Supervised Learning of Graph Structure

Andrea Torsello and Luca Rossi

Dipartimento di Scienze Ambientali, Informatica e Statistica,
Università Ca' Foscari Venezia, Italy
{torsello,lurossi}@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. In this paper we take a simple yet effective Bayesian approach to attributed graph learning. We present a naïve node-observation model, where we make the important assumption that the observation of each node and each edge is independent of the others, then we propose an EM-like approach to learn a mixture of these models and a Minimum Message Length criterion for components selection. Moreover, in order to avoid the bias that could arise with a single estimation of the node correspondences, we decide to estimate the sampling probability over all the possible matches. Finally we show the utility of the proposed approach on popular computer vision tasks such as 2D and 3D shape recognition.

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 [6]. 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. In this context

M. Pelillo and E.R. Hancock (Eds.): SIMBAD 2011, LNCS 7005, pp. 117–132, 2011.

spectral approaches have provided simple and effective procedures. For example Luo and Hancock [8] use graph spectral features to embed graphs in a low dimensional space where standard vectorial analysis can be applied. While embedding approaches like this one preserve the structural information present, they do not provide a means of characterizing the modes of structural variation encountered and are limited by the stability of the graph's spectrum under structural perturbation. 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 [18] 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 [14] 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 [15] 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 [16] 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.

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 independence 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 binary random variables θ_i associated with each node and τ_{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 ω_i^n and $\omega_{i,j}^e$. Note that θ_i and W_i^n need not be independent, nor do τ_{ij} and $W_{i,j}^e$. With this formalism, the generation of a graph from a naïve model is as follows: First we sample from the node binary indicator variables θ_i determining which nodes are observed, then we sample the variables $\tau_{i,j}$ indicating which edges between the observed nodes are generated, 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 constraint limits the generalization capability of the model and forces one to model explicitly even the observed random isotropic noise. To correct this we add the ability to generate nodes and edges not explicitly modeled by the core model. This is obtained by enhancing the stochastic model with an external node observation model that samples a number of random *external nodes*, 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 θ_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.

Given the node independence assumptions at the basis of the naïve graph model, if we knew the correspondences σ_g mapping the nodes of graph g to the

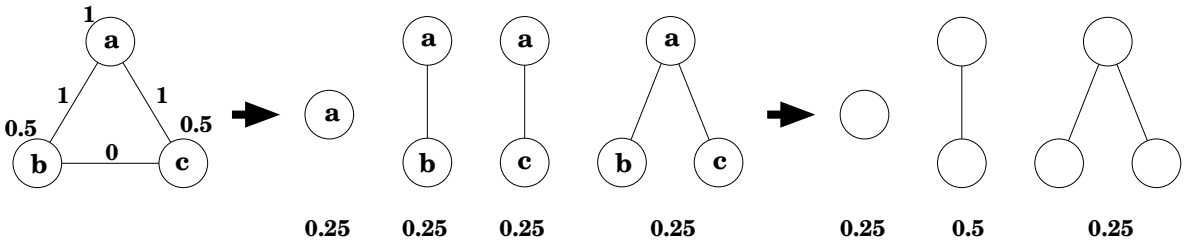

Fig. 1. A structural model and the generated graphs. When the correspondence information is lost, the second and third graph become indistinguishable.

nodes of the model $\mathcal{G}$, we could very easily compute the probability of observing graph g from model $\mathcal{G}$:

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

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, as shown in [15], 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.

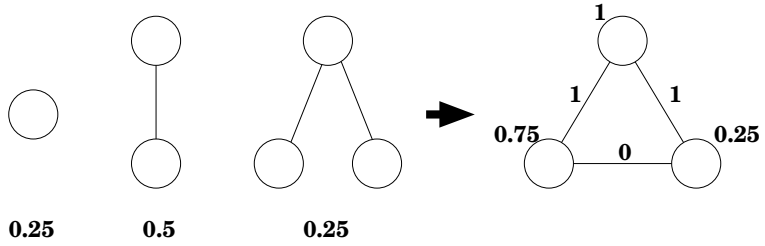

Fig. 2. Model estimation bias. If a single node correspondence is taken into account the estimated model will exhibit a bias towards one of multiple possible correspondences.

To solve this bias Torsello [15] proposed to marginalize the sampling probability over all possible correspondences, which, once extended to deal with external nodes, results in 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, Σ_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 Σ_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 Σ_n^m; hence, we have to resort to an estimation approach. In [15] 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 [11]. In particular, in the 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 [15] for models with external nodes, also eliminating the need to pad the observed graphs with dummy nodes to make them of the same size of the graph model.

