Graph Clustering with Tree-Unions

Andrea Torsello and Edwin R. Hancock

Dept. of Computer Science, University of York, York, YO10 5DD, UK

Abstract. This paper focuses on how to perform unsupervised learning of tree structures in an information theoretic setting. The approach is a purely structural one and is designed to work with representations where the correspondences between nodes are not given, but must be inferred from the structure. This is in contrast with other structural learning algorithms where the node-correspondences are assumed to be known. The learning process fits a mixture of structural models to a set of samples using a minimum descriptor length formulation. The method extracts both a structural archetype that describes the observed structural variation, and the node-correspondences that map nodes from trees in the sample set to nodes in the structural model. We use the algorithm to classify a set of shapes based on their shock graphs.

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. Concrete examples include the use of shock graphs to represent shape-skeletons [11], the use of trees to represent articulated objects and the use of aspect graphs for 3D object representation. The attractive feature of structural representations is that they concisely capture the relational arrangement of object primitives, in a manner which can be invariant to changes in object viewpoint. However, despite the many advantages and attractive features of graph representations, the methodology available for learning structural representations from sets of training examples is relatively limited. As a result, the process of constructing shape-spaces which capture the modes of structural variation for sets of graphs has proved to be elusive. Hence, geometric representations of shape such as point distribution models [10, 4], have proved to be more amenable when variable sets of shapes must be analyzed.

Recently there has been considerable interest in learning structural representations from samples of training data, in particular in the context of Bayesian networks [3, 1], mixtures of tree-classifiers [8], or general relational models [2]. 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. Although these approaches provide a powerful way to infer the relations between the observable quantities of the model under examination, they rely on the availability of correspondence information for the nodes of the different structures used in learning. However, in many cases the identity of the nodes and their correspondences across samples of training data are not to hand, Instead, the correspondences must be recovered using a graph matching technique during the learning process. Hence, there is a chicken and egg problem in structural

N. Petkov and M.A. Westenberg (Eds.): CAIP 2003, LNCS 2756, pp. 451–459, 2003.

learning. Before the structural model can be learned, the correspondences with it must be available, and yet the model itself must be to hand to locate correspondences.

The aim in this paper is to develop a framework for the unsupervised learning of generative models of tree-structures from sets of examples. We pose the problem as that of learning a union structure from the set of examples with hidden or unknown correspondences. The structure is constructed through a set of edit operations. Associated with each node of the structure is a random variable which represents the probability of the node. There are hence three quantities that must be estimated. The first of these are the correspondences between the nodes in the training examples and the estimated union structure. Secondly, there is the union structure itself. Finally, there are the node probabilities.

We cast the estimation of these three quantities in an information theoretic setting. The problem is that of learning a mixture of trees to represent the classes of tree present in the training data. We use as our information criterion the description length for the union structure and its associated node probabilities given correspondences with the set of training examples [9]. An important contribution is to demonstrate that the description length is related to the edit distance between the union structure and the training examples. >From our analysis it follows that the edit costs are directly related to the entropy associated with the node probabilities. We perform three sets of updates. First, correspondences are located so as to minimize the edit distance. Secondly, the union structure is edited to minimize the description length. Thirdly, we make maximum likelihood estimates of the node probabilities. It is important to note that the union model underpinning our method assumes node independence on the training samples. Using a mixture of unions we condition this independence on the class. This conditional independence assumption, while often unrealistic, is at the basis of the naive Bayes model [6] which has proven to be robust and effective for a wide range of classification problems. We apply the resulting framework to the problem of learning a generative model for sets of shock trees.

2 Tree Edit-Distance

This section introduces the tree edit-distance framework, explains how it can be used to estimate node-correspondences, and gives an overview of the algorithm we use to approximate it.

