Learning Mixtures of Weighted Tree-Unions by Minimizing Description Length

Andrea Torsello[1] and Edwin R. Hancock[2]

[1] Dipartimento di Informatica, Universita' Ca' Foscari di Venezia
via Torino 155, 30172 Venezia Mestre, Italy
torsello@dsi.unive.it
[2] Department of Computer Science, University of York
York YO10 5DD, England
erh@cs.york.ac.uk

Abstract. This paper focuses on how to perform the unsupervised clustering of tree structures in an information theoretic setting. We pose the problem of clustering as that of locating a series of archetypes that can be used to represent the variations in tree structure present in the training sample. The archetypes are tree-unions that are formed by merging sets of sample trees, and are attributed with probabilities that measure the node frequency or weight in the training sample. The approach is designed to operate when the correspondences between nodes are unknown and must be inferred as part of the learning process. We show how the tree merging process can be posed as the minimisation of an information theoretic minimum descriptor length criterion. We illustrate the utility of the resulting algorithm on the problem of classifying 2D shapes using a shock graph representation.

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 [10,15], the use of trees to represent articulated objects [8,19] 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 [6], have proved to be more amenable when variable sets of shapes must be analyzed. There are two reasons why pattern spaces are more easily constructed for curves and surfaces than for graphs. First, there is no canonical ordering for the nodes or edges of a graph. Hence,

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3023, pp. 13–25, 2004.

before a vector-space can be constructed, then correspondences between nodes must be established. Second, structural variations in graphs manifest themselves as differences in the numbers of nodes and edges. As a result, even if a vector mapping can be established then the vectors will be of variable length.

One way of circumventing this problem is to embed the graphs in a low dimensional space using the distances between graphs or by using simple graph features that do not require correspondence analysis. For instance, Cyr and Kimia have used a geometric procedure to embed graphs on a view-sphere [1]. Demerici and Dickinson [9] have shown how the minimum distortion embedding procedure of Linial, London and Rabinovich [11] can be used for the purposes of correspondence matching. A recent review of methods that could be used to perform the embedding process is provided in the paper of Hjaltason and Samet [7]. However, although this work provides a means of capturing the distribution of graphs and can be used for clustering, it does not provide an embedding which allows a generative model of detailed graph structure to be learned. In other words, the distribution does not capture in an explicit manner the variations in the graphs in terms of changes in node and edge structure. Recently, though, there has been considerable interest in learning structural representations from samples of training data, in particular in the context of Bayesian networks [5,3], mixtures of tree-classifiers [12], or general relational models [4]. Unfortunately, these methods require the availability of node correspondences as a prerequisite.

The aim in this paper is to develop an information theoretic framework for the unsupervised learning of generative models of tree-structures from sets of examples. We pose the problem as that of learning a mixture of union trees. Each tree union is an archetype that represents a class of trees. Those trees that belong to a particular class can be obtained from the relevant tree archetype by node removal operations. Hence, the union-tree can be formed using a sequence of tree merge operations. We work under conditions in which the node correspondences required to perform merges are unknown and must be located by minimising tree edit distance. Associated with each node of the union structure is a probability. This is a random variable which represents the frequency of the node in the training sample. Since every tree in the sample can be obtained from one of the union structures in the mixture, the tree archetypes are generative models. There are three quantities that must be estimated to construct this generative model. 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 using the description length for the union structure and its associated node probabilities given correspondences with the set of training examples [13]. With the tree-unions to hand, then we can apply use PCA to project the trees into a low dimensional vector space.

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 mixture of generative class archetypes. Each class archetype is constructed by merging sets of sample trees together to form a set of union structures. This merge process requires node correspondence information, and we work under conditions in which these are unknown and must be inferred as part of the learning process. Each tree in the sample can hence be obtained from one of the union-structures using a sequence of node removal operations. Thus the class archetypes are generative models since they capture in an explicit manner the structural variations for the sample trees belonging to a particular class in a probabilistic manner.

Suppose that the set of class archetypes constituting the mixture model is denoted by $\mathcal{H} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$. For the class c, the tree model $\mathcal{T}_c$ is a structural archetype derived from the tree-union obtained by merging the set of trees $\mathcal{D}_c \subseteq \mathcal{D}$ constituting the 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 $\mathcal{T}_c$. As a prerequisite, we require the set of node correspondences $\mathcal{C}$ between sample trees and the union tree for each class.

