Learning Mixtures of Tree-Unions by Minimizing Description Length

Andrea Torsello and Edwin R Hancock

Dept. of Computer Science, University of York
Heslington, York, YO10 5DD, UK
atorsell@cs.york.ac.uk

Abstract. This paper focuses on how to perform the 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 description length formulation. The method extracts both a structural archetype that desribes 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 [1] have been used with considerable success in computer vision in the abstraction and recognition of object shape and scene structure. Concrete examples include the use of shock graphs to represent shape-skeletons [11, 19], the use of trees to represent articulated objects [13, 8, 24] and the use of aspect graphs for 3D object representation [2]. 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 [18, 7], 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 [6, 4], mixtures of tree-classifiers [15], 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 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 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 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 [16]. 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 [12] 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. The shock tree is an abstraction of 2D shape which is obtained by assigning labels to the branches of the Blum skeleton for the object boundary. [20]. The shock labels are related to the differential structure of the object boundary. They distinguish whether the radius of the bitangent circle to the object boundary is increasing, decreasing, constant, locally maximum or locally minimum. The shock-graph is hence a tree-like characterization of the differential structure of the boundaries of 2D shapes, where nodes represent sections of the morphological skeleton of the shape and edges represent their

adjacency relations. Changes in shape give rise to structural variations in the shock tree. By fitting our mixture of tree-unions to sets of shock trees we are able to construct a shape-space for the set of examples. We both learn shape classes present in the training data, and construct a shape space for each class. To construct the shape-space, we develop our previously published work where the node frequencies are used as the components of a pattern-vector [23]. Here, we construct a generative model and the node probabilities for each union-structure are used as the components of the pattern vectors in shape-space. Moreover, we can potentially sample example trees from the generative model learned in this way.

Hence we make a number of contributions. There have been a previous attempts to learn trees and mixtures of trees, and to apply these methods to vision For instance Meilla [15] has developed a probabilistic framework for learning mixtures of trees. Our work develops these theoretical ideas by establishing the link between description length and tree edit distance. From an applications perspective, there have been several attempts to use tree representations in vision. As a concrete example, Liu and Gieger [13] have used free trees to represent articulated objects. The FORMS system of Zhu and Yuille [24] also uses tree representations. Ioffe and Forsyth [8] have used related ideas of model walking people using mixtures of trees. Our work provides a means of learning tree representations that can be used to construct shape-spaces for such applications.

Finally, from the perspective of shock-tree analysis we also provide a number of concrete contributions. Any attempt to learn the modes of structural variation linked to a shape has to deal with the lack of prior knowledge about the correspondences between skeletal components belonging to different sample training shapes. Graph-matching allows the explicit pairwise comparison of graphical representations. For example [21, 22, 14] use edit distance to extract the node correspondences and provide a measure of dissimilarity between structures. Furthermore, in [14] we use a pairwise clustering algorithm to classify the shapes based on the edit distance between their shock-graphs. These approaches, while effective, give no insight into the generative model which gives rise to the observed distribution of shock-trees for a particular shape. Furthermore, the notion of distance that pairwise comparison approaches rely on is purely geometrical and it does not differentiate between shape elements that present a great variation among the training samples and elements that are virtually invariant. Recently there have been some attempts to extend the graph matching approach to take into account a set of sample training structures. In [9] the authors use the mean graph as a representative of the training samples, while [23] introduces the tree-union as a model of the structural variability present in a set of trees. An important advantage that the union approach has over the mean graph is that it represents explicitly how the training samples vary as well as what their common features. The generative tree model that we are proposing is obtained using the tree-union as the structural archetype for every tree in the distribution. Following this approach, we pose the shape classification problem as one of unsupervised learning of a mixture model, where each element of the

mixture is a tree-union which represents the intrinsic structural variations in a shape class.

The outline of this paper is as follows. In Section 2 we describe the generative tree model that underpins our graph-clustering method. This focuses on details of the tree-union, and structure in terms of order-relations, and the maximum likelihood framework for node probability estimation. Section 3 extends the framework to mixtures of tree-unions. Here we show how the problem of selecting the mixture of trees may be posed as a process of minimizing a description length criterion. Section 4 turns to details of how the description length criterion may be minimized. This is realized by commencing with an over-specific model in which there is a mixture component per data sample. We then merge pairs of trees so as to maximize the gain in description length advantage. In Section 5, we explore the relationship between the change in description length gained through tree merge operations and the corresponding tree edit distance. Here we show that the edit costs are related to the node entropies (and hence the node probabilities). This demonstrates that we effectively have a means by which tree edit costs may be learned. In Section 6 we provide experiments which demonstrates the utility of our method for the problem of clustering shock trees. Finally, Section 7 offers some conclusions and directions for future work.

2 Generative Tree Model

Consider the set or sample of trees $\mathcal{D} = \{t_1, t_2, \ldots, t_n\}$. Our aim in this paper is to cluster these trees, i.e. to perform unsupervised learning of the class structure of the sample. We pose this problem as that of learning a generative model for the distribution of trees in a pattern space. The distribution of trees in the pattern space is modeled using a mixture model. Each class or cluster of trees is represented by a separate generative model. In other words, the components of the mixture model must be capable of capturing the structural variations for the sample trees which belong to a separate class using a probability distribution.