The idea behind edit distance is that it is possible to identify a set of basic edit operations on nodes and edges of a structure, and to associate with these operations a cost. The edit-distance is found by searching for the sequence of edit operations that will make the two graphs isomorphic with one-another and which have minimum cost. The optimal sequence can be found using only structure reducing operations. This can be explained by the fact that we can transform node insertions in one tree into node removals in the other. This means that the edit distance between two trees is completely determined by the subset of residual nodes left after the optimal removal sequence, or, equivalently, by the nodes that are in correspondence. In particular the distance between two trees t and t' is:

$$D(t,t') = \sum_{i \notin \mathrm{Dom}(\mathcal{M})} r_i + \sum_{j \notin \mathrm{Im}(\mathcal{M})} r_j + \sum_{<i,j> \in \mathcal{M}} m_{ij}. \qquad (1)$$

Here r_i and r_j are the costs of removing i and j respectively, $\mathcal{M}$ is the set of pairs of nodes from t and t' that match, $m_{i,j}$ is the cost of matching i to j, and $\text{Dom}(\mathcal{M})$ and $\text{Im}(\mathcal{M})$ are the domain and image of the relation $\mathcal{M}$. Letting $\mathcal{N}^t$ be the set of nodes of tree t, the distance can be rewritten as:

$$D(t,t') = \sum_{i \in \mathcal{N}^t} r_i + \sum_{j \in \mathcal{N}^{t'}} r_j + \sum_{<i,j> \in \mathcal{M}} (m_{ij} - r_i - r_j).$$

Hence the distance is minimized by the set of correspondences that maximizes the utility

$$\mathcal{U}(\mathcal{M}) = \sum_{<i,j> \in \mathcal{M}} (r_i + r_j - m_{ij}). \tag{2}$$

Let O be the set of matches that satisfy the constraints residing on the tree, then the node correspondence that minimize the edit distance is

$$M^* = \underset{M \in O}{\text{argmax}} \, \mathcal{U}(M). \tag{3}$$

Let us assume that we know the utility of the best match rooted at every descendent of nodes i and j of t and t' respectively. We aim to find the set of siblings with greatest total utility. To do this we make use of a derived structure: the association graph. The nodes of this structure are pairs drawn from the Cartesian product of the descendents of i and j and each pair correspond to a particular association between a node in one tree to a node in the other. That is, for each pair of nodes a and b, children of i and j, we have an association node (a, b). We connect two such associations if and only if there is no inconsistency between the two associations, that is the corresponding subtree is obtainable. Furthermore we assign to the association (a, b) a weight equal to the utility of the best match rooted at a and b. The maximum weight clique of this graph is the set of consistent siblings with maximum total utility, hence the set of children of i and j that guarantee the optimal isomorphism. Given a method to obtain a maximum weight clique, we can use it to obtain the solution to our isomorphism problem. We refer to [13] for heuristics for the weighted clique problem.

3 Edit-Intersection and Edit-Union

As shown in the previous section, the edit distance between two trees is completely determined by the set of nodes that do not get removed by edit operations, that is, in a sense, the *intersection* of the sets of nodes. We would like to extend the approach to more than two trees so that we can represent the structural variations present in a set of examples trees T. To this end we assume that there is an underlying "structure model",which determines a distribution of tree structures, and that each tree is a sample drawn from that distribution. In this way edit operations are linked to sampling error, and their cost to the error probability. We, then, need a way to estimate the underlying structural model. Our model has three components: a set of nodes, a partial order relation between these nodes and a sampling probability for each node. Sampling from this distribution means sampling nodes according to their probability and extracting the minimal descriptions of the order relation restricted to the sampled nodes.

Restricting the analysis to the structural part of the model, our interpretation is equivalent to having a generating hierarchical structure, namely the tree-union, and obtaining the various tree samples by applying structure-reducing edit operations to it. The sampling process applies to this structure the edit operation E_i with probability $1-\theta^i$, where θ^i is the sampling probability of node i. Hence, given the structure of the tree-union, the set of correspondences $\mathcal{C} : (\bigcup_t \mathcal{N}^t) \rightarrow \mathcal{N}$ from the nodes of the tree samples to the nodes of the union, and the sampling probability of each node $\Theta : \mathcal{N} \rightarrow [0,1]$, we can express the probabilty of sampling a tree t as:

$$\Phi(t|\mathcal{C},\Theta) = \prod_{i\in\mathcal{N}} E_i(t|\mathcal{C},\theta^i), \tag{4}$$

where $E_i(t|\theta^i)$ is the sampling probability of node i and is defined as:

$$E_i(t|\theta^i) = \begin{cases} \theta^i & \text{if } \exists j \in \mathcal{N}^t, \mathcal{C}(j) = i \\ 1-\theta^i & \text{otherwise.} \end{cases} \tag{5}$$