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 deleted (not observed) and graph nodes can come from the external node model, 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 m 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 [13]

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 $\bar{M}_{1,\dots,n}^{h_1,\dots,h_n}$, corresponding to a correspondence σ, that has been sampled with probability $P(\sigma) = (\bar{M})_{1,h_1} \cdot (\bar{M}_1^{h_1})_{2,h_2} \cdot \ldots \cdot (\bar{M}_{1,\dots,n-1}^{h_1,\dots,h_{n-1}})_{n,h_n}$.

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 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 θ_i, ω_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_σ is the deterministic correspondence matrix associated with σ. 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 models, 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 [16] 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 learning process is initiated with a graph model that has several mixture components, each with more nodes that have been observed in any graph in the training set. We iteratively perform the EM learning procedure on the oversized model and, with the observation probabilities to hand, we decide whether to prune a node from a mixture component or a whole mixture component and after the model reduction we reiterate the EM parameter estimation and the pruning until no model simplification reduces the message length.

The pruning strategy adopted is a greedy one, selecting the operation that guarantees the largest reduction in message length given the current model parameters. Note that this greedy procedure does not guarantee optimality since the estimate is clearly a lower bound, as the optimum after the pruning can be in a very different point in the model-parameter space, but it does still give a good initialization for leaving the reduced parameter set.

In order to compute the reduction in message length incurred by removing a node, while sampling the correspondences we compute the matching probability not only of the current model, but also of the models obtained from the current one with any singe node removal. Note, however, that this does not increase the time complexity of the sampling approach and incurs only in a small penalty.

3 Experimental Evaluation

In order to asses the performance of the proposed approach, we run several experiments on graphs arising from different classification problems arising from 2D and 3D object recognition tasks, as well as one synthetic graph-classification testbed. The generative model is compared against standard nearest neighbor and nearest prototype classifiers based on the distances obtained using several graph matching techniques at the state of the art. In all cases the prototype is selected by taking the set-median of the training set. The performance of the

generative model is assessed in terms of the classification performance for the classification task to hand. For this reason, for all the experiments we plot the precision and recall values:

$$\text{precision} = \frac{tp}{tp + fp} \qquad \text{recall} = \frac{tp}{tp + fn}$$

where tp indicates the true positives, tn the true negatives and fn the false negatives.

With the exception to the last set of experiments, all the graphs used have a single numerical attribute associated to each node and no attributes linked with the edges. The last set of experiments, on the other hand, is based on edge-weighted graphs with no node attribute.

For the node-attributed graphs, we adopted the rectified Gaussian model used in [14]. To this end, we define a single stochastic node observation model X_i for each node i. We assume X_i is normally distributed with mean μ_i and standard deviation σ_i. When sampling node i from the graph model, a sample x_i is drawn from X_i. If $x_i \geq 0$ then the node is observed with weight $w_i = x_i$, otherwise the node will not be present in the sampled graph. Hence the node observation probability is $\theta_i = 1 - \text{erfc}(\mu_i/\sigma_i)$ where erfc is the complementary error function

$$\text{erfc} = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}s^2\right) ds\,.$$

The edge observation model, on the other hand is a simple Bernoulli process.

3.1 Shock Graphs

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 was to compare the classification results obtained learning a generative model to what can be obtained 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 and nearest prototype rules 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 that 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