The set of tree-models constituting the mixture model is denoted by $\mathcal{H} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$. Each tree model $\mathcal{T}$ is a structural archetype derived from the tree-union over the set of trees constituting a class. Associated with the archetype is a probability distribution which captures the variations in tree structure within the class. Hence, the learning process involves estimating the union structure and the parameters of the associated probability distribution for the class model $\mathcal{T}$. As a prerequisite, we require the set of node correspondences $\mathcal{C}$ between sample trees and the union tree for each class.

The learning process is cast into an information theoretic setting and the estimation of the required class models is effected using optimization methods. The quantity to be optimized is the description length for the sample-data set $\mathcal{D}$. The parameters to be optimized include the structural archetype of the model $\mathcal{T}$ as well as the node correspondences $\mathcal{C}$ between samples and the archetype. Hence, the inter-sample node correspondences are not assumed to be known a priori. Since the correspondences are uncertain, we must solve two interdependent opti-

mization problems. These are the optimization of the union structure given a set of correspondences, and the optimization of the correspondences given the tree structure. These dual optimization steps are approximated by greedily merging similar tree-models.

The basic ingredients of our structural learning approach are:

1. A structural model of tree variation.
2. A probability distribution on the said model.
3. A structural optimization algorithm that allows us to merge two structural models in a way that minimizes the description length.

In prior work, we have described how tree unions can be used as structural models for samples of trees [23]. However, the union is constructed so as to minimize tree-edit distance. Here we intend to use the union structure as a class model. However, we extend the idea in two important ways. First, we pose the recovery of the union tree in an information theoretic setting. Second, we aim to characterize uncertainties in the structure by assigning probabilities to nodes. Hence, the structural model is provided by the tree-union of the set of samples assigned to a mixture component, while the frequencies with which nodes from the sample set are mapped to nodes in the model provide the probability distribution. By adopting this information theoretic approach we demonstrate that the tree-edit distance, and hence the costs for the edit operations used to merge trees, are related to the entropies associated with the node probabilities. As a result, we provide a framework in which tree edit distances are learned. This has been a longstanding problem since Fu and his co-workers introduced the idea of edit distance in the early 1980's [17, 3].

The basis of the proposed structural learning approach is a generative model of trees which allows us to assign a probability distribution to a sample of hierarchical trees. A hierarchical tree t is defined by a set of nodes $\mathcal{N}^t$ and a tree-order relation $\mathcal{O}^t \subset \mathcal{N}^t \times \mathcal{N}^t$ between the nodes. A tree-order relation $\mathcal{O}^t$ is an order relation with the added constraint that if $(x, y) \in \mathcal{O}^t$ and $(z, y) \in \mathcal{O}^t$, then either $(x, z) \in \mathcal{O}^t$ or $(z, x) \in \mathcal{O}^t$. A node b is said to be a *descendent* of a, or $a \rightsquigarrow b$, if $(a, b) \in \mathcal{O}^t$, furthermore, b descendent of a is also a *child* of a if there is no node x such that $a \rightsquigarrow x$ and $x \rightsquigarrow b$, that is there is non node between a and b in the tree-order.

Given this definition, we can construct a generative model for a class of trees $\mathcal{D}_c \subset \mathcal{D}$. This model $\mathcal{T} = \mathcal{T} = (\mathcal{N}, \mathcal{O}, \Theta)$ is an instance of a set of nodes $\mathcal{N}$. Associated with the set of nodes is a tree order relation $\mathcal{O} \subset \mathcal{N} \times \mathcal{N}$ and a set $\Theta = \{\theta^i, i \in \mathcal{N}\}$ of sampling probabilities θ^i for each node $i \in \mathcal{N}$.

A sample from this model is a hierarchical tree $t = (\mathcal{N}^t, \mathcal{O}^t)$ with node set $\mathcal{N}^t \subset \mathcal{N}$ and a node hierarchy $\mathcal{O}^t$ that is the restriction to $\mathcal{N}^t$ of $\mathcal{O}$.

The probability of observing the sample tree t given the model tree $\mathcal{T}$ is $P\{t|\mathcal{T}\} = \prod_{i \in \mathcal{N}^t} \theta^i \prod_{j \in (\mathcal{N} \setminus \mathcal{N}^t)} (1 - \theta^j)$. The model underpinning this probability distribution is as follows. First, we assume that the set of nodes $\mathcal{N}$ for the union structure $\mathcal{T}$ spans all the nodes that might be encountered in the set of sample trees. Second, we assume that the sampling error acts only on nodes, while the hierarchical relations are always sampled correctly. That is, if nodes i and j

satisfy the relation $i\mathcal{O}j$, node i will be an anchestor of node j in each tree-sample that has both nodes. This assumption implies that two nodes will always satisfy the same hierarchical relation whenever they are both present in a sample tree. A consequences of this assumptions is that the structure of a sample tree is completely determined by restricting the order relation of the model $\mathcal{O}$ to the nodes observed in the sample tree. Hence, the links in the sampled tree can be seen as the minimal representation of the order relation between the nodes. The sampling process is equivalent to the application of a set of node removal operations to the archetypical structure $\mathcal{T} = (\mathcal{N}, \mathcal{O}, \Theta)$, which makes the archetype a union of the set of all possible tree samples.