Our aim is to cast the learning process into an information theoretic setting. The estimation of the required class models is effected using a simple greedy optimization method. The quantity to be optimized is the descriptor 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 the samples in the set $\mathcal{D}$ 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 optimization 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.

We characterize uncertainties in the structure obtained by tree merge operations by assigning probabilities to nodes. By adopting an 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.

2.1 Probabilistic Framework

More formally, the basis of the proposed structural learning approach is a generative tree model which allows us to assign a probability distribution to a sample of hierarchical trees. Each hierarchical tree t is defined by a set of nodes $\mathcal{N}^t$, a tree-order relation $\mathcal{O}^t \subset \mathcal{N}^t \times \mathcal{N}^t$ between the nodes, and, in the case of weighted trees, a weight set $W^t = \{w_i^t | i \in \mathcal{N}^t\}$ where w_i^t is the weight associated with

node i of tree t. 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, if b is a descendent of a then it 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 no node between a and b in the tree-order.

Our aim is to construct a generative model for a class of trees $\mathcal{D}_c \subset \mathcal{D}$. The structural component of this model $\mathcal{T}_c$ consists of a set of nodes $\mathcal{N}_c$ and an associated tree order relation $\mathcal{O}_c \subset \mathcal{N}_c \times \mathcal{N}_c$. Additionally, there is a set $\Theta_c = \{\theta_i^c, i \in \mathcal{N}_c\}$ of sampling probabilities θ_i^c for each node $i \in \mathcal{N}_c$. Hence the model is the triple $\mathcal{T}_c = (\mathcal{N}_c, \mathcal{O}_c, \Theta_c)$. A sample from this model is a hierarchical tree $t = (\mathcal{N}^t, \mathcal{O}^t)$ with node set $\mathcal{N}^t \subset \mathcal{N}_c$ and a node hierarchy $\mathcal{O}^t$ that is the restriction to $\mathcal{N}^t$ of $\mathcal{O}_c$. In other words, the sample tree is just a subtree of the class archetype, which can be obtained using a simple set of edit operations that prune the archetype.

The develop our generative model we make a number of simplifying assumptions. First, we drop the class index c to simplify notation. Second, we assume that the set of nodes for the union structure $\mathcal{T}$ spans each of the encountered sample trees $\mathcal{D}$, i.e. $\mathcal{N} = \bigcup_{t \in \mathcal{D}} \mathcal{N}^t$. Third, 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 $i\mathcal{O}j$, then node i will be an ancestor of node j in each tree-sample that has both nodes.

Our assumptions imply that two nodes will always satisfy the same hierarchical relation whenever they are both present in a sample tree. A consequence of this assumption 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 viewed 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.

To define a probability distribution over the union structure $\mathcal{T}$, we require the correspondences between the nodes in each sample tree t and the nodes in the class-model $\mathcal{T}$. We hence define a map $\mathcal{C} : \mathcal{N}^t \to \mathcal{N}$ from the set $\mathcal{N}^t$ of the nodes of t, to the nodes of the class model $\mathcal{T}$. The mapping induces a sample-correspondence for each node $i \in \mathcal{N}$. When the nodes of the sample trees have weights associated with them, then we would expect the sampling likelihood to reflect the distribution of weights. Hence, the simple probability distribution described above, which is based on uniform sample node probability, is not sufficient because it does not take into account the weight distribution.To overcome this shortcoming, in addition to the set of sampling probabilities Θ, we associate with the union model a weight distribution function. Here we assume that the weight distribution is a rectified Gaussian. For the node i of the union tree the weight probability distribution is given by

$$p(w_j|C(j) = i) \begin{cases} \frac{1}{\theta_i \sigma_i \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(w_j - \mu_i)^2}{{\sigma_i}^2}\right) & \text{if } w_j \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where the weight distribution has mode μ_i and standard deviation σ_i. The sampling probability is the integral of the distribution over positive weights, i.e.

$$\theta_i = \int_0^\infty \frac{\exp\left(-\frac{1}{2}\frac{(w-\mu_i)^2}{{\sigma_i}^2}\right)}{\sigma_i\sqrt{2\pi}}dw = 1 - \text{erfc}(\tau_i), \tag{1}$$

where $\tau_i = \mu_i/\sigma_i$ and erfc is the complementary error function. Taking into account the correspondences, the probability for node i induced by the mapping is