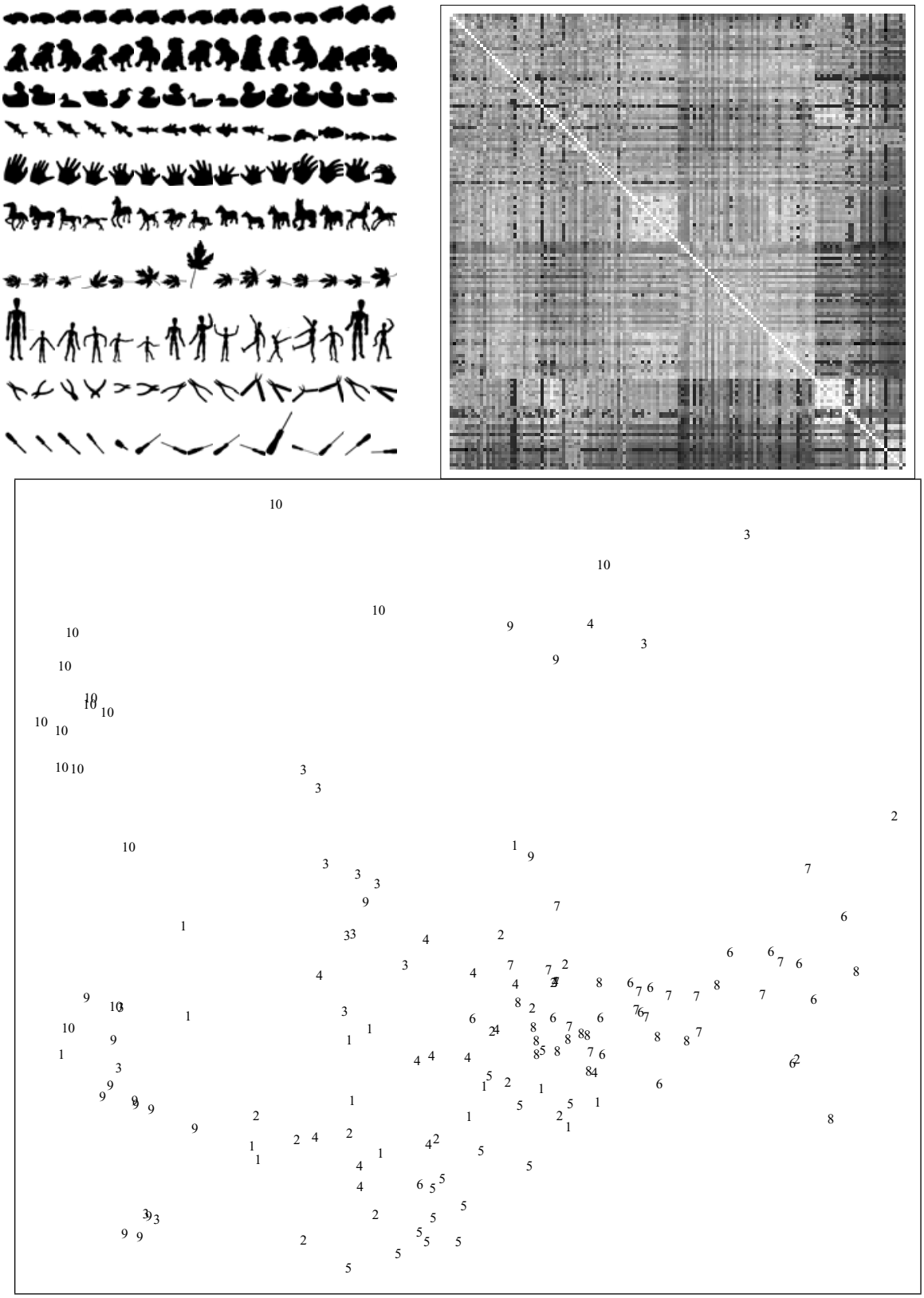

Fig. 3. Top row: Left, shape database; right, edit distance matrix. Bottom row: Multidimensional Scaling of the edit distances.

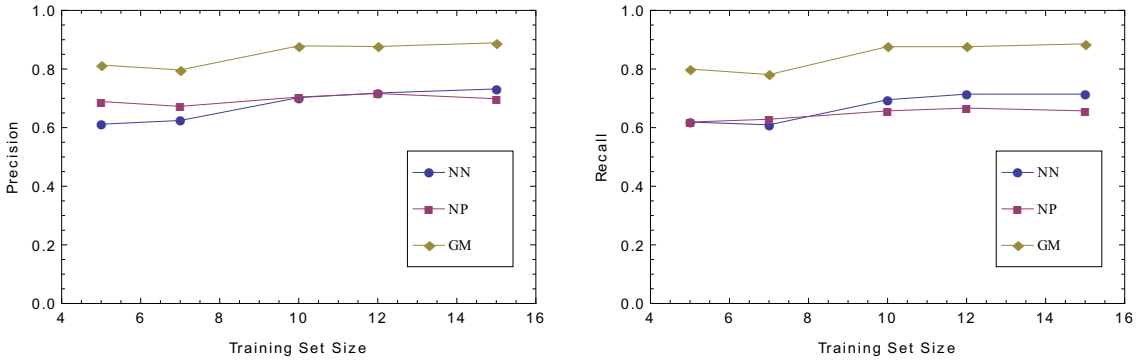

Fig. 4. Precision and Recall on the shock graph dataset as the number of training samples increases

improvement of about 15% on both precision and recall values 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.