The definition of the structural distribution assumes that we know the correspondences between the nodes in the sample tree t and the nodes in the class-model $\mathcal{T}$. When obtaining a sample from the generative model this assumption obviously holds. However, given a tree t, the probability that this tree is a sample from the class model $\mathcal{T}$ depends on the tree, the model, but also on the way we map the nodes of the tree to the corresponding nodes of the model. To capture this correspondence problem, we define a map $\mathcal{C} : \mathcal{N}^t \rightarrow \mathcal{N}$ from the set $\mathcal{N}^t$ of the nodes of t, to the nodes of the model.

The mapping induces a sample-correspondence for each node $i \in \mathcal{N}$. The correspondence probability for the node i is

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

while the probability of sampling the tree t from the model $\mathcal{T}$ given the set of correspondences $\mathcal{C}$ is

$$\Phi(t|\mathcal{T}, \mathcal{C}) = \begin{cases} \prod_{i \in \mathcal{N}} \phi(i|t, \mathcal{T}, \mathcal{C}) & \text{if } \forall v, w \in \mathcal{N}^t, v \rightsquigarrow w \iff \mathcal{C}(v) \rightsquigarrow \mathcal{C}(w) \\ 0 & \text{otherwise.} \end{cases}$$

Given a set $\mathcal{D} = \{t_1, t_2, \ldots, t_n\}$ of sample trees, we would like to estimate the tree model $\mathcal{T}$ that generated the samples, and the mapping $\mathcal{M}$ from the nodes of the sample trees to the nodes of the tree model. Here we use a maximum likelihood method to estimate the parameters. The log-likelihood of the sample data $\mathcal{D}$ given the tree-union model $\mathcal{T}$ and the correspondence mapping function $\mathcal{C}$ is $\mathcal{L}(\mathcal{D}|\mathcal{T}, \mathcal{C}) = \sum_{t \in \mathcal{D}} \log \left[\Phi(t|\mathcal{T}, \mathcal{C})\right]$. Our aims is to optimize the log-likelihood with respect to two variables: the correspondence map $\mathcal{C}$ and the tree union model $\mathcal{T}$. These variables, though, are not independent. The reason for this is that they both depend on the node-set $\mathcal{N}$. However, the dependency to the node-set can be lifted. The value of the log-likelihood function does not depend on the actual number of nodes because nodes with no associated samples will have correspondence probability $\phi(i|t, \mathcal{T}, \mathcal{C}) = 1$. Hence, the dependency to the node-set can be lifted by simply assuming that the node set is $Im(\mathcal{C})$, the image of the correspondence map. With this simplification, the remaining variables are: the order relation $\mathcal{O}$, the set of sampling probabilities Θ, and the map $\mathcal{C}$.

Given $\mathcal{C}$, it is easy to maximize with respect to the remaining two sets of variables. log-likelihood function is maximized by any order relation $\mathcal{O}$ that is

consistent with the hierarchies for the sample trees (if any exists). Let $n_i(\mathcal{C})$ be the number of trees $t \in \mathcal{D}$ such that $\exists j|\mathcal{C}(j) = i$, that is there is a node that maps to i. Furthermore, let $m = \#\mathcal{D}$ be the number of trees in the data set, then the sampling probability θ^i for the node i that maximizes the likelihood function is $\hat{\theta}^i = \frac{n_i(\mathcal{C})}{m}$. When the optimal sampling probabilities are substituted into the log-likelihood, we have that

$$\hat{\mathcal{L}}(\mathcal{D}|\mathcal{C}) = \sum_{i \in \mathcal{N}} m \left[\frac{n_i(\mathcal{C})}{m} \log \left(\frac{n_i(\mathcal{C})}{m} \right) + \left(1 - \frac{n_i(\mathcal{C})}{m} \right) \log \left(1 - \frac{n_i(\mathcal{C})}{m} \right) \right] = - \sum_{i \in \mathcal{N}} m I(\hat{\theta}^i), \quad (1)$$

where $I(\hat{\theta}^i) = -\left[\hat{\theta}^i \log(\hat{\theta}^i) + (1 - \hat{\theta}^i) \log(1 - \hat{\theta}^i) \right]$ is the entropy of the sampling distribution for node i. This equation holds assuming that there exists an order relation that is respected by every hierarchical tree in the sample set $\mathcal{D}$. If this is not the case then the log-likelihood function takes on the value $-\infty$.

The structural component of the model is a tree union constructed from the trees in the sample $\mathcal{D}$ so as to maximize the likelihood function.In our previous work [23]. we have shown how the union tree may be constructed so that every tree in the sample set $\mathcal{D}$ may be obtained from it by using node removal operations alone. Hence every node in the tree sample is represented in the union structure. Moreover, the order-relations in the union structure are all preserved by pairs of nodes in the tree-samples in $\mathcal{D}$.

