# Learning Structure from Samples

Andrea Torsello
Dipartimento di Informatica
Universit Ca' Foscari di Venezia, Italy
E-mail : torsello@dsi.unive.it

**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 statistical models of graph structure. The key problem faced is that of lack of node correspondences which must be inferred together with the structural model. Here classic latent variable approaches cannot be applied due to the breaking up of the independence assumption between the node assignments, while structural pattern recognition typically opts for a maximum likelihood correspondence estimation which induces bias in the estimation. We present a very simple node-observation model, which can be seen as a generalization of the naive Bayes model to the graph domain, and an EM-like approach to learn a mixture of such models, where the exponential explosion of the E step is handled through a sampling approach and a Minimum Message Length criterion is used for model selection.

## 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, as they can concisely capture the relational arrangement of object primitives, in a manner which can be invariant to changes in object viewpoint. Despite their many advantages and attractive features, 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.

Recently, there has been considerable interest in learning structural representations from samples of training data, in particular in the context of Bayesian networks, or general relational models [5]. The idea is to associate random variables with the nodes of the structure and to use a structural learning process to infer the stochastic dependency between these variables. However, these approaches rely on the availability of correspondence information for the nodes of the different structures used in learning. In many cases the identity of the nodes and their correspondences across samples of training data are not known, rather, the correspondences must be recovered from structure.

In the last few years, there has been some effort aimed at learning structural archetypes and clustering data abstracted in terms of graphs. 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 [11] 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 supergraph is learned or estimated is largely heuristic in nature and is not rooted in a statistical learning framework. Torsello and Hancock [8] define 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 and is critically dependent on the order in which trees are merged. Further, the model structure and model parameter are tightly coupled, which forces the learning process to be approximated through a series of merges, and all the observed nodes must be explicitly represented in the model, which then must specify in the same way proper structural variations and random noise. The latter characteristic limits the generalization capabilities of the model. Torsello [9] recently proposed a generalization for graphs which allowed to decouple structure and model parameters and used a stochastic process to marginalize the set of correspondences, however the approach does not deal with attributes and all the observed nodes still need be explicitly represented in the model. Further, the issue of model order selection was not addressed. Torsello and Dowe [10] addressed the generalization capabilities of the approach by adding to the generative model the ability to add nodes, thus not requiring to model explicitly isotropic random noise, however correspondence estimation in this approach was cumbersome and while it used a minimum message length principle for selecting model-complexity, that could be only used to choose from different learned structures since it had no way to change the complexity while learning the model.

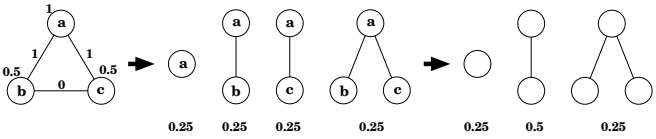


Figure 1. A structural model and the generated graphs. When the correspondence information is lost, the second and third graph become indistinguishable.

## 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 model is a mixture of naïve models where observation of each node and each edge is independent of the others, thus imposing a conditional indepen-

dence assumption similar to naïve Bayes classifier, but allowing correlation to pop up by mixing the models.

The naïve graph model $\mathcal{G}$ is composed by a structural part, i.e., a graph $G = (V, E)$, and a stochastic part. The structural part encodes the structure, here $V$ are all the nodes that can be generated directly by the graph, and $E \subseteq V \times V$ is the set of possible edges. The stochastic part, on the other hand, encodes the variability in the observed graph. To this end we have a series of Bernoulli variables $\theta_i$ associated with each node and $\tau_{ij}$ associated with each edge, which give us respectively the probability that the corresponding node is generated by the model, and the probability that the corresponding edge is generated, conditioned on the generation of both endpoints. Further, to handle node- and edge-attributes, we assume the existence of generative models $W_i^n$ and $W_{i,j}^e$ that model the observable node and edge attribute respectively, and that are parametrized by the (possibly vectorial) quantities $\omega_i^n$ and $\omega_{i,j}^e$. In this way the generation of a graph from a naïve model is as follows: First we sample from the node Bernoulli trials $\theta_i$ which nodes are observed, then we sample the Bernoulli trials $\tau_{i,j}$ for all the edges between the observed nodes, and finally we sample the attributes $W_i^n$ and $W_{i,j}^e$ for all observed nodes and edges, thus obtaining the full attributed graph.

