7 Conclusion

In this paper we have compared two representations of 3D segmented images. Both models use equivalent topological representations which are 3D topological maps described by combinatorial maps. We have given the relation between the models and the correspondence allowing to convert each model into the other one. These models differ in the way of representing the geometry of 3D segmented images. We have compared geometrical embeddings in use in each model and we have defined correspondence between themselves. Finally we have proposed to unify these models in order to get the advantages of each of them. A promising direction is to adapt the map construction used in the HLE model to improve and generalize the construction of the boundary image. Another aspect that has not been considered in this paper but that must be developed is the contribution to the unification of both models to their extension to 4D images.

References

[1] Y. Bertrand, G. Damiand, and C. Fiorio. Topological encoding of 3d segmented images. In *Discrete Geometry for Computer Imagery*, number 1953 in Lecture Notes in Computer Science, pages 311–324, Uppsala, Sweden, december 2000.

[2] J.P. Braquelaire and L. Brun. Image segmentation with topological maps and inter-pixel representation. *Journal on Visual Communication and Image Representation*, 9(1):62–79, 1998.

[3] J.P. Braquelaire, P. Desbarats, J.P. Domenger, and C.A. Wüthrich. A topological structuring for aggregates of 3D discrete objects. In *Proc. of the 2nd IAPR-TC-15 Workshop on Graph Based Representation*, pages 193–202. Österreichische Computer Gesellschaft, 1999. ISBN 3-85804-126-2.

[4] L. Brun, J.P. Domenger, and J.P. Braquelaire. Discrete maps : a framework for region segmentation algorithms. In J.-M. Jolion and W.G. Kropatsch, editors, *Graph Based Representation in Pattern Recognition, GBR'97*, volume 12 of *Computing Suplement*, pages 83–92. SpingerWienNewYork, 1998.

[5] R. Cori. Un code pour les graphes planaires et ses applications. In *Astérisque*, volume 27. Soc. Math. de France, Paris, France, 1975.

[6] G. Damiand. *Définition et étude d'un modèle topologique minimal de représentation d'images 2D et 3D.* Thèse de doctorat, Université Montpellier II, décembre 2001.

[7] P. Desbarats. *Strucuration d'images segmentées 3D discrètes.* Thèse de doctorat, Université Bordeaux 1, décembre 2001.

[8] J. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices of the American Mathematical Society*, 7, 1960.

[9] A. Jacques. Constellations et graphes topologiques. In *Combinatorial Theory and Applications*, volume 2, pages 657–673, 1970.

[10] P. Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Commputer Aided Design*, 23(1):59–82, 1991.

[11] W.T. Tutte. A census of planar maps. *Canad. J. Math.*, 15:249–271, 1963.

Tree Edit Distance from Information Theory

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 presents a method for estimating the cost of tree edit operations. The approach poses the problem as that of estimating a generative model for a set of tree samples. The generative model uses the tree-union as the structural archetype for every tree in the distribution and assigns to each node in the archetype the probability that the node is present in a sample. A minimum descriptor length formulation is then used to estimate the structure and parameters of this tree model as well as the node-correspondences between trees in the sample-set and the tree model.

1 Introduction

The task of measuring the similarity or distance between pairs of graphs has attracted considerable interest over the past three decades [1,2,5,17,3]. It is an important problem since the availability of reliable distance information allows a number of practical problems involving graphs to be addressed. These include graph-matching, graph retrieval and graph clustering.

Some of the earliest work on the problem was presented by Shapiro and Haralick [19] who demonstrated how counting consistent subgraphs could lead to a metric distance between graphs. Fu and his co-workers [5,17] devoted considerable effort to extending the concept of string edit distance to graphs. The contribution here was to show how the distance between graphs could be gauged using a series of costs for edge and node relabeling, and for edge and node insertion or deletion. However, the estimation of the necessary costs remains an open problem.

