

Attributed Tree Homomorphism Using Association Graphs

Massimo Bartoli Marcello Pelillo

Dipartimento di Informatica, Università Ca' Foscari di Venezia
Via Torino 155, 30172 Venezia Mestre, Italy

Kaleem Siddiqi

Center for Intelligent Machines, McGill University
3480 University Street, Montréal, PQ, H3A 2A7 Canada

Steven W. Zucker

Center for Computational Vision and Control, Yale University
P.O. Box 208285, New Haven CT 06520-8285

Abstract

The matching of hierarchical relational structures is of significant interest in computer vision and pattern recognition. We have recently introduced a new solution to this problem, based on a maximum clique formulation in a (derived) “association graph.” This allows us to exploit the full arsenal of clique finding algorithms developed in the algorithm community. However, thus far we have focussed on one-to-one correspondences (isomorphisms), which appears to be too strict a requirement for many vision problems. In this paper we provide a generalization of the association graph framework to handle many-to-one correspondences. We define a notion of an ϵ -homomorphism (a many-to-one mapping) between attributed trees, and provide a method of constructing a weighted association graph where maximal weight cliques are in one-to-one correspondence with maximal similarity subtree homomorphisms. We then solve the problem by using replicator dynamical systems from evolutionary game theory.

1. Introduction

The matching of relational structures is a classic problem in computer vision and pattern recognition, instances of which arise in areas as diverse as object recognition, motion and stereo analysis (see, e.g., [2]). A well-known approach to solving this problem consists of transforming it into the equivalent problem of finding a maximum clique in an auxiliary graph structure, known as the *association graph* [2]. The idea goes back to Ambler *et al.* [1], and has since been successfully employed in a variety of different problems. This framework is attractive because it casts relational structure matching as a pure graph-theoretic problem, for which a solid theory and powerful algorithms have

been developed. Although the maximum clique problem is known to be *NP*-hard, powerful heuristics exist which efficiently find good approximate solutions [3].

In many computer vision problems, relational structures are organized in a hierarchical manner, i.e., are *trees*. However, in standard association graph formulations, the solutions are not constrained to preserve this partial order. Hence, the extension of such techniques to tree matching problems is of considerable interest. We have recently introduced a solution to this problem by providing a novel way of deriving an association graph, based on the graph-theoretic notions of connectivity and the distance matrix [4]. We have proved that in this new formulation there is a one-to-one correspondence between maximum cliques in the derived association graph, and maximum subtree isomorphisms. The framework has also been extended to handle the matching of trees whose nodes have one or more associated attributes, by casting the attributed tree matching problem as an equivalent problem of finding a maximum weight clique in a “weighted” association graph.

Whereas thus far we have focussed on one-to-one relations (isomorphisms) between the structures being matched, it is clear that this is too strict a requirement for problems where structural alterations may occur in the underlying trees, i.e., nodes are deleted or additional nodes are present. In this paper we provide a generalization of the association graph framework to handle many-to-one correspondences. We define a notion of an ϵ -homomorphism (a many-to-one mapping) between weighted attributed trees, and provide a method of constructing an association graph where maximal weight cliques are in one-to-one correspondence with maximal similarity subtree homomorphisms. We then solve the problem by using *replicator* dynamical systems from evolutionary game theory. We illustrate the approach by matching articulated and deformed shapes described by shock trees.

2. Tree homomorphisms

Formally, an *attributed tree* is a triple $T = (V, E, \alpha)$, where (V, E) is the “underlying” rooted tree and $\alpha : V \rightarrow \mathcal{A}$ is a function which assigns an attribute vector $\alpha(u)$ to each node $u \in V$. Two nodes $u, v \in V$ are said to be *adjacent* (denoted $u \sim v$) if they are connected by an edge. We shall also consider a function $\delta : \mathcal{A} \rightarrow \mathbb{R}_+^*$ which assigns to each set of attributes (and therefore to each node in the tree) a real positive number. This will be interpreted as the negligibility of the corresponding node in the tree. Specifically, a node will be declared “negligible” if the value of the function δ corresponding to its attributes is smaller than a fixed threshold ϵ . This will allow us to associate a *cluster* of nodes (defined in the following) in the first subtree to a single node in the other one, thereby defining a many-to-one mapping from the first to the second tree.

For technical simplicity, we shall transform our node-weighted tree into an edge-weighted *derived attributed tree*, by simply moving the δ -value associated to every node to the edge connecting it to its parent. The root of the original tree will become a child of a newly created dummy root, and its weight will be moved to the corresponding edge. In the derived tree, we shall speak of a “negligible” edge, when its weight is less than threshold ϵ .