3 Mixture Model

A single tree-union may be used to represent a distribution of trees that belong to a single class $\mathcal{D}_c$. Defining characteristic of the class is tha fact that the nodes present in the sample trees satisfy a single order relation $\mathcal{O}_c$. However, the sample set $\mathcal{D}$ may have a more complex class structure and it may be necessary to describe it using multiple tree unions. Under these conditions the unsupervised learning process must allow for the multiple classes, and we represent the distribution sample trees using a mixture model over separate union structures. Let the set of union structures be denoted by $\mathcal{H} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_c, \ldots, \mathcal{T}_k\}$, and let the corresponding mixing proportions be represented by the vector $\bar{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_c, \ldots, \alpha_k)$. The mixture model for the distribution of sample trees is

$$P(t|\mathcal{H}, \mathcal{C}) = \sum_{c=1}^{k} \alpha_c \Phi(t|\mathcal{T}_c, \mathcal{C}).$$

where z_c^t is an indicator variable, that is 1 if tree t belongs to the mixture component c, and 0 otherwise. The log-likelihood function for the mixture model over the sample-set $\mathcal{D}$ is:

$$\mathcal{L}(\mathcal{D}|\mathcal{H}, \mathcal{C}, \bar{z}) = \sum_{t \in \mathcal{D}} \sum_{c=1}^{k} \left[\ln \alpha_c + z_c^t \ln \Phi(t|\mathcal{T}_c, \mathcal{C}) \right],$$

It is well known that the maximum likelihood criterion cannot be directly used to estimate the number of mixture components, since the maximum of the likelihood function is a monotonic function on the number of components. In order to overcome this problem we use use the Minimum Description Length (MDL) principle. The MDL principle [16] asserts that the model that best describes a set of data is that which minimizes the combined cost of encoding the model and the error between the model and the data.

Our model is prescribed by the the vector of mixing proportions $\bar{\alpha}$ and the set of union structures $\mathcal{H} = \{\mathcal{T}_1, \ldots, \mathcal{T}_c, \ldots, \mathcal{T}_k\}$. The union structure $\mathcal{T}_c = \{\mathcal{N}_c, \mathcal{O}_c, \Theta_c\}$ for the mixture component indexed c consists of a set of nodes $\mathcal{N}_c$, a set of order relations $\mathcal{O}_c$ and a set of node probabilities $\Theta_c = \{\theta_c^i, i \in \mathcal{N}_c\}$, where θ_c^i is the probability for the node i in the union-tree indexed c. To describe or encode the fit of the model to the data, for each tree sample t we use the indicator variables $\bar{z}_c^t$ which indicates from which tree model the sample was drawn. Additionally, for each node in the model, we need to describe or encode whether or not the node was present in the sample.

By virtue of Shannon theorem, the cost incurred describing or encoding the model $\mathcal{H}$ is $-\log[P(\mathcal{H})]$, while the cost of describing data $\mathcal{D}$ using that model is $-\log[P(\mathcal{D}|\mathcal{H})]$. Making the dependence on the correspondences $\mathcal{C}$ explicit, we have: $LL(\mathcal{D}|\mathcal{H}) = -\mathcal{L}(\mathcal{D}|\mathcal{H}, \mathcal{C})$. Asymptotically the cost of describing the vector of mixing components $\bar{\alpha}$ and the set of indicator variables $\bar{z} = \{z_c^T, t \in \mathcal{D}, c = 1, ..., k\}$ is bounded by $nI(\bar{\alpha})$, where n is the number of samples in $\mathcal{D}$ and $I(\bar{\alpha}) = -\sum_{c=1}^{k} \alpha_c \log(\alpha_c)$ is the entropy of the mixture distribution $\bar{\alpha}$. The cost of describing the structure of a union model is proportional to the number of nodes contained within it, while the cost of describing the sampling probability θ_c^i of node i for model c and the existence of this node in each of the $n\alpha_c$ samples generated by union c is asymptotically equal to $n\alpha_c I(\theta_c^i)$. Here $I(\theta_c^i) = -\theta_c^i \log(\theta_c^i) - (1-\theta_c^i)\log(1-\theta_c^i)$ is the entropy associated with the node sampling probability. Hence, given a model $\mathcal{H}$ consisting of k tree-unions, where the component $\mathcal{T}_c$ has d_c nodes and a mixing proportion α_c, the description length for the model, conditional on the set of correspondences is $\mathcal{C}$ is:

$$\mathrm{LL}(\mathcal{D}|\mathcal{H}, \mathcal{C}) = nI(\bar{\alpha}) + \sum_{c=1}^{k} \sum_{i=1}^{d_c} \left[n\alpha_c I(\theta_c^i) + l \right]. \tag{2}$$

where l is the description length per node of the tree-union structure, which we set to 1. Given that $p_c^i = \{j \in \mathcal{N}^t | t \in \mathcal{D}, \mathcal{C}(j) = i\}$ is the set of nodes from sample trees in $\mathcal{D}$ mapped by $\mathcal{C}$ to node i of model c, the node probability θ_c^i is estimated using $\hat{\theta}_c^i = \frac{\#p_c^i}{d_c}$.