This early work, drew its inspiration from structural pattern recognition, which was at the time a field in its infancy. Subsequent work, aimed to take a more principled approach to the problem by drawing on information theory and probabilistic models. For instance, adopting an information theoretic framework, Wong and You [25] have shown how the cost of edit operations can be measured using changes in entropy. The probabilistic analysis of the graph-matching problem has been the focus of recent activity. For instance, Christmas, Kittler and Petrou [4] have developed a probabilistic relaxation scheme which measures the compatibility of matching using a distribution function for pairwise attributes residing on the edges of graphs. Wilson and Hancock [24], have developed a discrete relaxation algorithm in which the distribution of correspondence errors and structural errors are modeled probabilistically.

E. Hancock and M. Vento (Eds.): IAPR Workshop GbRPR 2003, LNCS 2726, pp. 71–82, 2003.

Recently there has been renewed interest in placing the concept of graph edit distance on a more rigorous footing, One of the criticisms that can be leveled at the early work is that it lacks the formality of the treatment which underpins the string edit distance. Bunke and Kandel [3] have demonstrated the relationship between the size of the maximum common subgraph and edit distance under the assumption of uniform edit costs. Myers, Wilson and Hancock [14] have shown how string edit distance can be used to model the distribution of structural error in graph neighborhoods. Robles-Kelly and Hancock [16] have shown how to use eigenvectors of the adjacency matrix to convert graphs to strings, so that their similarity may be assessed using string edit distance.

However, despite this recent work the question of how to estimate edit costs remains largely unanswered. This can be viewed as a problem of machine learning, where the costs associated with edit operations can be estimated from the frequency of occurrence of the corresponding edit operation in a set of training data. In this paper, we describe an information theoretic framework which can be used to estimate tree edit distances. The approach is as follows. We pose the problem of learning the edit operations as that of constructing a union-tree over a set of example trees. Associated with each node in the union structure is a probability which indicates the frequency of the node in the tree sample. The merging of trees to form the union structure is cast as the problem of locating a structure that minimizes description length. The recovery of the optimal union structure involves three update steps. First, correspondences between the sample trees and the current union structure must be located. Second, the node probabilities must be updated, Thirdly, and finally, the set of edit operations that result in the optimal union must be identified. These three steps may each be formulated in terms of the description length. Moreover, the cost associated with the edit operations is related to the description length advantage, and this in turn is given by the node probabilities. Hence, by estimating node probabilities we learn the edit costs.

2 Tree Edit-Distance

The idea behind edit distance is that it is possible to identify a set of basic edit operations on the 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. In previous work we have shown that the optimal sequence can be found using only structure reducing operations [23]. 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.

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

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

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

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

$$\mathcal{U}(\mathcal{M}) = \sum_{(u,v) \in \mathcal{M}} (r_u + r_v - m_{uv}). \quad (2)$$

3 Generative Tree Model

The aim of this paper is to learn the cost of the tree edit operations described above. We do this by fitting a generative model of tree structures to a set of sample-trees. This allows us to globally estimate the node correspondences solving the identification problem, and determine the modes of structural variation in the distribution of tree structures. 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 training 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.

Consider the set or sample of trees $\mathcal{D} = \{t_1, t_2, \ldots, t_n\}$. Our aim is to learn a generative model for the distribution of trees in a pattern space. The tree model must be capable of capturing the structural variations for the sample trees using a probability distribution. The information on the structural variation can then be used to determine the cost of the edit operations. The resulting approach assigns low cost to operations associated to common variations, and higher cost to more unlikely variations.

In prior work, we have described how tree unions can be used as structural models for samples of trees [22]. However, the union is constructed so as to minimize tree-edit distance. Here we intend to use the union structure as a class model, and use the model to derive the distance. This approach extends 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.

A 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

ution which captures the variations in tree structure within the class. e, the learning process involves estimating the union structure and the ameters of the associated probability distribution for the class model $\mathcal{T}$. As a erequisite, we require the set of node correspondences $\mathcal{C}$ between sample trees and the union tree.