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

2.2 Estimating Node Parameters

We can compute the log-likelihood of the sample data $\mathcal{D}$ given the tree-union model $\mathcal{T}$ and the correspondence mapping function $\mathcal{C}$. Under the assumption that the sampling process acts independently on the nodes of the structure the log-likelihood is

$$\mathcal{L}(\mathcal{D}|\mathcal{T},\mathcal{C}) = \sum_{t\in\mathcal{D}}\sum_{i\in\mathcal{N}^t} \ln\left[\phi(i|t,\mathcal{T},\mathcal{C})\right]$$

Our ultimate aim is to optimize the log-likelihood with respect to the correspondence map $\mathcal{C}$ and the tree union model $\mathcal{T}$. These variables, though, are not independent since they both depend on the node-set $\mathcal{N}$. A variation in the actual identity and number of the nodes does not change the log-likelihood. Hence the dependency on the node-set can be lifted by simply assuming that the node set is the image of the correspondence map i.e. $\text{Im}(\mathcal{C})$. As we will see later, the reason for this is that those nodes that remain unmapped do not affect the maximization process.

We defer details of how we estimate the correspondence map $\mathcal{C}$ and the order relation $\mathcal{O}$ to later sections of the paper. However, assuming estimates of them are to hand, then we can make maximum likelihood estimates of the selected node model. That is, the set of sampling probabilities Θ in the unweighted case, and the node parameters $\bar{\tau}$ and $\bar{\sigma}$ in the weighted case.

To proceed, let $K_i = \{j \in \mathcal{N}^t | t \in \mathcal{D}, C(j) = i\}$ be the set of nodes in the different trees for which $\mathcal{C}$ maps a node to i and let $p_i = |K_i|$ be the number of trees satisfying this condition. Further, let n_i be the number of trees in $\mathcal{D}$ for which $\mathcal{C}$ results in no mapping to the node i. Using the *weighted* node model, the log-likelihood function can be expressed as the sum of per-node log-likelihood functions

$$\mathcal{L}(\mathcal{D}|\mathcal{T},\mathcal{C}) = \sum_{i\in\mathcal{N}} \log\left(\text{erfc}(\tau_i)^{n_i}(2\pi\sigma_i)^{-\frac{p_i}{2}}\exp\left[-\frac{1}{2}\sum_{j\in K_i}\left(\frac{w_j^t}{\sigma_i}-\tau_i\right)^2\right]\right). \tag{2}$$

To estimate the parameters of the weight distribution, we take the derivatives of the log-likelihood function with respect to σ_i and τ_i and set them to zero. When $n_i > 0$, we maximize the log likelihood by setting ${\tau_i}^0 = \operatorname{erfc}^{-1}\left(\frac{n_i}{n_i+p_i}\right)$, and iterating the recurrence:

$$ {\sigma_i}^{(k)} = -\frac{{\tau_i}^{(k)}}{2}\overline{W} + \sqrt{\left(\frac{{\tau_i}^{(k)}}{2}\overline{W}\right)^2 + \overline{W^2}} \qquad {\tau_i}^{(k+1)} = {\tau_i}^{(k)} - \frac{f({\tau_i}^{(k)}, {\sigma_i}^{(k)})}{\frac{d}{d{\tau_i}^{(k)}} f({\tau_i}^{(k)}, {\sigma_i}^{(k)})} \tag{3} $$

where $\overline{W} = \sum_{j\in K_i} w_j^t$, $\overline{W^2} = \sum_{j\in K_i} \left(w_j^t\right)^2$, and $f(\tau_i, \sigma_i) = n_i \operatorname{erfc}'(\tau_i) + p_i \operatorname{erfc}(\tau_i)\left(\frac{\overline{W}}{\sigma_i} - \tau_i\right)$.