Clearly this approach can generate only graphs with fewer or equal nodes than $V$. This limitation limits the generalization capability of the model and forces one model even random isotropic noise explicitly. To correct this we add to the model the ability to generate nodes and edges not explicitly modeled by the core model. This is obtaining by enhancing the stochastic model with an external node observation model that samples a number of random *external*, i.e., nodes not explicitly modeled in the generative model. The number of external nodes generated is assumed to follow a geometric distribution of parameter $1 - \bar{\theta}$, while the probability of observing edges that have external nodes as one of the endpoints is assumed to be the result of a Bernoulli trial with a common observation probability $\bar{\tau}$. Further, we assume common attribute models $\bar{W}^n$ and $\bar{W}^e$ for external nodes and edges, parametrized by the quantities $\bar{\omega^n}$ and $\bar{\omega^e}$. This way external nodes allow us to model random isotropic noise in a compact way.

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 model is unattributed with null probability of generating external nodes. The numbers next to the nodes and edges of the model represent the values of $\theta_i$ and $\tau_{i,j}$ respectively. 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.

If we knew the correspondences $\sigma_g$ mapping the nodes of graph $g$ to the nodes of the model $\mathcal{G}$, we could very easily compute the probability of observing graph $g$ from model $\mathcal{G}$:

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

where the indexes $i \in V$ and $(i, j) \in E$ indicate product over the internal nodes and edges, while, with an abuse of the formalism, we write $i \notin V$ and $(i, j) \notin E$ to refer to external nodes and edges. With the ability to compute the probability of generating any graph from the model, we can compute the complete data likelihood and do maximum likelihood estimation of the model $\mathcal{G}$, however, here we are interested in the situation where the correspondences are not known and must be inferred from the data as well.

Almost invariably, the 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.

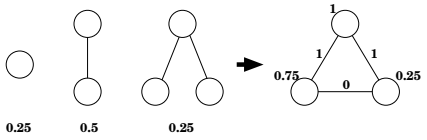


Figure 2. Model estimation bias. If a single node correspondence is taken into account the estimated model will exhibit a bias towards one of multiple possible correspondences.

To solve this bias Torsello [9] proposed to marginalize the sampling probability over all possible correspondences, hence obtaining the probability

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

where $\hat{g}$ is is the quotient of $g$ modulo permutation of its nodes, i.e., the representation of $g$ where the actual order of the nodes is ignored, $\Sigma_n^m$ is the set of all possible partial correspondences between the $m$ nodes of graph $g$ and the $n$ nodes of model $\mathcal{G}$, and $\Sigma_g$ is the set of symmetries of $g$, i.e., the set of graph isomorphisms from $g$ onto itself.

Clearly, averaging over all possible correspondences is not possible due to the super-exponential growth of the size of

$\Sigma_n^m$; hence, we have to resort to an estimation approach. In [9] was proposed an importance sampling approach to compute a fast-converging estimate of $P(g|\mathcal{G})$. Note that similar importance sampling approaches marginalizing over the space of correspondences have been used in [2] and [6]. In particular, in latter work the authors show that the estimation has expected polynomial behavior.

### 2.1 Correspondence Sampler

In order to estimate $P(g|\mathcal{G})$, and to learn the graph model, we need to sample correspondences with probability close to the posterior $P(\sigma|g,\mathcal{G})$. Here we generalize the approach in [9] for models with external nodes Assume that we know the node-correspondence matrix $M = (m_{ih})$, which gives us the marginal probability that model node $i$ corresponds to graph node $h$. Note that, since model nodes can be not observed and graph nodes can be external, we have that $\forall h, \sum_i m_{ih} \leq 1$ and $\forall i, \sum_h m_{ih} \leq 1$. We turn the inequalities into equalities by extending the matrix $M$ into a $(n+1)\times(m+1)$ matrix $\bar{M}$ adding $n+m$ slack variables, where the first $n$ elements of the last column are linked with the probabilities that a model node is not observed, the first $m$ elements of the last row are linked with the probability that an observed node is external and element at index $n+1, m+1$ is unused. $\bar{M}$ is a partial doubly-stochastic matrix, i.e., its first $n$ rows and its first $n$ columns add up to one.