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 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 samples and the archetype. The inter-sample node correspondences are not assumed to be known a priori. Since the correspondences are unknown, 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.

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.

Hence, the structural model is provided by the tree-union of the set of samples, 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 [17,5].

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 subset of the samples $\mathcal{D}_c \subset \mathcal{D}$. This model $\mathcal{T}$ 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 $i \in \mathcal{N}$.

$$\mathcal{T} = (\mathcal{N}, \mathcal{O}, \Theta) \tag{3}$$

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:

$$P\{t|\mathcal{T}\} = \prod_{i \in \mathcal{N}^t} \theta^i \prod_{j \in (\mathcal{N} \setminus \mathcal{N}^t)} (1 - \theta^j) \tag{4}$$

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 ancestor 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 extracting a sample from the 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 $\mathcal{C} : \mathcal{N}^t \to \mathcal{N}$ 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

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

4 Minimizing the Descriptor Length

The generative tree model provides us with a probability distribution for tree structure. 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 correspondence $\mathcal{C}$ from the nodes of the sample trees to the nodes of the tree model.

Here we use a minimum descriptor length approach. The MDL principle [15] 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.

By virtue of Shannon theorem, the cost of describing data $\mathcal{D}$ using model $\mathcal{T} = \{\mathcal{N}, \mathcal{O}, \Theta\}$ is $-\sum_{t \in \mathcal{D}} \log[\Phi(t|\mathcal{T}, \mathcal{C})] = -\mathcal{L}(\mathcal{D}|\mathcal{T}, \mathcal{C})$, while the cost incurred describing or encoding the model is $-\log[P(\mathcal{T})]$. Here we assume that the tree probability decays exponentially with the number of node. That is $P(\mathcal{T}) = \exp(-ld)$, where d is the number of nodes in $\mathcal{T}$. Using this prior, the cost incurred in describing the structure is proportional to the number of nodes and the per-node structural description cost is l.

Our model is described by the union structure $\mathcal{T} = \{\mathcal{N}, \mathcal{O}, \Theta\}$. The model consists of a set of nodes $\mathcal{N}$, an order relation $\mathcal{O}$ and a set of node probabilities $\Theta = \{\theta^i, i \in \mathcal{N}\}$, where θ^i is the probability for the node n in the union-tree indexed c. To describe or encode the fit of the model to the data, for each node in the model, we need to describe or encode whether or not the node was present in the sample.

Hence, given a model $\mathcal{T}$ consisting of d nodes, the descriptor length for the model, conditional on the set of correspondences is $\mathcal{C}$ is:

$$\mathrm{LL}(\mathcal{D}|\mathcal{T}, \mathcal{C}) = \sum_{i=1}^{d} [-\mathcal{L}(\mathcal{D}|\mathcal{T}, \mathcal{C}) + l]. \tag{5}$$

Our aim is to optimize the descriptor length 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 $\mathcal{N}$. A variation in the actual identity does not change the log-likelihood, 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. This can be done because, as we will see, nodes that are not mapped to do not affect the optimization process. 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 minimize the descriptor length with respect to the remaining two sets of variables. The length of the description is minimized by any order relation $\mathcal{O}$ that is consistent with all the hierarchies for the sample trees (if any exists). Let $p_i(\mathcal{C})$ 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

$$\theta^i = \frac{p_i(\mathcal{C})}{m}.$$

When the optimal sampling probabilities are substituted into the descriptor length, we have that

$$\hat{L}L(\mathcal{D}|\mathcal{C}) = m \sum_{i \in \mathcal{N}} \left[\frac{p_i(\mathcal{C})}{m} \log\left(\frac{p_i(\mathcal{C})}{m}\right) + \left(1 - \frac{p_i(\mathcal{C})}{m}\right) \log\left(1 - \frac{p_i(\mathcal{C})}{m}\right) \right] + nl = -m \sum_{i \in \mathcal{N}} I(\theta^i) + nl, \tag{6}$$

where $I(\theta^i) = -\left[\theta^i \log(\theta^i) + (1 - \theta^i)\log(1 - \theta^i)\right]$ is the entropy of the sampling distribution for node i, and d is the number of nodes in $\mathcal{T}$. 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 descriptor length takes on the value ∞.