3 Mixture Model

We now commence our discussion of how to estimate the order relation $\mathcal{O}$ for the tree union $\mathcal{T}$, and the set of correspondences $\mathcal{C}$ needed to merge the sample trees to form the tree-union. We pose the problem as that of fitting a mixture of tree unions to the set of sample trees. Each tree-union may be used to represent a distribution of trees that belong to a single class $\mathcal{D}_c$. The defining characteristic of the class is the 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 multiple classes. We represent the distribution of sample trees using a mixture model over separate union structures. Suppose that there are k tree-unions and that the tree union for the class c is denoted by $\mathcal{T}_c$, and that the mixing proportion for this tree-union is α_c. The mixture model for the distribution of sample trees is

$$ P(t|\bar{\mathcal{T}}, \mathcal{C}) = \sum_{c=1}^{k} \alpha_c \prod_{t\in\mathcal{D}} \prod_{i\in\mathcal{N}^t} \phi(i|t, \mathcal{T}_c, \mathcal{C}). $$

The expected log-likelihood function for the mixture model over the sample-set $\mathcal{D}$ is:

$$ \mathcal{L}(\mathcal{D}|\bar{\mathcal{T}}, \mathcal{C}, \bar{z}) = \sum_{t\in\mathcal{D}} \sum_{i\in\mathcal{N}^t} \sum_{c=1}^{k} z_c^t \alpha_c \ln \phi(i|t, \mathcal{T}_c, \mathcal{C}), $$

where z_c^t is an indicator variable, that takes on the value 1 if tree t belongs to the mixture component c, and is zero otherwise.

We hence require an information criterion that can be used to select the set of tree merge operations over the sample set $\mathcal{D}$ that results in the optimal set of tree-unions. 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 the Minimum Description Length (MDL) principle [13], which 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. The MDL principle allows us to select from a family of possibilities the most parsimonious model that best approximates the underlying data.

More formally, the expected descriptor length of a data set $\mathcal{D}$ generated by an estimate $\mathcal{H}$ of the true or underlying model $\mathcal{H}^*$ is

$$E\left[\mathrm{LL}(\mathcal{D},\mathcal{H})\right] = -\int P(\mathcal{D}|\mathcal{H}^*)\log\left[P(\mathcal{D}|\mathcal{H})P(\mathcal{H})\right]\,d\mathcal{D} =$$

$$-\frac{1}{P(\mathcal{H}^*)}\int P(\mathcal{D},\mathcal{H}^*)\log\left[P(\mathcal{D},\mathcal{H})\right]\,d\mathcal{D} =$$

$$-\frac{1}{P(\mathcal{H}^*)}\left[\int P(\mathcal{D},\mathcal{H}^*)\log\left(P(\mathcal{D},\mathcal{H}^*)\right)\,d\mathcal{D}+\int P(\mathcal{D},\mathcal{H}^*)\log\left(\frac{P(\mathcal{D},\mathcal{H})}{P(\mathcal{D},\mathcal{H}^*)}\right)\,d\mathcal{D}\right] =$$

$$\frac{1}{P(\mathcal{H}^*)}\left[I(P(\mathcal{D},\mathcal{H}^*)) + KL(P(\mathcal{D},\mathcal{H}^*),P(\mathcal{D},\mathcal{H}))\right], \quad (4)$$

where

$$I(P(\mathcal{D},\mathcal{H}^*)) = -\int P(\mathcal{D},\mathcal{H}^*)\log\left(P(\mathcal{D},\mathcal{H}^*)\right)\,d\mathcal{D}$$

is the entropy of the joint probability of the data and the underlying model $\mathcal{H}^*$, and

$$KL(P(\mathcal{D},\mathcal{H}^*),P(\mathcal{D},\mathcal{H})) = -\int P(\mathcal{D},\mathcal{H}^*)\log\left(\frac{P(\mathcal{D},\mathcal{H})}{P(\mathcal{D},\mathcal{H}^*)}\right)\,d\mathcal{D}$$

is the Kullback-Leiber divergence between the joint probabilities using the underlying model $\mathcal{H}^*$ and the estimated model $\mathcal{H}$. This quantity is minimized when $\mathcal{H}=\mathcal{H}^*$, and hence $P(\mathcal{D},\mathcal{H}) = P(\mathcal{D},\mathcal{H}^*)$.

Under these conditions $KL(P(\mathcal{D},\mathcal{H}^*),P(\mathcal{D},\mathcal{H})) = 0$ and $E[LL(\mathcal{D},\mathcal{H})] = I(P(\mathcal{D},\mathcal{H}))$. In other words, the description length associated with the maximum likelihood set of parameters is just the expected value of the negative log likelihood, i.e. the Shannon entropy.