4 Learning the Mixture

Finding the global minimum of the description length is an intractable combinatorial problem. Hence, we resort to a local search technique. A widely used method for minimizing the description length of a mixture model is to use the

Expectation-Maximization algorithm. Unfortunately, the complexity of the maximization step for our union-tree model grows dramatically with the number of trees in the union. The problem arises from the fact that the membership indicators admit the possibility that each union can potentially include every sample-tree.

We have adopted a different approach which allows us to limit the complexity of the maximization step. The approach we have used is as follows.

- Commence with an overly-specific model. We use 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 by merging pairs of tree-unions. The candidates for merging are chosen so that they maximally decrease the description length.
- The algorithm stops when there are no merges remaining that can decrease the description length.

This algorithm bears some resemblance with the spanning tree clustering algorithm [10]. Both algorithms iteratively merge samples or clusters that satisfy a minimum distance or maximum similarity criterion. The main difference is that, in our algorithm, the similarity matrix is cannot be assumed fixed as is the case with the spanning tree algorithm. Rather, it changes after each merge to reflect the changes in the joint model. This change in the distance matrix will limit the amount of chaining allowed in the clusters. This is due to the fact that the models describing the two clusters that are merged are substituted by a single model. This new model must be able to describe the variation present in both clusters, hence, its mean must be placed in model-space somewere between the means of two models. This implies that the distance to the remaining cluster must vary. Regardless of these differences, the algorithm is still guaranteed to converge to a local minimum with at most a linear number of merges.

The main requirement of our description length minimization algorithm is that we can optimally merge two tree models. That is that we can find a structure from which it is possible to sample every tree previously assigned to the two models. From equation 2 we see that the description length is linear with respect to the contribution from each component of the mixture. In fact, writing the description cost of component c as $LL_c(\mathcal{D}|\mathcal{T}_c,\mathcal{C}) = \sum_{j=1}^{dc}\left[na_c I(\theta_c^j)+l\right]$, where na_c is the number of data samples assigned to the component indexed c, the description cost becomes: $\mathrm{LL}(\mathcal{D}|\hat{\mathcal{T}},\mathcal{C}) = nI(\boldsymbol{\alpha})+\sum_{c=1}^{k}\mathrm{LL}_c(\mathcal{D}|\mathcal{T}_c,\mathcal{C})$. Furthermore, the description length per component $LL_c(\mathcal{D}|\mathcal{T}_c,\mathcal{C})$ is linear in the number of model nodes. This allows us to pose the minimization of the description length as a linear optimization problem with a combinatorial constraint. In particular, as will be shown in the next section, we can pose the model-merging problem as an instance of a particular minimum edit-distance problem.

Given two tree models $\mathcal{T}_1$ and $\mathcal{T}_2$, we wish to construct a union $\hat{\mathcal{T}}$ whose structure respects the hierarchical constraints present in both $\mathcal{T}_1$ and $\mathcal{T}_2$, and that also minimizes the quantity $\mathrm{LL}(\hat{\mathcal{T}})$. Since the trees $\mathcal{T}_1$ and $\mathcal{T}_2$ already assign node correspondences $\mathcal{C}_1$ and $\mathcal{C}_2$ from the data samples to the model, we can

simply find a map $\mathcal{M}$ from the nodes in $\mathcal{T}_1$ and $\mathcal{T}_2$ to $\hat{\mathcal{T}}$ and transitively extend the correspondences from the samples to the final model $\hat{\mathcal{T}}$ in such a way that $\hat{\mathcal{C}}(v) = \hat{\mathcal{C}}(w) \Leftrightarrow w = \mathcal{M}(v)$.

Reduced to the merge of two structures, the correspondence problem is reduced to finding the set of nodes in $\mathcal{T}_1$ and $\mathcal{T}_2$ that are in common. Starting with the two structures, we merge the sets of nodes that would reduce the description length by the largest amount while still satisfying the hierarchical constraint. That is we merge nodes v and w of $\mathcal{T}_1$ with node v' and w' of $\mathcal{T}_2$ respectively if and only if $v \rightsquigarrow w \Leftrightarrow v' \rightsquigarrow w'$, where $a \rightsquigarrow b$ indicates that a is an ancestor of b.

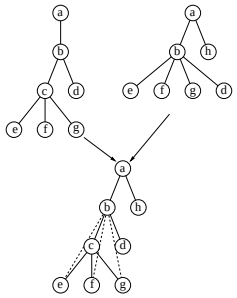


Fig. 1. Merging two trees.