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 appearing in (6). In our previous work [22]. 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 $\mathcal{D}$.

5 Computing the Model

Finding the global minimum of the descriptor length is an intractable combinatorial problem. Hence, we resort to a local search technique.

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.

Interestingly, 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 $\hat{\mathcal{C}}(v) = \hat{\mathcal{C}}(w) \Leftrightarrow w = \mathcal{M}(v)$.

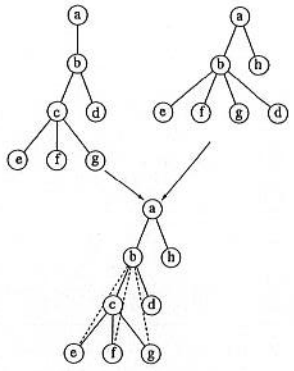


Fig. 1. The new model is obtained by merging matching nodes.

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 descriptor 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.

Let m_1 and m_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 $\theta v = \frac{p_v}{m_1+m_2}$ and $\theta v' = \frac{p'_v}{m_1+m_2}$ respectively, while the sampling probability of the merged node is $\theta vv' = \frac{p_v+p_{v'}}{m_1+m_2}$. Hence, the descriptor length advantage obtained by merging the nodes v and v' is:

$$\mathcal{A}(v, v') = (m_1 + m_2)\left[I(\theta v) + I(\theta v') - I(\theta vv')\right] + l. \tag{7}$$

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

$$\mathcal{A}(\mathcal{M}) = \sum_{(v,v')\in\mathcal{M}} \mathcal{A}(v, v') = \sum_{(v,v')\in\mathcal{M}} \left[(m_1 + m_2)\left[I(\theta v) + I(\theta v') - I(\theta vv')\right] + l\right]. \tag{8}$$

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 utility of the merged model. The costs that allowed the problem to be posed as an edit distance problem are $r_v = (m_1+m_2)I(\theta v)+l$ for the removal of node v, and $m_{vv'} = (m_1 + m_2)I(\theta vv') + l$ for matching node v with node v'.

With these edit costs, the utility in (2) assumes the same value of the advantage in descriptor length:

$$\mathcal{U}(\mathcal{M}) = \sum_{(u,v)\in\mathcal{M}} \left[(m_1 + m_2)(I(\theta u) + I(\theta v) - I(\theta uv)) + l\right]. \tag{9}$$

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.

We initialize our algorithm by calculating the sum of the descriptor length of using a per tree-sample in $\mathcal{D}$. The descriptor length is given by $l\sum_{t\in\mathcal{D}} \#\mathcal{N}^t$, where $\#\mathcal{N}^t$ is the number of nodes in the tree-sample t. For each pair of initial union model we calculate the merged structure and the advantage in descriptor length arising from merging the models. From the set of potential merges, we can identify the one which reduces the descriptor cost by the greatest amount. At this point we calculate the union and the advantage in descriptor cost obtained by merging the newly obtained model with each of the remaining model, and we iterate the algorithm until we merge every tree model.

6 Computing the Distance

To find the set correspondences that minimizes the edit distance between two trees we make use of two results presented in [21]. 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

$$\mathcal{M}^* = \operatorname*{argmax}_{\mathcal{M}\in O} \mathcal{U}(\mathcal{M}). \tag{10}$$

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 [21] for a detailed explanation of the approach.