3.2 3D Shapes

The second test set is based on a 3D shape recognition task. We collected a number of shapes from the McGill 3D Shape Benchmark [12] and we extracted their medial surfaces. The final dataset was obtained by transforming these skeletal representations into an attributed graph. Figure 5 shows the shapes, their graph distance matrix and a Multidimensional Scaling representaion of the distances. The distances between the graphs were computed using the normalized metric described in [17], which in turn relies on finding a maximal isomorphism between the graphs, for which we adopted the association graph-based approach presented in [10]. Both the distance matrix and the Multidimensional Scaling show that the classes are well separated, resulting in a relatively easy classification task.

Once again we tested the generative model performance against the nearest neighbor and the nearest prototype classifier. Figure 6 confirms our intuition that this was indeed an easy task, since both the nearest neighbor and the nearest prototype classifiers achieve the maximum performance. Yet, the generative model performs extremely well, even when the training set contains just a very few samples. As for the performance gap between the nearest neighbor and the generative model, it is probably due to the very naïve way of estimating the initial node correspondences, and could be probably reduced using a more sophisticated initialization.

3.3 Synthetic Data

To further assess the effectiveness of the proposed approach we tested it on synthetically generated data, where the data generation process is compatible

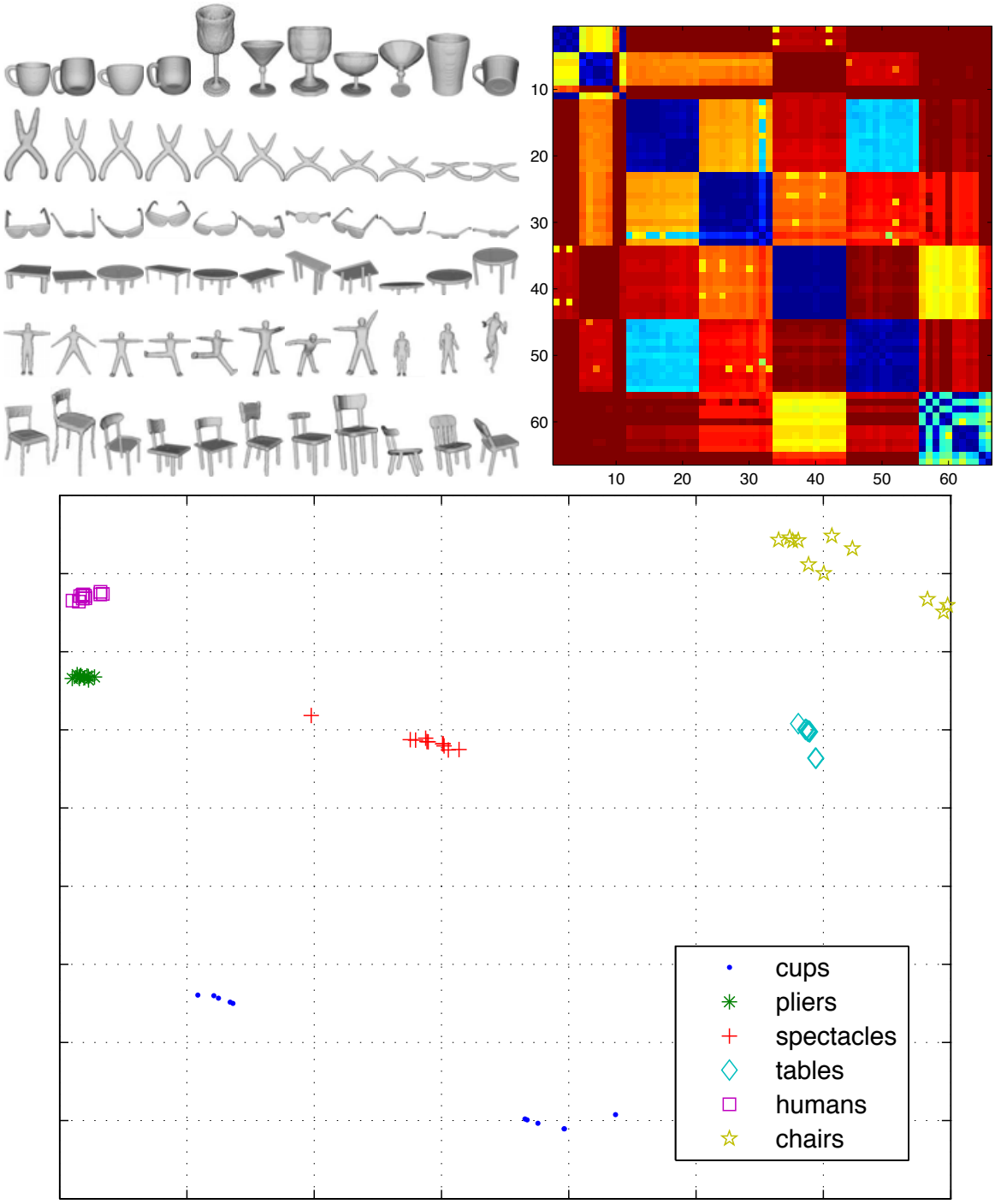

Fig. 5. Top row: Left, shape database; right, distance matrix. Bottom row: Multidimensional Scaling of the graph distances.