Let n_1 and n_2 be the number of tree samples from $\mathcal{D}$ that are respectively assigned to $\mathcal{T}_1$ and $\mathcal{T}_2$. Further let p_v and $p_{v'}$ be the number of times the nodes v and v' in $\mathcal{T}_1$ and $\mathcal{T}_2$ are respectively in correspondence with nodes of trees in the data sample $\mathcal{D}$. The sampling probabilities for the two nodes, if they are not merged, are $\theta v = \frac{p_v}{n_1+n_2}$ and $\theta v' = \frac{p'_v}{n_1+n_2}$ respectively, while the sampling probability of the merged node is $\theta vv' = \frac{p_v+p_{v'}}{n_1+n_2}$. Hence, the description length advantage obtained by merging the nodes v and v' is:

$$\mathcal{A}(v, v') = (n_1 + n_2)\left[I(\theta v) + I(\theta v') - I(\theta vv')\right] + l. \tag{3}$$

This implies that the set of merges $\mathcal{M}$ that minimizes the description length of the combined model maximizes the advantage function

$$\mathcal{A}(\mathcal{M}) = \sum_{(v,v')\in\mathcal{M}} \mathcal{A}(v, v') = \sum_{(v,v')\in\mathcal{M}} [(n_1 + n_2)\left[I(\theta v) + I(\theta v') - I(\theta vv')\right] + l]. \tag{4}$$

Assuming that the class archetypes $\mathcal{T}_1$ and $\mathcal{T}_2$ are trees, finding the set of nodes to be merged can be transformed into a tree-edit distance problem. That is, assigning particular costs to node removal and matching operations, the set of correspondences that minimize the edit distance between the archetypes of $\mathcal{T}_1$ and $\mathcal{T}_2$ also maximizes the advantage of the merged model. The costs that allowed the problem to be posed as an edit distance problem are $r_v = (n_1 + n_2)I(\theta v) + l$ for the removal of node v, and $m_{vv'} = (n_1 + n_2)I(\theta vv') + l$ for matching node v with node v'. In the next section, we will discuss this relationship in more detail.

At the end of the node merging operation we are left with a set of nodes that respects the original partial order defined by all the hierarchies in the sample-trees. We initialize our algorithm by calculating the description length of a model in which there is a mixing component per tree-sample in $\mathcal{D}$. The description length is given by $-\log(n) + l\sum_{t\in\mathcal{D}} \#\mathcal{N}^t$, where $n = \#\mathcal{D}$ is the number of samples and $\#\mathcal{N}^t$ is the number of nodes in the tree-sample t. For each pair of initial mixture components we calculate the union and the description length of the merged structure. From the set of potential merges, we can identify the one which reduces the description cost by the greatest amount. The mixing proportion for this optimal merge is equal to the sum of the proportions of the individual

unions. At this point we calculate the union and description cost obtained merging the newly obtained model with each of the remaining components, and we iterate the algorithm until no more merges that reduce the description length can be found.

5 Tree Edit-Distance

As noted in earlier, the description length advantage is related to the edit distance between tree structures. This is an important observation. One of the difficulties with graph edit distance [17, 3] is that there is no methodology for assigning costs to edit operations. By contrast, in the work reported in this paper the description length change associated with tree merge operations are determined by the node probabilities, and these in turn may be estimated from the available sample of trees. Hence by establishing a link between tree edit distance and description length, we provide a means by which edit costs may be estimated.

Hence, in this section we review the computation of tree edit-distance developed in our previous work [22]. In particular, we describe how tree edit distance may be used to estimate node-correspondences, and give an overview of the algorithm we use to approximate the computation of tree edit distance.

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. This means that the constraints posed by the edit-distance framework on the set of matching nodes are equivalent to those required to merge nodes on the model archetypes. Namely, that they preserve the hierarchy present in the two original structures.

The edit-distance between two trees t and t' can be defined in terms of the matching nodes:

$$D(t,t') = \sum_{i \notin \mathrm{Dom}(\mathcal{M})} r_i + \sum_{j \notin \mathrm{Im}(\mathcal{M})} r_j + \sum_{<i,j> \in \mathcal{M}} m_{ij}. \tag{5}$$

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 $\mathrm{Dom}(\mathcal{M})$ and $\mathrm{Im}(\mathcal{M})$ 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

$$D(t,t') = \sum_{u \in \mathcal{N}^t} r_u + \sum_{v \in \mathcal{N}^{t'}} r_v + \sum_{(u,v) \in \mathcal{M}} (m_{uv} - r_u - r_v).$$

Hence the distance is minimized by the set of correspondences that maximizes the utility $\mathcal{U}(\mathcal{M}) = \sum_{(u,v)\in\mathcal{M}}(r_u + r_v - m_{uv})$.

Setting $r_u = (n_1 + n_2)I(\theta u) + l$, $r_v = (n_1 + n_2)I(\theta v) + l$, and $m_{uv} = (n_1 + n_2)I(\theta uv) + l$, we have

$$\mathcal{U}(\mathcal{M}) = \sum_{(u,v)\in\mathcal{M}} [(n_1 + n_2)(I(\theta u) + I(\theta v) - I(\theta uv)) + l], \tag{6}$$

which is equal to the advantage in description length in (4). Since the combinatorial problem underlying both edit-distance and model merge share the same hierarchical constraints and objective function, the solution to one problem can be derived from the solution to the other. In particular the set of common nodes obtained through the edit-distance approach is equal to the set of nodes to be merged to optimally merge the tree-models.