7 Experimental Results

The aim of this session is to illustrate the behavior of the tree merging algorithm and the effects of the estimated distances. To meet this goal we have randomly generated a prototype tree model and, from this tree, we generated some tree-samples. The procedure for generating the random tree prototype was as follows:

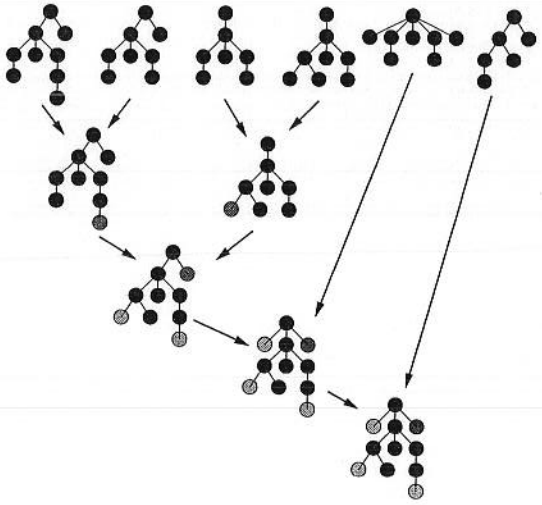


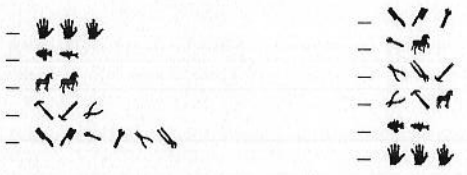
Fig. 2. Merging sample trees into a tree-model.

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 sampling probability of the nodes is assigned from a uniform distribution.

Figure 2 illustrates an example merge of 6 sample trees. The figure shows the structural archetype of the merged models after each stage. The color of the nodes in the tree models represents the sampling probability of that node: nodes sampled with higher probability are displayed in darker colors, while lighter colors are associated to nodes with low sampling probability.

In order to asses the quality of the extracted distance we compare clusters defined by performing pairwise clustering of the distance matrix [21,12]. We evaluate the approach on the problem of shock tree matching. We calculate the pairwise edit distance between shock-trees extracted from a database of 16 shapes.

Figure 3 shows the clusters using the two distance measures. The first column shows the clusters extracted fitting the generative model, while the second column displays the cluster extracted from the edit-distances with uniform cost. While there is some merge and leakage, the cluster extracted by fitting a generative tree model clearly outperform those obtained using uniform edit-cost. The second to last cluster extracted using the tree-model 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 information, correctly clusters the shock-trees together.



c) Mixture of tree models

b) Pairwise clustering from edit-distance

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

8 Conclusions

This paper presented a method for estimating of the cost of tree edit operations. The approach poses the problem as that of estimating a generative model for a set of tree samples. The generative model uses the tree-union as the structural archetype of trees in the distribution and assigns to each node in the archetype the probability that the node is present in a sample. A minimum descriptor length formulation is then used to estimate the structure and parameters of this tree model as well as the node-correspondences between trees in the sample-set and the tree model. Finally, we use the extracted correspondences and sampling probability to set the removal cost for each node.

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. H. Bunke and G. Allermann, Inexact graph matching for structural pattern recognition, *Pattern Recognition Letters*, Vol 1, pp. 245–253, 1983.
3. H. Bunke and A. Kandel, Mean and maximum common subgraph of two graphs, *Pattern Recognition Letters*, Vol. 21, pp. 163–168, 2000.
4. W. J. Christmas, J. V. Kittler, and M. Petrou, Structural matching in computer vision using probabilistic relaxation, *PAMI*, Vol. 17(8), pp. 749–764, 1995.
5. 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.
6. N. Friedman and D. Koller, Being Bayesian about Network Structure, *Machine Learning*, to appear, 2002
7. L. Getoor et al., Learning Probabilistic models of relational structure, in *8th Int. Conf. on Machine Learning*, 2001.
8. 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.
9. S. Ioffe and D. A. Forsyth, Human Tracking with Mixtures of Trees, *ICCV*, Vol. I, pp. 690–695, 2001.
10. 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.