With this marginal node-correspondence matrix to hand, we can sample a correspondence as follows: First we can sample the correspondence for model node 1 picking a node $h_1$ with probability $m_{1,h_1}$. Then, we 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 $\bar{m}_{j,k}$ by $P(g_{h_1,k}|\mathcal{G}_{1,j})$, i.e., the probability that the edges/non-edges between $k$ and $h_1$ map to the model edge $(1,j)$. The multiplied matrix is then projected to a double-stochastic matrix $\bar{M}_1^{h_1}$ using a Sinkhorn projection [7] adapted to partial doubly-stochastic matrix, where the alternate row and column normalization is performed only on the first $n$ rows and $m$ columns. We can then sample a correspondence for model node 2 according to the distribution of the second row of $M_1^{h_1}$ and compute the conditional matching probability $\bar{M}_{1,2}^{h_1,h_2}$ in much the same way we computed $M_1^{h_1}$. and iterate until we have sampled a complete set of correspondences, obtaining a fully deterministic conditional matching probability $\bar{M}_{1,\ldots,n}^{h_1,\ldots,h_n}$, corresponding to a correspondence $\sigma$, that has been sampled with probability $P(\sigma) = (\bar{M})_{1,h_1} \cdot (\bar{M}_1^{h_1})_{2,h_2} \cdot \ldots \cdot (\bar{M}_{1,\ldots,n-1}^{h_1,\ldots,h_{n-1}})_{n,h_n}$.

### 2.2 Estimating the Model

With the correspondence samples to hand, we can easily perform a maximum likelihood estimation of each model parameter by observing that, by construction of the model, conditioned on the correspondences the node and edge observation are independent to one another. Thus, we need only to maximize the node and edge models independently, ignoring

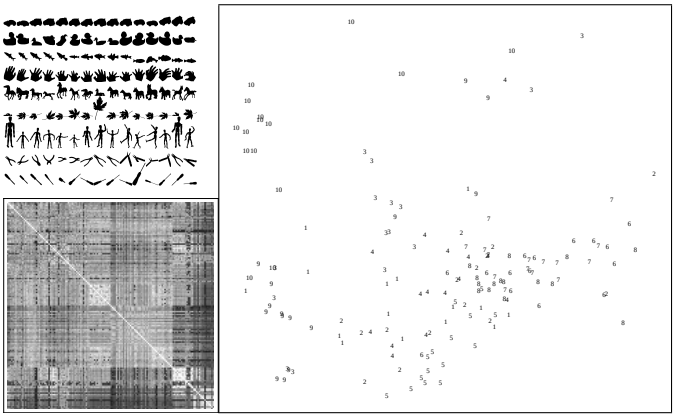

Figure 3. Left column: Top, shape database; bottom, edit distance matrix. Right column: Multidimensional Scaling of the edit distances.

what is going on in the rest of the graph. Thus, we define the sampled node and edge likelihood functions as

$$\mathcal{L}_i(S,\mathcal{G}) = \prod_{g\in S}\sum_{\sigma}\frac{P(g_{\sigma(i)}|\theta_i,\omega_i^n)}{P(\sigma)}$$

$$\mathcal{L}_{i,j}(S,\mathcal{G}) = \prod_{g\in S}\sum_{\sigma}\frac{P(g_{\sigma(i),\sigma(j)}|\tau_{i,j},\omega_{i,j}^e)}{P(\sigma)}$$

from which we can easily obtain maximum likelihood estimates of the parameters $\theta_i$, $\omega_i^n$, $tau_{i,j}$, and $\omega_{i,j}^e$.

Further, we can use th samples to update the initial node-correspondence matrix in the following way

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

where $M_\sigma$ is the deterministic correspondence matrix associated with $\sigma$. Thus in our learning approach we start with a initial guess for the node-correspondence matrix and improve on it as we go along. In all our experiments we initialize the matrix based only on local node information, i.e. $m_{i,h}$ is equal the probability that model node $i$ generates the attributes of graph model $h$.