That is $E_i(t|\theta^i)$ is θ^i if tree t samples node i, $1-\theta^i$ otherwise. The probability of a sample set $\mathcal{D}$ is, hence, $P(\mathcal{D}|\mathcal{C},\Theta) = \prod_{t\in\mathcal{D}} \Phi(t|\mathcal{C},\Theta)$.

4 Estimating the Structural Model

To estimate the structural part of the model we need to obtain the set of nodes of the model and correspondences from the nodes in the samples to the nodes of the model. With this correspondences, the nodes of the model span every node in the samples, and hence, the node set can be considered the "union" of the set of nodes of the samples. We refer to [14] for an analysis of the properties of the structure behind this "tree-union".

Formally, we would like to find the set of nodes $\mathcal{N}$, the sampling probability of each node $\Theta : \mathcal{N} \rightarrow [0,1]$, and the set of correspondences $\mathcal{C} : (\bigcup_t \mathcal{N}^t) \rightarrow \mathcal{N}$ from the nodes of the tree samples to the nodes of the union. To this purpose, given a sample set $\mathcal{D}$, we could use a maximum likelihood estimator

$$\mathcal{C}^* = \underset{\mathcal{C}}{\operatorname{argmax}} \left[P(\mathcal{D}|\mathcal{C},\Theta)\right].$$

In many real-world problems the underlying structural model might not be single: when dealing with shock graphs, for example, samples drawn from a single shape-class might be related to a single structural model, but it is reasonable to assume that the structures of the skeletons of shapes that are perceptually very different are not generated by a single model. For this reason, when fitting a generative model of tree distribution, we want to allow for the samples to be drawn from multiple tree-union models . Namely we would like to fit a mixture of tree-unions.

The mixture model is parametrized by the number of mixtures k, their sampling probability α_i, and the various union models U_i. The Union models are defined by their correspondences $\mathcal{C}_i$ and sampling probabilities Θ_i. That is the probabbility of a tree t is

$$P(t|\alpha,\mathcal{C},\mathbf{\Theta}) = \sum_{m=1}^{k} \alpha_m \Phi(t|\mathcal{C}_m,\Theta_m) \tag{6}$$

Here we use the Minimum Description Length (MDL) principle to describe the cost of the mixture model and the model representing it. Here the model is captured by the mixing proportions α_i, the union structures and the sampling probabilities θ_i^n for each union i and node n. To describe the data, we need, for each tree sample, to describe from which union model the sample was drawn. additionally, for each node in the union, we need to describe whether the node was present in the sample. Asymptotically the cost of describing the mixing components α_i and the component each one of n samples is drawn from is bo unded by $nI(\bar{\alpha})$, where $I(\bar{\alpha}) = -\sum_{m=1}^{k} \alpha_m \log(\alpha_m)$ is the entropy of the mixture distibution α. The cost of describing a the structure of a union mode can be considered proportional to the number of nodes, while the cost of describing the sampling probability θ_i^n of node n of union i and the existence of such node in each samples of $n\alpha_i$ samples generated by union i is asymptotically $n\alpha_i I(\theta_i^n)$. Here $I(\theta_i^n) = -\ theta_i^n \log(\theta_i^n) - (1-\theta_i^n)\log(1-\theta_i^n)$ is the entropy of the node sampling probability. Hence, given a model $\mathcal{H}$ with k unions, each with d_i nodes and probability α_i of being sampled, and node correspondences C, the descriptor length is:

$$\mathrm{LL}(\mathcal{H}) = nI(\boldsymbol{\alpha}) + \sum_{m=1}^{k}\sum_{j=1}^{dm}\left[n\alpha_m I(\theta_m^j) + c\right] . \tag{7}$$

In this equation, c is the length per node of the description of the structure of the edit union, in our experiments set to 1, while the sample probability θ_m^j is estimated from the correspondences as the fraction of trees generated by union m that sample node j.