Given a fixed threshold $\epsilon \geq 0$, we define the distance $d_\epsilon(u, v)$ between two nodes u and v in an attributed tree, as the number of non-negligible edges (i.e. with weight less than ϵ) on the (unique) path from u to v . We define the function $\text{lev}_\epsilon(u)$ of a node in an attributed tree, as the distance d_ϵ from the root of the tree and node u .

Given $\epsilon \geq 0$, we define an ϵ -*cluster* in a derived attributed tree T as a subset of nodes of T such that for every two nodes u and v in it, we have:

$$d_\epsilon(u, v) = 0 .$$

A set with only one node is a particular case of cluster that we call a *singleton*. It can easily be proved that given an ϵ -cluster C , for all $u, v \in C$, and $z \notin C$, we have:

$$d_\epsilon(u, z) = d_\epsilon(v, z) .$$

From this observation, the next proposition follows [5]:

Proposition 1 *Given two ϵ -clusters C' and C'' , for any pair of nodes, one in C' and the other in C'' , their ϵ -distance is constant.*

This allows us to extend the notion of ϵ -distance to pairs of clusters, which we shall denote by $d_\epsilon(C', C'')$, and in turn to generalize the notion of adjacency to clusters. Specifically, two disjoint ϵ -clusters C' and C'' in a derived

attributed tree are said to be ϵ -*adjacent* (denoted $C' \sim_\epsilon C''$) if:

$$d_\epsilon(C', C'') = 1 .$$

It is clear that when C' and C'' are singletons, this is equivalent to the traditional notion of adjacency between nodes.

Finally, we are in a position to introduce the notion of a parent-child relationship between pairs of clusters. Let C' and C'' be two disjoint ϵ -clusters in a derived attributed tree T . We say that C' is the ϵ -*parent* of C'' when:

$$C' \sim_\epsilon C''$$

and

$$\text{lev}_\epsilon(C') \leq \text{lev}_\epsilon(C'')$$

where $\text{lev}_\epsilon(C)$ is defined as $d_\epsilon(\text{root}(T), C)$.

Definition 1 *Let $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$ be two attributed trees. Any onto function $\gamma : H_1 \rightarrow H_2$, with $H_1 \subseteq V_1$ and $H_2 \subseteq V_2$, is called a subtree ϵ -homomorphism if the subgraph induced by H_2 is connected (i.e. is a tree) and if for any $u, v \in H_2$ it satisfies the following properties:*

$$\gamma^{-1}(u) \text{ is an } \epsilon\text{-cluster} \quad (1)$$

$$u \text{ is the parent of } v \Leftrightarrow \gamma^{-1}(u) \text{ is the } \epsilon\text{-parent of } \gamma^{-1}(v) . \quad (2)$$

Clearly, in realistic applications, it would be desirable to find a homomorphism which pairs nodes having “similar” attributes. To this end, let σ be any similarity measure on the attribute space, i.e., any (symmetric) function which assigns a positive number to any pair of attribute vectors. If $\gamma : H_1 \rightarrow H_2$ is a subtree ϵ -homomorphism between two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$, the overall similarity between the matched structures can be defined as follows:

$$S(\gamma) = \sum_{u \in H_2} \sum_{w \in \gamma^{-1}(u)} \sigma(\alpha_1(w), \alpha_2(u))$$

The ϵ -homomorphism γ is called a *maximal similarity subtree ϵ -homomorphism* if there is no other subtree ϵ -homomorphism $\gamma' : H'_1 \rightarrow H'_2$ such that H_1 is a strict subset of H'_1 and $S(\gamma) < S(\gamma')$. It is called a *maximum similarity subtree ϵ -homomorphism* if $S(\gamma)$ is largest among all subtree ϵ -homomorphisms between T_1 and T_2 .

Homomorphisms are related to isomorphisms in the following intuitive way. Let $\gamma : H_1 \rightarrow H_2$ be an ϵ -homomorphism between two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$. We construct a graph $T'_1 = (V'_1, E'_1)$ defined as follows:

$$V'_1 = \{C \subseteq V_1 \mid \exists u \in H_2 \text{ such that } C = \gamma^{-1}(u)\}$$

$$E'_1 = \{(C_1, C_2) \in V'_1 \times V'_1 \mid C_1 \sim_\epsilon C_2\}$$

Then, there exists a subtree of T_2 , say T'_2 , such that T'_1 and T'_2 are isomorphic.