As noted above, the cost incurred in describing or encoding the model $\bar{\mathcal{T}}$ is $-\log\left[P(\bar{\mathcal{T}})\right]$, while the cost of describing the data $\mathcal{D}$ using that model is $-\log\left[P(\mathcal{D}|\bar{\mathcal{T}})\right]$. Making the dependence on the correspondences $\mathcal{C}$ explicit, we have that the description length is $LL(\mathcal{D}|\mathcal{T}) = -\mathcal{L}(\mathcal{D}|\bar{\mathcal{T}},\mathcal{C})$.. Asymptotically the cost of describing the set of mixing components $\bar{\alpha} = \{\alpha_c; c = 1,...,k\}$ and the set of indicator variables $\bar{z} = \{z_c^t | t \in \mathcal{D}, c = 1,...,k\}$ is bounded by $mI(\bar{\alpha})$, where m 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 $\bar{\alpha}$. We assume that the weight distribution is encoded as a histogram. Hence, we commence by dividing the weight space of the samples associated with the node i of the union-tree c into buckets of

width $k\sigma_i^c$. As a result, the probability that a weight falls in a bucket centered at x is, for infinitesimally small k $b_c^i(x) = \frac{k}{\theta_i^c\sqrt{2\pi}} \exp[-\frac{1}{2}(\frac{x}{\sigma_i^c} - \tau_i^c)^2]$. Hence, the asymptotic cost of describing the node parameters τ_i^c and σ_i^c and, at the same time, describing within the specified precision the $n\alpha_c$ samples associated to node i in union c, is

$$\mathrm{LL}_c^i(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) = -(m\alpha_c - p_i)\log(1-\theta_i^b) - \sum_{j=1}^{p_i} \log\left(b_c^i(w_j^i)\right).$$

where $\theta_i^c = 1-\mathrm{erfc}(\tau_i)$ is the sampling probability for node i and p_i is the number of times the correspondence $\mathcal{C}$ maps a sample-node to i. Hence $(m\alpha_c - p_i)$ is the number of times node i has not been sampled according to the correspondence map $\mathcal{C}$. As a result

$$\mathrm{LL}(\mathcal{D}|\mathcal{H}, \mathcal{C}) = mI(\bar{\alpha}) + \sum_{c=1}^{k} \sum_{i\in\mathcal{N}_c} \left[\mathrm{LL}_c^i(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) + l\right]. \tag{5}$$

where l is the description length per node of the tree-union structure, which we set to 1.

4 Learning the Mixture

With the description length criterion to hand, our aim is to locate tree merges that give rise to the set of tree unions that optimally partition the training data $\mathcal{D}$ into non-overlapping classes. Unfortunately, locating the global minimum of the descriptor length in this way is an intractable combinatorial problem. Moreover, the Expectation-Maximization algorithm may not be used since the complexity of the maximization step grows exponentially due to the fact that the membership indicators admit the possibility that each union can potentially include every sample-tree. Hence, we resort to a local search technique, which allows us to limit the complexity of the maximization step. The approach 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 unit sample probability.
- Iteratively generalize the model by merging pairs of tree-unions. The candidates for merging are chosen so that they maximally decrease the descriptor length.
- The algorithm stops when there are no merges remaining that can decrease the descriptor length.

The main requirement of our description length minimization algorithm is that we can optimally merge two tree models. 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, given two nodes $v \in \mathcal{N}_1$ and $v' \in \mathcal{N}_2$, then $\hat{\mathcal{C}}(v) = \hat{\mathcal{C}}(v') \Leftrightarrow v' = \mathcal{M}(v)$.

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

The descriptor length advantage obtained by merging the nodes v and v' is:

$$\mathcal{A}(v, v') = \mathrm{LL}^{v}(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) + \mathrm{LL}^{v'}(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) - \mathrm{LL}^{(vv')}(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) + l. \tag{6}$$