5 Minimizing the Descriptor Length

Finding the global minimum of the descriptor length is an intractable combinatorial problem, so we have to resort to some local search technique. A common approach to minimizing the descriptor length of a mixture model is to use the Expectation-Maximization algorithm. Unfortunately, the complexity of the maximization step on our union-tree model grows dramatically with the number of trees in the union. This means that, when we relax the membership variables for the EM algorithm, each union will effectively include every sample-tree.

We have chosen a different approach that would allow us to limit the complexity of the maximization. The approach we have used is as follows.

- Start with an overly-specific model: a structural model per sample-tree, where each model is equiprobable and structurally identical to the respective sample-tree, and each node has sample probability 1.
- Iteratively generalize the model merging two tree-unions. The mixture components to be merged are chosen in such a way that their merger maximally decreases the descriptor length.
- The algorithm stops when there are no merges left that would decrease the descriptor length.

Both the EM algorithm and our approach are descent methods in the sense that each iteration strictly decreases the objective function. The main difference is in the direction of descent. The update direction of the EM algorithm is closer to the gradient, while our approach is, basically, a coordinate descent method: at each iteration we move only along one of the coordinates in the parameter space. The greatest advantage of coordinate descent methods is the extremely low space and time complexity of each it eration step. Furthermore, in our particular case, we are guaranteed convergence to a local minimum with at most a linear number of merges.

5.1 Merging Two Unions

The main requirement of our minimization algorithm is that we can optimally merge two union models. That is that we can find the optimal structure that generates every tree-sample previously assigned to the two models.

From equation 7 we see that the descriptor length is linear with respect to $\mathrm{LL}_i(\mathcal{H}_i)$, the descriptor length of union i. That is $\mathrm{LL}(\mathcal{H}) = nI(\alpha) + \sum_{m=1}^{k} \mathrm{LL}_m(\mathcal{H}_m)$, where $\mathrm{LL}_m(\mathcal{H}_m) = \sum_{j=1}^{dm} \left[na_m I(\theta_m^j) + c\right]$. Here na_m is simply the number of samples assigned to component m and the remaining part of the equation is linear in the nodes.

Given two tree unions U_1 and U_2, we need to construct a union $\hat{U}$ whose structure respects the the hierarchical constraints present in U_1 and U_2 and that minimizes $\mathrm{LL}_m(\mathcal{H}_m)$. Since U_1 and U_2 already assign node correspondences from the samples to the model, we can simply find the correspondences from the nodes in U_1 and U_2 to $\hat{U}$ and transitively extending the correspondences from the samples to the final model $\hat{U}$.

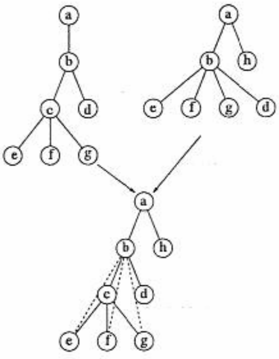

Fig. 1. The Union is defined by the common nodes.