The only thing left to estimate is the value of $|\Sigma_g|$, but that can be easily obtained using our sampling approach observing that it is proportional to the probability of sampling an isomorphism between $g$ and a deterministic model obtained from $g$ by setting the values of $tau_{i,j}$ to 1 or 0 according the existence of edge $(i,j)$ in $g$, and setting $\bar{\theta} = 0$. It interesting to note that in this corner case, our sampling approach turns out to be exactly the same sampling approach used in [1] to show that the graph isomorphism problem can be solved in polynomial time. Hence, our sampling approach is expected polynomial for deterministic model. and we can arguably be confident that it will perform similarly well for low entropy models.

### 2.3 Model Selection

Given this sampling machinery to perform maximum likelihood estimation of the model parameters for the naïve mod-

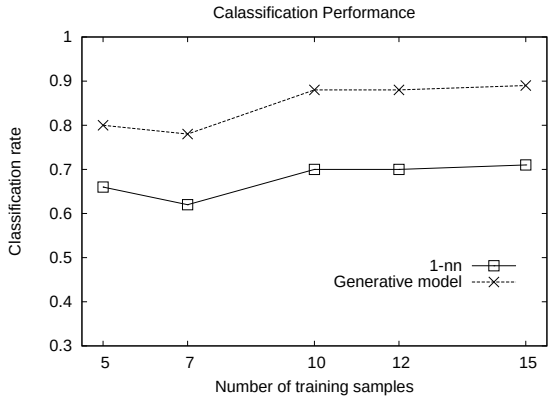


Figure 4. Classification rate of the nearest neighbor rule applied to the edit-distance between shock graphs and of the Bayes decision rule based on the generative models.

els, we adopt a standard EM approach to learn mixtures of naïve models.

This, however, leaves us with a model selection problem, since model likelihood decreases with the number of mixture components as well as with the size of the naïve models. To solve this problem we follow [10] in adopting a minimum message length approach to model selection, but we deviate from it in that we use the message length to prune an initially oversized model.

Thus we seek to minimize the combined cost of a two part message resulting in the penalty function

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

where $|S|$ is the number of samples and $D$ the number of parameters for the structural model.

The pruning strategy adopted is a greedy one, where after each iteration we check if the objective can be reduced by removing a node or a mixture component, and perform the model modification that promises the largest reduction in $I_1$.

## 3. Experimental Evaluation

We experimented on learning models for shock graphs, a skeletal based representation of shape. We extracted graphs from a database composed of 150 shapes divided into 10 classes of 15 shapes each. Each graph had a node attribute that reflected the size of the boundary feature generating the corresponding skeletal segment. Our aim is to compare the classification results obtained learning a generative model to what can be obtain using standard graph matching techniques and a nearest neighbor classifier. Figure 3 shows the shape database, the matrix of extracted edit distances between the shock graphs, and a multidimensional scaling representation of the distances; here numbers correspond to classes. As we can see, recognition based on this representation is a hard problem, as the class structure is not very clear in these distances and there is considerable class overlap.

In Figure 4 we compare the classification performance obtained with the nearest neighbor rule with the one obtained by learning the generative models and using Bayes decision rule for classification, i.e., assigning each graph to the class of the model with largest probability of generating it. Note that the graphs are never classified with a model the had the same graph in the training set, thus in the case of the 15 training samples, the correct class had only 14 samples, resulting in a leave-one-out scheme. Figure 4 shows a clear improvement classification accuracy, consistently of about 15% regardless the number of samples in the training set, thus proving that learning the modes of structural variation present in a class rather than assuming an isotropic behavior with distance, as has been done for 40 years in structural pattern recognition, gives a clear advantage.

## 4. Conclusions

In this paper we have addressed to problem of learning a generative model for graphs from samples. The model is based on a naïve node independence assumptions, but mixes such simple models in order to capture node correlation. The correspondences are estimated using a fast sampling approach, the node and edge parameters are then learned using maximum likelihood estimates, while model selection adopts a minimum descriptor length principle.

Experiments show that learning the graph structure gives a clear advantage over the isotropic behavior assumed by the vast majority of the approaches in the structural pattern recognition literature.

### References


[1] L. Babai, P. Erdös, S. M. Selkow, "Random Graph Isomorphism." *SIAM J. Comput.* 9(3):638–635, 1980.

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

[6] M. G. Rabbat, M. A. T. Figueiredo, and R. D. Nowak, "Network Inference From Co-Occurrences." *IEEE Trans. Information Theory*, 54(9):4053–4068, 2008.

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

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

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

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

[11] 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.