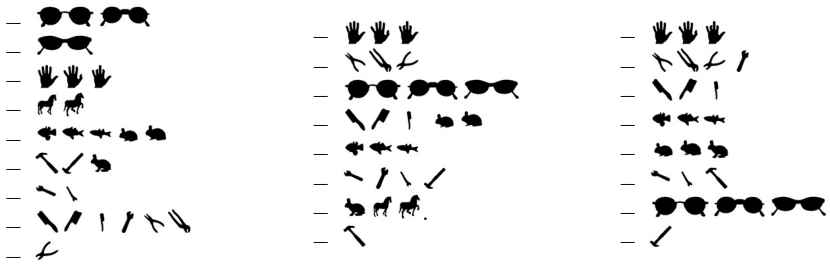
The set of merges $\mathcal{M}$ that minimizes the descriptor length of the combined tree-union also maximizes the advantage function

$$\mathcal{A}(\mathcal{M}) = \sum_{(v,v') \in \mathcal{M}} \mathcal{A}(v, v').$$

For each pair of initial mixture components we calculate the union and the descriptor length of the merged structure. From the set of potential merges, we can identify the one which is both allowable and which reduces the descriptor 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 descriptor cost that results from merging the newly obtained model with each of the remaining components. We iterate the algorithm until no more merges can be found that reduce the descriptor length.

5 Pattern Spaces from Union Trees

We can uses the union-trees to embed the shapes of the same class in a pattern space using principal components analysis. To do this we place the nodes of the union tree $\mathcal{T}_c$ in an arbitrary order. To each sample tree t we associate a pattern-vector $\boldsymbol{x}_t = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, where $n = |\mathcal{N}_c|$ is the number of nodes in the tree model $\mathcal{T}_c$. Here $x_t(i) = w_i^T$ if the tree has a node mapped to the i-th node of the sample and is zero otherwise. For each union-tree $\mathcal{T}_c$ we compute the mean pattern-vector $\hat{\boldsymbol{x}}_c = \frac{1}{|\mathcal{N}_c|} \sum_{t \in \mathcal{N}_c} \boldsymbol{x}_t$ and covariance matrix $\Sigma_c = \frac{1}{|\mathcal{N}_c|} \sum_{t \in \mathcal{N}_c} (\boldsymbol{x}_t - \hat{\boldsymbol{x}}_c)(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_c)^T$ where $\mathcal{N}_c$ is the set of sample trees merged to form the tree union $\mathcal{T}_c$. Suppose that the eigenvectors (ordered to decreasing eigenvalue) are $\boldsymbol{e}_1, \boldsymbol{e}_2, ...\boldsymbol{e}_{\mathcal{N}_c}$. The leading l_{sig} eigenvectors are used to form the columns of the matrix $E = (\boldsymbol{e}_1|\boldsymbol{e}_2|....|\boldsymbol{e}_{l_{sig}})$. We perform PCA on the sample-trees by projecting the pattern-vectors onto the leading eigenvectors of the covariance matrix. The



a) Mixture of unattributed tree models. b) Weighted edit-distance. c) Union of attributed trees.

Fig. 1. Clusters extracted with a purely-structural mixture of trees approach versus pairwise clustering of attributed distances obtained with edit distance and tree union.

projection of the pattern-vector for the sample tree indexed t is $\boldsymbol{y}_t = E^T \boldsymbol{x}_t$. The distance between the vectors in this space is $D^{PCA}(t,t')(\boldsymbol{y}_t - \boldsymbol{y}_{t'})^T(\boldsymbol{y}_t - \boldsymbol{y}_{t'})$.

6 Experimental Results

We illustrate the utility of the tree-clustering algorithm on sets of shock trees. The shock tree is a graph-based representation of the differential structure of the boundary of a 2D shape. We augment the skeleton topology with a measure of feature importance based on the rate of change of boundary length with distance along the skeleton.

6.1 Clustering Examples

To illustrate the clustering process, we commence with a study on a small database of 25 shapes. In order to asses the quality of the method, we compare the clusters defined by the components of the mixture with those obtained by applying a graph spectral pairwise clustering method recently developed by Robles-Kelly and Hancock [14] to the distances between graphs. This method locates the clusters by iteratively extracting the eigenvectors from the matrix of edit-distances between the graphs. The edit distances are computed in two alternative ways. First, we compute weighted edit distance using the method outlined in [17]. The second method involves computing the distance matrix using the projected vectors by embedding the trees in a single tree union [18]. These two distance measures are enhanced with geometrical information linked to the nodes of the trees in the form of a node weight. The weight of each node is equal to the proportion of the boundary length that generated the skeletal branch associated to the node.