with the naïve model adopted in the proposed learning approach. To this end, we have randomly generated 6 different weighted graph prototypes, with size ranging from 3 to 8 nodes. For each prototype we started with an empty graph and then we iteratively added the required number of nodes each labeled with a random mean and variance. Then we added the edges and their associated observation probabilities up to a given edge density. Given the prototypes, we sampled 15 observations from each class being careful to discard graphs that

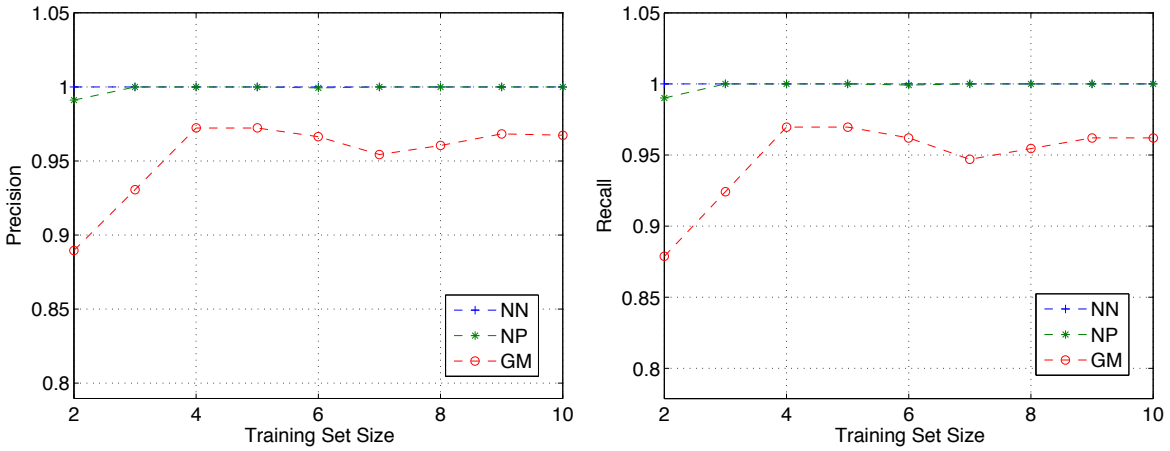

Fig. 6. Precision and Recall on the 3D shapes dataset

were disconnected. Then we proceeded as in the previous set of experiments computing the dissimilarities between the graphs and learning the graph models.

Generating the data with the same model used for learning might seem to give an unfair advantage to our generative model, but the goal of this set of experiments is asses the ability of the learning procedure to obtain a good model even in the presence of very large model-overlap. A positive result can also provide evidence for the validity of the optimization heuristics.

Figure 7 shows the distance matrix of the synthetic data and the corresponding Multidimensional Scaling representation. There is a considerable overlap between different classes, which renders the task particularly challenging for the nearest neighbor and nearest prototype classifiers. Yet, our generative model was able to learn and describe this large intra class variability, thus coping with the class overlap. Figure 8 plots the precision and recall curves for this set of experiments. Even with a relatively small training set, our approach achieves nearly 90% precision and recall, and as the number of observed samples increases, it yields perfect classification. On the other hand, the nearest neighbor classifier is not able to increase its precision and recall above the 84% limit, while the nearest prototype approach exhibits even lower performance.

3.4 Edge-Weighted Graphs

In the finals set of experiments, we applied the approach to an object recognition task. To this end we used a subset of the COIL-20 dataset [9]. For each image we extracted the most salient points using a Matlab implementation of the corner detector described in [7], the salient points where connected according to a Delaunay triangulation, thus resulting in an edge-weighted graph, were the edge-weights correspond to the distance between the salient points.