11. T. Liu and D. Geiger, Approximate Tree Matching and Shape Similarity, *ICCV*, pp. 456–462, 1999.
12. B. Luo, et al., A probabilistic framework for graph clustering, in *CVPR*, pp. 912–919, 2001.
13. M. Meilă. *Learning with Mixtures of Trees.* PhD thesis, MIT, 1999.
14. R. Myers, R. C. Wilson, E. R. Hancock, Bayesian graph edit distance, *PAMI*, Vol. 22(6), p. 628–635, 2000.
15. J. Riassen, Stochastic complexity and modeling, *Annals of Statistics*, Vol. 14, pp. 1080–1100, 1986.
16. A. Robles-Kelly and E. R. Hancock, String edit distance, random walks and graph matching, in *Structural, Syntactic, and Statistical Pattern Recognition*, pp.104–112, LNCS 2396, 2002.
17. A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for 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. L. G. Shapiro and R. M. Haralick, Structural descriptions and inexact matching, *PAMI*, Vol. 3(5), pp. 504–519, 1981.
20. A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker, Indexing using a spectral encoding of topological structure, in *CVPR*, 1999.
21. A. Torsello and E. R. Hancock, Efficiently computing weighted tree edit distance using relaxation labeling, in *EMMCVPR*, LNCS 2134, pp. 438–453, 2001.
22. A. Torsello and E. R. Hancock, Matching and embedding through edit-union of trees, in *ECCV*, LNCS 2352, pp. 822–836, 2002.
23. A. Torsello and E. R. Hancock, Computing approximate tree edit-distance using relaxation labeling, *Pattern REcognition Letters*, Vol. 24(8), pp. 1089–1097
24. R. C. Wilson and E. R. Hancock, Structural matching by discrete relaxation, *PAMI*, 1997.
25. A. K. C. Wong and M. You, Entropy and distance of random graphs with application to structural pattern recognition, *PAMI*, Vol. 7(5), pp. 599–609, 1985.

Self-Organizing Graph Edit Distance

Michel Neuhaus and Horst Bunke

Department of Computer Science, University of Bern,
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{mneuhaus,bunke}@iam.unibe.ch

Abstract. This paper addresses the issue of learning graph edit distance cost functions for numerically labeled graphs from a corpus of sample graphs. We propose a system of self-organizing maps representing attribute distance spaces that encode edit operation costs. The self-organizing maps are iteratively adapted to minimize the edit distance of those graphs that are required to be similar. To demonstrate the learning effect, the distance model is applied to graphs representing line drawings and diatoms.

1 Introduction

Graphs are a powerful and universal concept to represent structured objects. Particularly in pattern recognition and related areas they have gained significant amounts of attention recently [1,2,3]. If both unknown objects and known prototypes of pattern classes are represented by graphs, then the problem of pattern classification turns into the problem of measuring the similarity (or, equivalently, the distance) of graphs. A number of graph similarity measures haven been proposed in the literature [4,5,6]. Among those measures, graph edit distance has become quite popular [7,8,9]. In graph edit distance, a set of graph edit operations are introduced. These edit operations are used to transform one of a pair of given graphs into the other. The edit distance of two graphs is defined as the length of the shortest sequence of edit operations required to transform one graph into the other. A more powerful graph distance model is obtained if a cost is assigned to each edit operation. The costs are to be defined in such a way that edit operations that are very likely to occur have a low cost, while the cost of infrequent edit operations is high. Using such a cost model, the edit distance now becomes the cost of the cheapest sequence of edit operations transforming one of the given graphs into the other. Graph edit distance is very powerful from the theoretical point of view. However, it has a severe practical limitation in that no automatic procedures are available to automatically infer edit operation costs based on a sample set of graphs. In all applications reported in the literature, the edit costs are derived manually, using knowledge of the problem domain and heuristics. Obviously, such a procedure is not feasible in complex tasks.

In the present paper we propose a procedure to infer the costs of a set of edit operations automatically from a sample set of graphs. We focus on a special class of graphs, which is characterized by the existence of node and/or edge labels, where each label is a vector from the n-dimensional space of real numbers.

E. Hancock and M. Vento (Eds.): IAPR Workshop GbRPR 2003, LNCS 2726, pp. 83–94, 2003.