Figure 1 shows the clusters extracted from the database of 25 shapes. The first column shows the clusters extracted through the mixture of tree unions

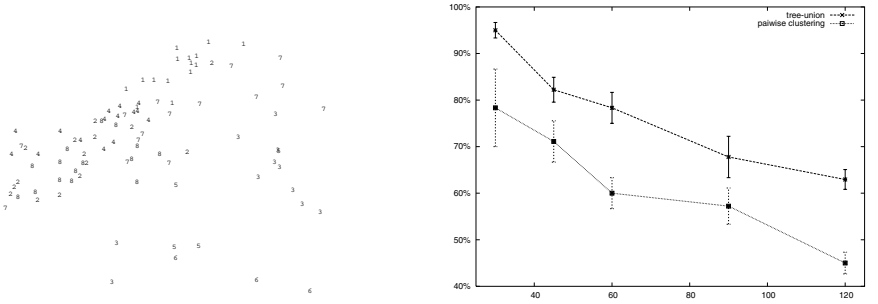


Fig. 2. Left: 2D multi-dimensional scaling of the pairwise distances of the shock graphs. (The numbers correspond to the shape classes.); Right: Proportion of correct classifications obtained with the mixture of tree versus those obtained with pairwise clustering.

approach, and relies on a purely structural representation of shape. The second column displays the clusters extracted from the weighted edit-distances between shock-trees; here the structural information is enhanced with geometrical information. The third column shows the clusters extracted from the distances obtained by embedding the geometrically-enhanced shock-trees in a single tree-union. While there is some merge and leakage, the clusters extracted with the mixture of tree unions compare favorably with those obtained using the alternative clustering algorithms, even though these are based on data enhanced with geometrical information. The second to last cluster extracted using the mixture of tree unions deserves some further explanation. The structure of the shock-trees of the distinct tools in the cluster are identical. Hence, by using only structural information, the method clusters the shock-trees together. To distinguish between the objects, geometrical information must be provided too. Hence, the two alternative clustering methods are able to distinguish between the wrenches, brushes and pliers.

A more challenging experimental vehicle is provided by a larger database of 120 trees, which is divided into 8 shape classes containing 15 shapes each. To perform an initial evaluation of this database, we have applied multidimensional scaling to the weighted edit distances between the shock graphs for the different shapes. By doing this we embed points representing the graphs in a low dimensional space spanned by the eigenvectors of a similarity matrix computed from the pairwise distances. In Figure 2 we show the projection of the graphs onto the 2D space spanned by the leading two eigenvectors of the similarity matrix. Each label in the plot corresponds to a particular shape class. Label 1 identifies hands, label 2 horses, label 3 ducks, 4 men, 5 pliers, 6 screwdrivers, 7 dogs, and, finally, label 8 is associated with leaves. The plot clearly shows the difficulty of this clustering problem. The shape groups are not well separated. Rather, there is a good deal of overlap between them. Furthermore, there are a considerable number of outliers.

To asses the ability of the clustering algorithm to separate the shape classes, we performed experiments on an increasing number of shapes. We commenced

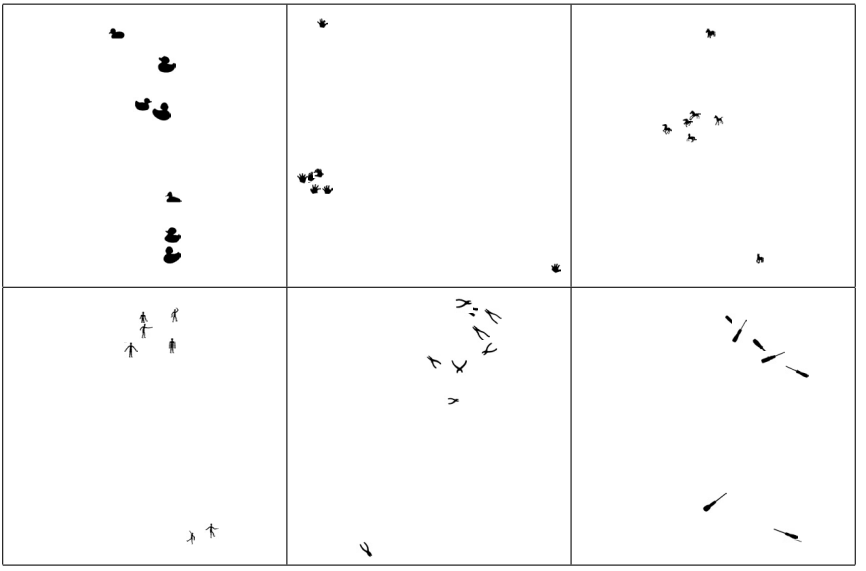


Fig. 3. Principal components analysis of the union embedding of the clusters.