With this representation we used different node and edge observation models. Since nodes are not attributed, we used simple Bernoulli models for them. For the edges, on the other hand, we used a combined Bernoulli and Gaussian model: a Bernoulli process establishes whether the edge is observed, and if it is the weight is drawn according to an independent Gaussian variable. The reason for

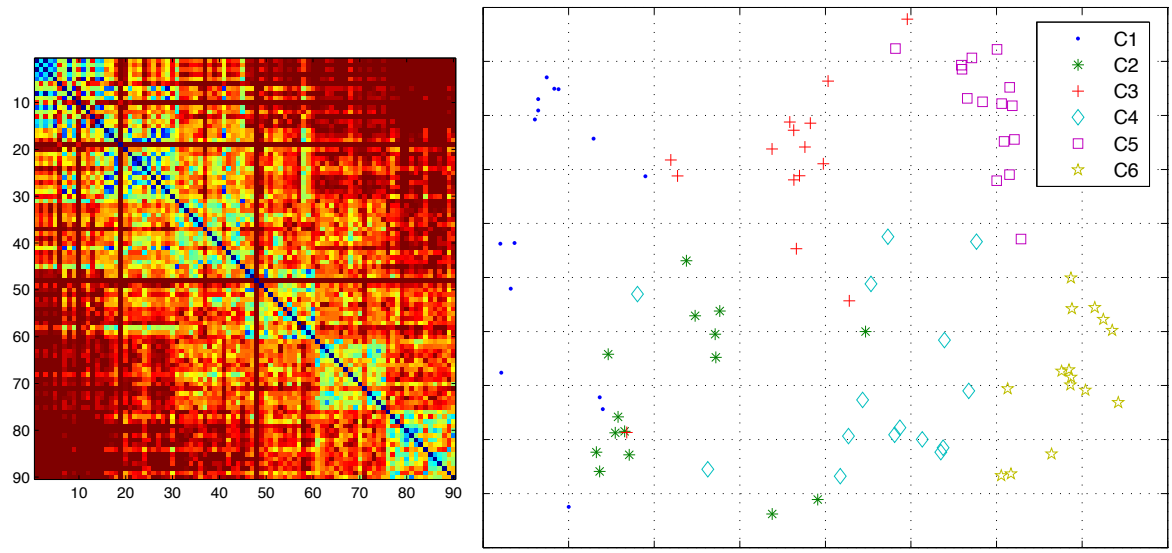

Fig. 7. Distance matrix and MDS of distances for the Synthetic Dataset

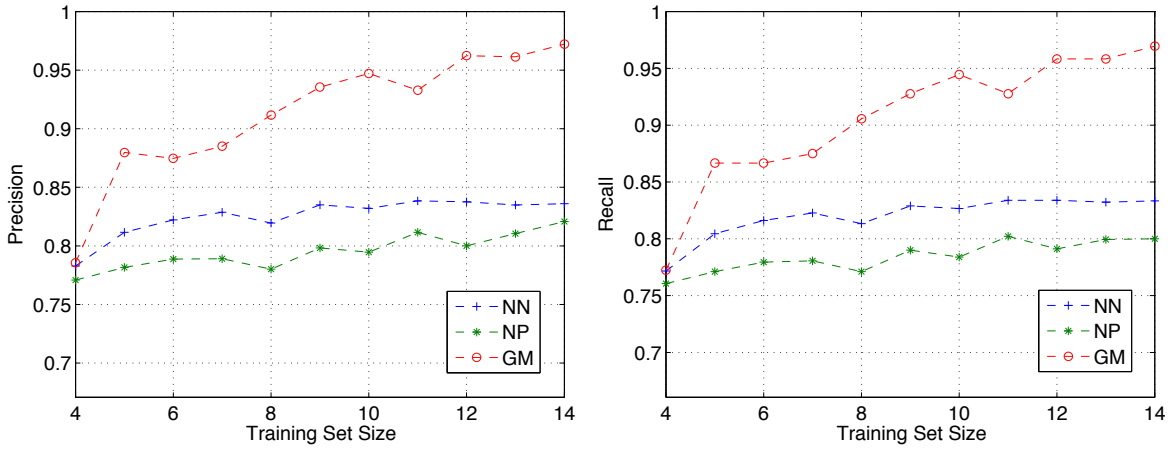

Fig. 8. Precision and Recall on the synthetic dataset

this different weight model resides in the fact that the correlation between the weight and the observation probability that characterized the rectified Gaussian model did not fit the characteristics of this representation.

To compute the distances for the nearest neighbor and nearest prototype rule, we used the graph matching algorithm described in [5], which is capable of dealing with edge-weighted graphs. Once the correspondences where computed, we adopted the same metric as before. As Figure 9 shows, the generated dataset is even more complex than the synthetic one. This is mainly due to the instability of the corner detector, which provided several spurious nodes resulting in very large intra-class structural variability.