Reduced to two structures, the correspondence problem is reduced to finding the set of nodes in U_1 and U_2 that are in common. Starting with the two structures, we merge the set of nodes that would reduce the descriptor length by the largest amount while still satisfying the hierarchical constraint. That is we merge nodes v and w of U_1 with node v' and w' of U_2 respectively if and only if $v \rightsquigarrow w \Leftrightarrow v' \rightsquigarrow w'$, where $a \rightsquigarrow b$ indicates that a is an ancestor of b. Assuming that the structures of U_1 and U_2 are trees, finding the set of nodes to be merged is equivalent to solving a tree-edit distance problem where the utility of a match is equivalent to the advantage in descriptor length we obtain through the merger. Let n_1 and n_2 be the number of samples in U_1 and U_2 respectively, and p_v and $p_{v'}$ the number of times nodes v and v' are sampled in U_1 and U_2 respectively, the sampling probability of the two nodes if they are not matched is $\theta v = \frac{p_v}{n_1+n_2}$ and $\theta v' = \frac{p'_v}{n_1+n_2}$ respectively, while the sampling probability of the node if the two are merged is $\theta v v' = \frac{p_v+p_{v'}}{n_1+n_2}$. Hence, the advantage in descriptor length we obtain through the merger is:

$$\mathcal{U}(v, v') = (n_1 + n_2)\left[I(\theta v) + I(\theta v') - I(\theta v v')\right] + c. \tag{8}$$

From an edit distance point of view this is equivalent to saying that the cost of removing node v is $r_v = (n_1 + n_2)I(\theta v) + c$, while the cost of matching v to v' is $m_{vv'} = (n_1 + n_2)I(\theta v v') + c$.

At the end of the node merging operation we are left with a set of nodes that respects the original partial order defined by the various hierarchies in the sample-trees. The links of our model will be obtained from the partial order by constructing the minimal representation. When this representation is a tree, every sample tree can be obtained from this structure with a sequence of node removal operations.

6 Experimental Results

We evaluate the approach on the problem of shock tree matching. The idea behind the shock formulation of shape is to evolve the boundary of an object to a canonical skeletal form using the eikonal equation. The skeleton represents the singularities (shocks) in the curve evolution, where inward moving boundaries collide. Once the skeleton is to hand, the next step is to devise ways of using it to characterize the shape of the original boundary. Here we follow Zucker, Siddiqi, and others, by labeling points on the skeleton using so-called shock-classes [11]. According to this taxonomy of local differential structure, there are different classes associated with behavior of the radius of the maximal circle bitangent to the boundary. The so-called shocks distinguish between the cases where the local osculating circle has maximum radius, minimum radius, constant radius or a radius which is strictly increasing or decreasing. We abstract the skeletons as trees in which the level in the tree is determined by their time of formation [11]. The later the time of formation, and hence their proximity to the center of the shape, the higher the shock in the hierarchy.

In order to asses the quality of the method we compare clusters defined by the components of the mixture with those obtained with those obtained using the graph clustering method described in [13, 7]. In our experiments we use only structural information to characterize the shapes, while [7] enhance the representation with geometrical information and [13] presents results both with purely structural and enhanced representations.

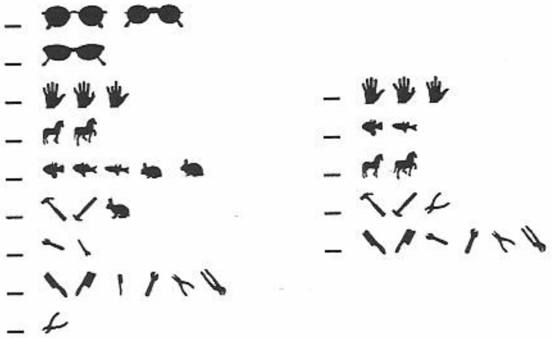

Fig. 2. Clusters extracted by the mixture of trees.

Figure 2 shows the clusters extracted on a database of 25 shapes and on a reduced database of 16 shapes. While there is some merger and leakage, the results outperform those obtained through pairwise clustering of the purely structural skeletal representations. Furthermore, it compares favorably with the pairwise clustering algorithm even where the latter is enhanced with geometrical information linked to the nodes of the trees.

6.1 Synthetic Data

To augment these real world experiments, we have fitted the model on synthetic data. The aim of the experiments is to characterize the sensitivity of the classification approach to class merger. To meet this goal we have randomly generated some prototype trees and, from each tree, we generated structurally perturbed copies. The trees are perturbed by randomly adding the required amount of nodes.