To find the set correspondences that minimizes the edit distance between two trees we make use of two results presented in [22]. We call $\Omega(t)$ the closure of tree t, $E_i(t)$ the edit operation that removes node i from t and $\mathcal{E}_i(\Omega(t))$ the equivalent edit operation that removes i from the closure. The first result is that edit and closure operations commute: $\mathcal{E}_i(\Omega(t)) = \Omega(E_i(t))$. For the second result we need some more definitions: We call a subtree s of $\Omega(t)$ *obtainable* if for each node i of s if there cannot be two children a and b so that (a, b) is in $\Omega(t)$. In other words, for s to be obtainable, there cannot be a path in t connecting two nodes that are siblings in s. We can, now, introduce the following:

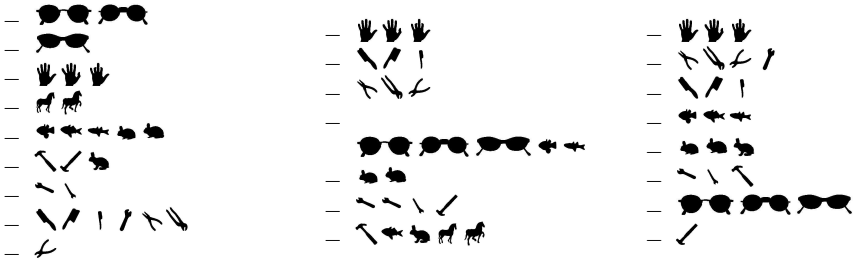
Theorem 1. *A tree $\hat{t}$ can be generated from a tree t with a sequence of node removal operations if and only if $\hat{t}$ is an obtainable subtree of the directed acyclic graph $\Omega(t)$.*

By virtue of the theorem above, the node correspondences yielding the minimum edit distance between trees t and t' form an obtainable subtree of both $\Omega(t)$ and $\Omega(t')$, hence we reduce the problem to the search for a common substructure that maximizes the utility: the maximum common obtainable subtree (MCOS). That is, Let O be the set of matches that satisfy the obtainability constraint, the node correspondence that minimized the edit distance is $\mathcal{M}^* = \text{argmax}_{\mathcal{M}\in O}\,\mathcal{U}(\mathcal{M})$.

The solution to this problem is obtained by looking for the best matches at the leaves of the two trees, and by then propagating them upwards towards the roots. 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. To propagate the matches to i and j we need to find the set of siblings with greatest total utility. This problem can be transformed into a maximum weighted clique problem on a derived structure and then approximated using a heuristical algorithm. When the matches have been propagated to all the pairs of nodes drawn from t and t', the set of matches associated with the maximum utility give the solution to the maximum common obtainable subtree problem, and hence the edit-distance. We refer to [22] for a detailed explanation of the approach.

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 [20]. 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 maximal 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 [20]. The later the time of formation, and hence their proximity to the center of the shape, the higher the shock in the hierarchy.



a) Mixture of unattributed tree models

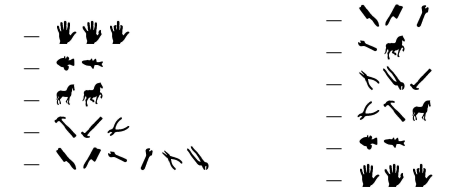
b) Weighted Edit-Distance

c) Union of attributed trees

Fig. 2. Clusters extracted with a purely-structural mixture of trees approach versus pairwise clustering of attributed distances obtained with edit distance and tree union.

In order to asses the quality of the method we compare clusters defined by the components of the mixture with those obtained with other two graph-clustering algorithms. The first graph-clustering method we compare to, is the one described in [22, 14]. This method extracts the clusters by applying a pairwise clustering algorithm to the matrix of edit-distances between the graphs. The second method extracts the clusters by applying the same pairwise clustering algorithm to a different distance matrix, namely the distance obtained from the embedding space defined by a single tree-union that encompasses every shape [23]. In our experiments the data clustered with the mixture of tree-unions approach use only structural information to characterize the shapes. On the other hand the cluster extracted using edit-distance and tree-union are based on data enhanced with geometrical information linked to the nodes of the trees.

Figure 2 shows the clusters extracted on a database of 25 shapes. The first column shows the clusters extracted through the mixture of trees approach on purely structural representation of shape. The second column displays the cluster extracted from the weighted edit-distances of shock-trees enhanced with geometrical information. The geometric information added to the nodes is the proportion of the border length that genetated the skeletal branch associated with the node. The third and last column shows the clusters extracted from the distances obtained by embedding the the geometrically-enhanced shock-trees in a single tree-union. While there is some merge and leakage, the cluster extracted with the mixture of trees method compare favorably with those obtained using the other two clustering algorithm, even where these are based on data enhanced with geometrical information. The second to last cluster extracted by the mixture of trees approach deserves some explanation: the structure of the shock-trees of the tools in the cluster is identical. Hence the model, which uses only structural infoprmation, correctly clusters the shock-trees toghether. To overcome this problem we need to provide more information than just the shock structure. The geometrical information allows the other methods to distinguish between wrenches, brushes and pliers.



c) Mixture of tree models

b) Pairwise clustering from edit-distance

Fig. 3. Clusters obtained from non-attributed edit-distance and mixture of trees.