Figure 10 shows that even on this difficult dataset, we significantly outperform both the nearest neighbor and nearest prototype classifiers, emphasizing once again the advantages of our structural learning approach.

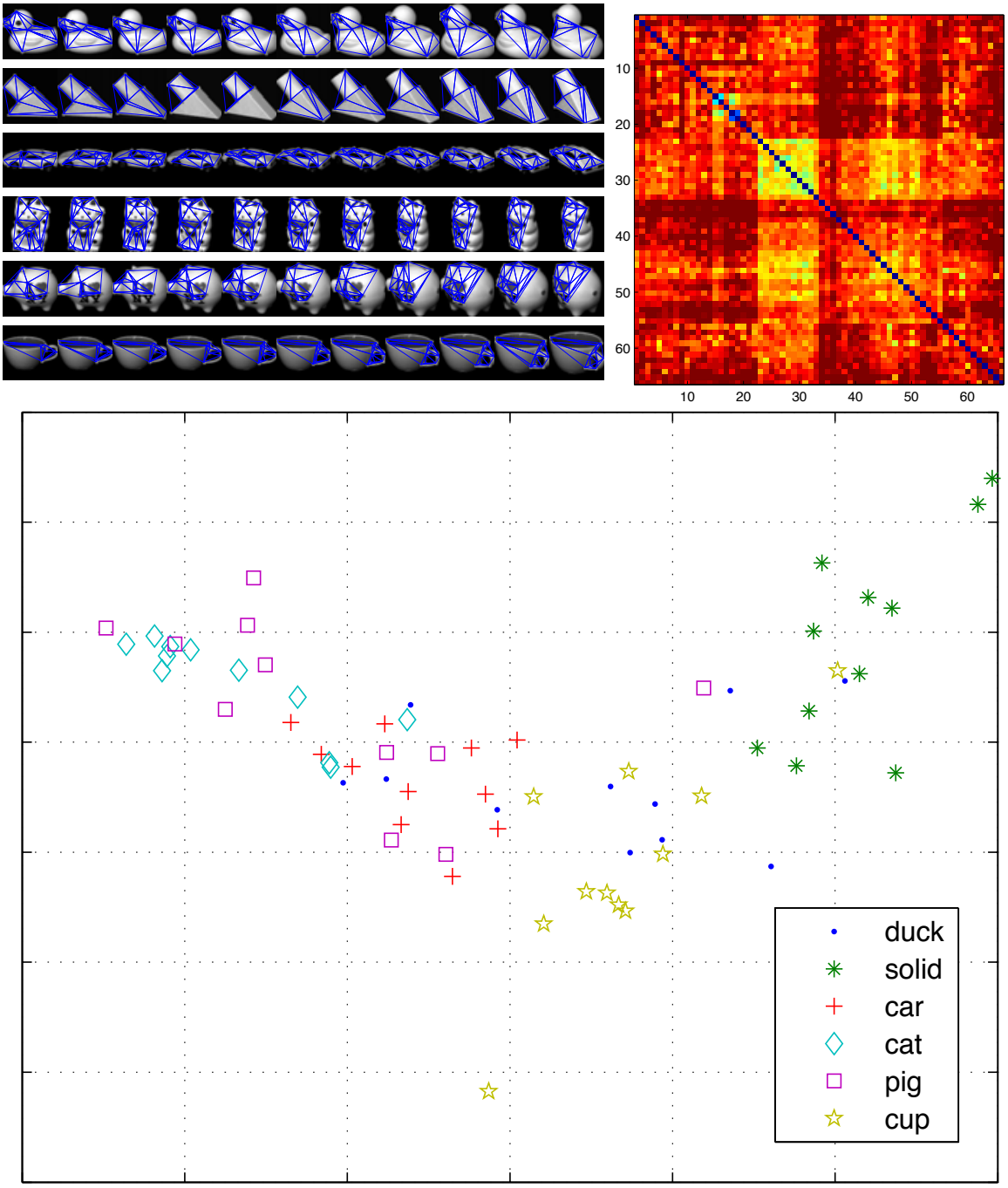

Fig. 9. Top row: Left, shape database; right, distance matrix. Bottom row: Multidimensional Scaling of the graph distances.