In our experiments we fit samples generated from an increasing number of prototypes and subject to an increasing amount of structural perturbation. We tested the classification performance on samples dawn from 2, 3, and 4 prototypes of 10 nodes each. The amount of noise is increased from an initial 10% of the total number of nodes to a maximum of 50%. Figure 3 plots the fraction of pairs of trees that are correctly classified as belonging to the same or different clusters as the noise is inc reased. From these experiment we can see that the approach works well with compact and well separated classes. The algorithm presents a sudden drop in performance when the structural variability of the class reaches 40% of the total number of nodes of the prototypes. Furthermore, when more prototypes are used, the distance between the clusters is smaller and, consequently the classes are harder to separate.

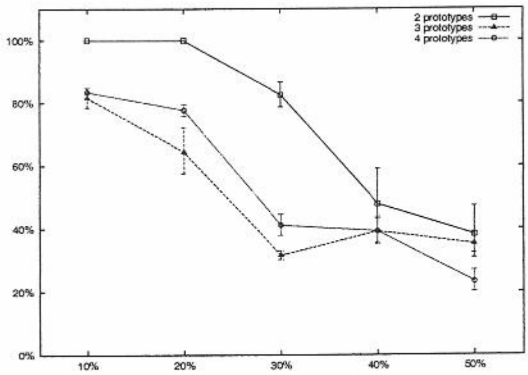

Fig. 3. Percentage of correct classifications under increasing structural noise.

7 Conclusions

This paper presented a novel algorithm to learn a generative model of tree structures. The approach uses the the Tree-Union as the structural archetype for every tree in the distribution and fits a mixture of these structural models using a minimal descriptor length formulation. In a set of experiments we apply the algorithm to the problem of unsupervised classification of shape using the shock-graphs. The results of these experiments are very encouraging, showing that the algorithm,although purely structural, compares favorably with pairwise classification approaches on attributed shock-graph. We are convinced that the results can be further improved by extending the model to take into account node-attributes.

References

1. N. Friedman and D. Koller, Being Bayesian about Network Structure, *Machine Learning*, to appear, 2002
2. L. Getoor et al., Learning Probabilistic models of relational structure, in *8th Int. Conf. on Machine Learning*, 2001.
3. D. Heckerman, D. Geiger, and D. M. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, *Machine Learning*, Vol. 20(3), pp. 197-243, 1995.
4. T. Heap and D. Hogg, Wormholes in shape space: tracking through discontinuous changes in shape, in *ICCV*, pp. 344-349, 1998.

5. X. Jiang, A. Muenger, and H. Bunke, Computing the generalized mean of a set of graphs, in *Workshop on Graph-based Representations, GbR'99*, pp 115-124, 2000.
6. P. Langley, W. Iba, and K. Thompson, An analysis of Bayesian classifiers, in *AAAI*, pp. 223-228, 1992
7. B. Luo, et al., Clustering shock trees, in *CVPR*, pp. 912-919, 2001.
8. M. Meilă. *Learning with Mixtures of Trees*. PhD thesis, MIT, 1999.
9. J. Riassen, Stochastic complexity and modeling, *Annals of Statistics*, Vol. 14, pp. 1080-1100, 1986.
10. S. Sclaroff and A. P. Pentland, Modal matching for correspondence and recognition, *PAMI*, Vol. 17, pp. 545-661, 1995.
11. K. Siddiqi et al., Shock graphs and shape matching, *Int. J. of Comp. Vision*, Vol. 35, 1999.
12. T. Sebastian, P. Klein, and B. Kimia, Recognition of shapes by editing shock graphs, in *ICCV*, Vol. I, pp. 755-762, 2001.
13. A. Torsello and E. R. Hancock, Efficiently computing weighted tree edit distance using relaxation labeling, in *EMMCVPR*, LNCS 2134, pp. 438-453, 2001.
14. A. Torsello and E. R. Hancock, Matching and embedding through edit-union of trees, in *ECCV*, LNCS 2352, pp. 822-836, 2002.