):377–388.
11. M. Hagenbuchner, A. Sperduti, and A.C. Tsoi (2003). A Self-Organizing Map for Adaptive Processing of Structured Data. IEEE Trans. Neural Networks, 14:491–505.
12. S. Ioffe and D. A. Forsyth (2001). Human tracking with mixtures of trees. In *Proc. Int. Conf. Computer Vision*, Vol. I, pp. 690–695.
13. B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker (1995). Shapes, shocks, and deformations I: the components of shape and the reaction-diffusion space. *Int. J. Computer Vision*, vol. 15, no. 3, pp. 189–224.
14. S. Todorovic and N. Ahuja (2006). 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.
15. A. Torsello and E. R. Hancock (2004). A Skeletal Measure of 2D Shape Similarity. *Computer Vision and Image Understanding*, 95(1):1-29.
16. A. Torsello and E. R. Hancock (2006). Learning Shape-Classes Using a Mixture of Tree-Unions. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(6):954-967.
17. C. S. Wallace (2005). *Statistical and Inductive Inference by Minimum Message Length*. Information Science and Statistics. Springer Verlag, ISBN 0-387-23795-X.
18. C. S. Wallace and D. M. Boulton. An information measure for classification. *Computer Journal*, 11(2):185–194, 1968.
19. C. S. Wallace and D. L. Dowe (1999). Minimum message length and Kolmogorov complexity. *Computer Journal*, 42(4):270–283.
20. C. S. Wallace and P. R. Freeman (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society series B*, 49(3):240–25. (See also Discussion on pp. 252–265.)

21. D. White and R. C. Wilson (2007). Spectral Generative Models for Graphs. In *Int. Conf. Image Analysis and Processing*, IEEE Computer Society, pp. 35–42.
22. S. C. Zhu and A. L. Yuille (1996). FORMS: A flexible object recognition and modelling system. *Int. J. Computer Vision*, vol. 20, no. 3, pp. 187–212.