with the 30 shapes from the first two shape classes, and then increased the number of shape classes under consideration until the full set of 120 shapes was included. Figure 2 plots the proportion of shapes correctly classified as the number of shapes is increased. The solid line plots the result obtained using the mixture of weighted tree unions, while the dotted line displays the results obtained with pairwise clustering of the weighted edit distances between the shapes. The mixture of tree unions clearly outperforms the pairwise clustering algorithm.

We now turn our attention to the results of applying PCA to the union trees, as described in Section 5. Figure 3 displays the first two principal components of the sample-tree distribution for the embedding spaces extracted from six shape classes. In most cases there appears to be a tightly packed central cluster with a few shapes scattered further away than the rest. This separation is linked to substantial variations in the structure of the shock trees. For example, in the shape-space formed by the class of pliers the outlier is the only pair-of-pliers with the nose closed. In the case of shape-space for the horse-class, the outliers appear to be the cart-horses while the inliers are the ponies.

7 Conclusions

In this paper we have presented an information theoretic framework for clustering trees and for learning a generative model of the variation in tree structure. The problem is posed as that of learning a mixture of tree unions. We demonstrate how the three sets of operations needed to learn the generative model,

namely node correspondence, tree merging and node probability estimation, can each be couched in terms of minimising a description length criterion. We provide variants of algorithm that can be applied to samples of both weighted and unweighted trees. The method is illustrated on the problem of learning shape-classes from sets of shock trees.

References

1. C. Cyr and B.Kimia, 3D Object Recognition Using Shape Similarity-Based Aspect Graph, ICCV 2001.
2. S. J. Dickinson, A. P. Pentland, and A. Rosenfeld, 3-D shape recovery using distributed aspect matching, *PAMI*, Vol. 14(2), pp. 174-198, 1992.
3. N. Friedman and D. Koller, Being Bayesian about Network Structure, *Machine Learning*, to appear, 2002
4. L. Getoor et al., Learning Probabilistic models of relational structure, in *8th Int. Conf. on Machine Learning*, 2001.
5. 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.
6. T. Heap and D. Hogg, Wormholes in shape space: tracking through discontinuous changes in shape, in *ICCV*, pp. 344-349, 1998.
7. G.R. Hjaltason and H. Samet, Properties of embedding methods for similarity searching in metric spaces, PAMI(25), pp. 530-549, 2003.
8. S. Ioffe and D. A. Forsyth, Human Tracking with Mixtures of Trees, *ICCV*, Vol. I, pp. 690-695, 2001.
9. Y. Keselman, A. Shokoufandeh, M.F. Demirci, and S.Dickinson, Many-to-many graph matching via metric embedding, CVPR03(I: 850-857).
10. 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.
11. N. Linial, E. London and Y. Rabinovich, The geometry of graphs and some of its applications, 35th Anual Symposium on Foundations of Computer Science, pp. 169–175, 1994.
12. M. Meilă. *Learning with Mixtures of Trees*. PhD thesis, MIT, 1999.
13. J. Rissanen, Stochastic complexity and modeling, *Annals of Statistics*, Vol. 14, pp. 1080-1100, 1986.
14. A. Robles-Kelly and E. R. Hancock. A maximum likelihood framework for iterative eigendecomposition. In *ICCV*, Vol. I, pp. 654-661, 2001.
15. A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker, Indexing using a spectral encoding of topological structure, in *CVPR*, 1999.
16. T. Sebastian, P. Klein, and B. Kimia, Recognition of shapes by editing shock graphs, in *ICCV*, Vol. I, pp. 755-762, 2001.
17. A. Torsello and E. R. Hancock. Efficiently computing weighted tree edit distance using relaxation labeling. In *EMMCVPR*, pp. 438-453, 2001.
18. A. Torsello and E. R. Hancock, Matching and embedding through edit-union of trees. In *ECCV*, pp. 822-836, 2002.
19. S. C. Zhu and A. L. Yuille, FORMS: A Flexible Object Recognition and Modelling System, *IJCV*, Vol. 20(3), pp. 187-212, 1996.