Figure 3 compares the results of graph clustering performed on purely structural information only. Here the clusters obtained through the mixture of tree-unions approach (left)is compared with those extracted by pairwise clustering of unweighted edit-distance (right)[22]. No geometrical information used to aid the edit-distance-based clustering process. These results suggest that the mixture of tree-unions method outperforms pairwise clustering of edit-distance on purely structural data.

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 merge. To meet this goal we have randomly generated some prototype trees and, from each tree, we generated structurally perturbed copies. The procedure for generating the random trees was as follows: we commence with an empty tree (i.e. one with no nodes) and we iteratively add the required number of nodes. At each iteration nodes are added as children of one of the existing nodes. The parents are randomly selected with uniform probability from among the existing nodes. The trees are perturbed by randomly adding the required amount of nodes.

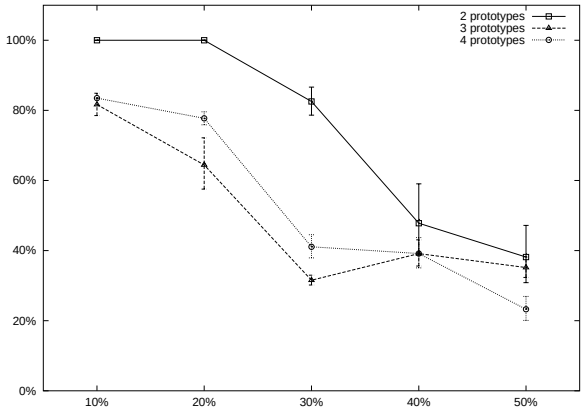


Fig. 4. Percentage of correct classifications under increasing structural noise.

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 4 plots the fraction of pairs of trees that are correctly classified as belonging to the same or different clusters as the noise is increased. 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.

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 description 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. H. G. Barrow and R. M. Burstall, Subgraph isomorphism, matching relational structures and maximal cliques, *Inf. Proc. Letters*, Vol. 4, pp.83, 84, 1976.

2. S. J. Dickinson, A. P. Pentlan, and A. Rosenfeld, 3-D shape recovery using distributed aspect matching, *PAMI*, Vol. 14(2), pp. 174-198, 1992.
3. M. A. Eshera and K-S Fu, An image understanding system using attributed symbolic representation and inexact graph-matching, *PAMI*, Vol 8, pp. 604-618, 1986.
4. N. Friedman and D. Koller, Being Bayesian about Network Structure, *Machine Learning*, to appear, 2002
5. L. Getoor et al., Learning Probabilistic models of relational structure, in *8th Int. Conf. on Machine Learning*, 2001.
6. 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.
7. T. Heap and D. Hogg, Wormholes in shape space: tracking through discontinuous changes in shape, in *ICCV*, pp. 344-349, 1998.
8. Ioffe, S. and Forsyth, D.A., Human Tracking with Mixtures of Trees, *ICCV*, Vol. I, pp. 690-695, 2001.
9. 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.
10. S. C. Johnson, Hierarchical clustering schemes, *Psychometrika*, Vol. 32(3), 1967.
11. B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, Shapes, shocks, and deformations I, *International Journal of Computer Vision*, Vol. 15, pp. 189-224, 1995.
12. P. Langley, W. Iba, and K. Thompson, An analysis of Bayesian classifiers, in *AAAI*, pp. 223-228, 1992
13. Liu, T. and Geiger, D. , Approximate Tree Matching and Shape Similarity, *ICCV*, pp. 456-462, 1999.
14. B. Luo, et al., A probabilistic framework for graph clustering, in *CVPR*, pp. 912-919, 2001.
15. M. Meilă. *Learning with Mixtures of Trees*. PhD thesis, MIT, 1999.
16. J. Riassen, Stochastic complexity and modeling, *Annals of Statistics*, Vol. 14, pp. 1080-1100, 1986.
17. A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs fro pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 13:353–362, 1983.
18. S. Sclaroff and A. P. Pentland, Modal matching for correspondence and recognition, *PAMI*, Vol. 17, pp. 545-661, 1995.
19. A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker, Indexing using a spectral encoding of topological structure, in *CVPR*, 1999.
20. K. Siddiqi et al., Shock graphs and shape matching, *Int. J. of Comp. Vision*, Vol. 35, 1999.
21. T. Sebastian, P. Klein, and B. Kimia, Recognition of shapes by editing shock graphs, in *ICCV*, Vol. I, pp. 755-762, 2001.
22. A. Torsello and E. R. Hancock, Efficiently computing weighted tree edit distance using relaxation labeling, in *EMMCVPR*, LNCS 2134, pp. 438-453, 2001.
23. A. Torsello and E. R. Hancock, Matching and embedding through edit-union of trees, in *ECCV*, LNCS 2352, pp. 822-836, 2002.
24. Zhu, S.C. and Yuille, A.L., FORMS: A Flexible Object Recognition and Modelling System, *IJCV*, Vol. 20(3), pp. 187-212, 1996.