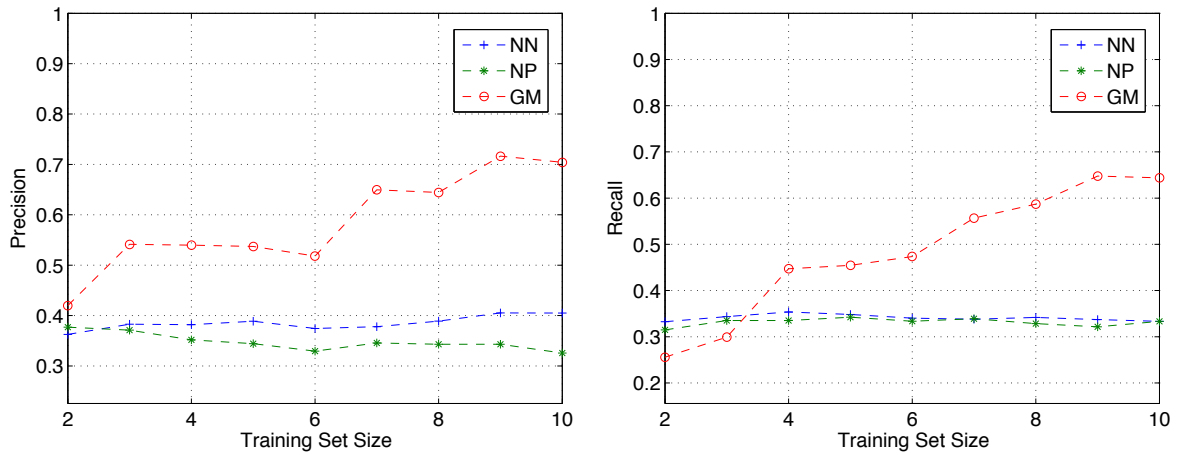

Fig. 10. Precision and Recall on the COIL-20 dataset

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 performed on a wide range of real world object recognition tasks as well as on synthetic data 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. In particular, the approach very clearly outperforms both the nearest neighbor and the nearest prototype rules regardless of the matching algorithm and the distance metric adopted.

References

1. Babai, L., Erdös, P., Selkow, S.M.: Random Graph Isomorphism. SIAM J. Comput. 9(3), 635–638 (1980)
2. Beichl, I., Sullivan, F.: Approximating the permanent via importance sampling with application to the dimer covering problem. J. Comput. Phys. 149(1), 128–147 (1999)
3. Bonev, B., et al.: Constellations and the Unsupervised Learning of Graphs. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 340–350. Springer, Heidelberg (2007)
4. Bunke, H., et al.: Graph Clustering Using the Weighted Minimum Common Supergraph. In: Hancock, E.R., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, pp. 235–246. Springer, Heidelberg (2003)
5. Cour, T., Srinivasan, P., Shi, J.: Balanced graph matching. In: Advances in NIPS (2006)

6. Friedman, N., Koller, D.: Being Bayesian about Network Structure. Machine Learning 50(1-2), 95–125 (2003)
7. He, X.C., Yung, N.H.C.: Curvature scale space corner detector with adaptive threshold and dynamic region of support. In: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR 2004), vol. 2, pp. 791–794. IEEE Computer Society, Washington, DC, USA (2004)
8. Luo, B., Hancock, E.R.: A spectral approach to learning structural variations in graphs. Pattern Recognition 39, 1188–1198 (2006)
9. Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library (COIL-20). Technical report (February 1996)
10. Pelillo, M.: Replicator equations, maximal cliques, and graph isomorphism. em Neural Computation 11(8), 1933–1955 (1999)
11. Rabbat, M.G., Figueiredo, M.A.T., Nowak, R.D.: Network Inference From Co-Occurrences. IEEE Trans. Information Theory 54(9), 4053–4068 (2008)
12. Siddiqi, K., et al.: Retrieving Articulated 3D Models Using Medial Surfaces. Machine Vision and Applications 19(4), 261–274 (2008)
13. Sinkhorn, R.: A relationship between arbitrary positive matrices and double stochastic matrices. Ann. Math. Stat. 35, 876–879 (1964)
14. Torsello, A., Hancock, E.R.: Learning Shape-Classes Using a Mixture of Tree-Unions. IEEE Trans. Pattern Anal. Machine Intell. 28(6), 954–967 (2006)
15. Torsello, A.: An Importance Sampling Approach to Learning Structural Representations of Shape. In: IEEE CVPR (2008)
16. Torsello, A., Dowe, D.: Learning a generative model for structural representations. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 573–583. Springer, Heidelberg (2008)
17. Torsello, A., Hidovic-Rowe, D., Pelillo, M.: Polynomial-time metrics for attributed trees. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1087–1099 (2005)
18. White, D., Wilson, R.C.: Spectral Generative Models for Graphs. In: Int. Conf. Image Analysis and Processing, pp. 35–42. IEEE Computer Society, Los Alamitos (2007)