3. The tree association graph

Let u and v be two nodes of an attributed tree T , joined by path $u = x_0 x_1 \dots x_n = v$, and let x^* be the node in the path nearest to the root of T ; formally: $x^* = \operatorname{argmin}_{x \in \{x_0, \dots, x_n\}} \operatorname{lev}(x)$. We define:

$$\begin{aligned} n(u, v) &= d(u, x^*) \\ p(u, v) &= d(x^*, v) \\ n_\epsilon(u, v) &= d_\epsilon(u, x^*) \\ p_\epsilon(u, v) &= d_\epsilon(x^*, v). \end{aligned}$$

Clearly, $n_\epsilon(u, v) \leq n(u, v)$ and $p_\epsilon(u, v) \leq p(u, v)$.

Definition 2 The weighted ϵ -tree association graph (ϵ -TAG) of two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$ is the graph $G_\epsilon = (V, E, \omega)$ where

$$V = V_1 \times V_2$$

such that for any two nodes (w, u) and (z, v) in V

$$(w, u) \sim (z, v) \Leftrightarrow \begin{cases} n_\epsilon(w, z) = n(u, v) \\ p_\epsilon(w, z) = p(u, v) \end{cases} \quad (3)$$

and ω is a function which assigns a positive weight to each node $(u, v) \in V$ as follows:

$$\omega(u, v) = \sigma(\alpha_1(u), \alpha_2(v)). \quad (4)$$

Intuitively, condition (3) imposes that the number of negligible ascent edges between w and z must be equal to the number of ascent edges between u and v . The same applies to descent edges. Notice that, when $\epsilon = 0$ we have $n_\epsilon(w, z) = n(w, z)$ and $p_\epsilon(w, z) = p(w, z)$, and therefore we obtain the same association graph structure as originally studied for the tree isomorphism case [4].

Given a subset of nodes C of V , the total weight assigned to C is simply the sum of all the weights associated with its nodes. A *maximal weight clique* in G is one which is not contained in any other clique having larger total weight, while a *maximum weight clique* is a clique having largest total weight. The maximum weight clique problem is to find a maximum weight clique of G [3].

The following result establishes a one-to-one correspondence between the attributed tree homomorphism problem and the maximum weight clique problem (see [5] for the proof).

Theorem 1 Any maximal (maximum) similarity subtree ϵ -homomorphism between two attributed trees induces a maximal (maximum) weight clique in the corresponding weighted ϵ -TAG, and vice versa.

Once the tree homomorphism problem has been formulated as a maximum weight clique problem, any clique finding algorithm can be employed to solve it (see [3] for a recent review). In the work reported in this paper, we used an approach recently introduced in [4, 6], which consists of transforming the maximum weight clique problem to an equivalent quadratic program. We then used a class of dynamical systems from evolutionary game theory called *replicator equations* to find approximate solutions of it. Due to space limitations, we refer the reader to [4, 6], and [7].

4. An example: Matching shock trees

We illustrate our framework with numerical examples of shape matching. We use a *shock graph* representation based on a coloring of the shocks (singularities) of a curve evolution process acting on simple closed curves in the plane [8]. Shocks are grouped into distinct types according to the local variation of the radius function along the medial axis. Intuitively, the radius function varies monotonically at a type 1, reaches a strict local minimum at a type 2, is constant at a type 3 and reaches a strict local maximum at a type 4. The shocks comprise vertices in the graph, and their formation times direct edges to form a basis for tree matching. Each graph can be reduced to a unique attributed rooted tree, providing the requisite hierarchical structure for our matching algorithm. The vector of attributes assigned to each node $u \in V$ of the attributed shock tree $T = (V, E, \alpha)$ is given by $\alpha(u) = (x_1, y_1, r_1, v_1, \theta_1; \dots; x_m, y_m, r_m, v_m, \theta_m)$. Here m is the number of shocks in the group, and $x_i, y_i, r_i, v_i, \theta_i$ are, respectively, the x coordinate, the y coordinate, the radius (or time of formation), the speed, and the direction of each shock i in the sequence, obtained as outputs of the shock detection process.

In order to apply our framework we must define a similarity measure between the attributes of two nodes u and v . The similarity measure used here is the same as the one described in [4]. The measure provides a number between 0 and 1, which represents the overall similarity between the geometric attributes of the two nodes being compared. The measure is designed to be invariant under rotations and translations of two shapes, and to satisfy the requirements of the weight function discussed above.

We selected 21 silhouettes representing seven different object classes; the tools shapes were taken from the Rutgers Tools database. For each object class a prototype object was chosen and matched against *all* entries in the database using the clique-finding replicator equations described in [4].

The selection of a proper value for the parameter ϵ is clearly crucial for the performance of the matching process.

Our intuition is that the larger the value of ϵ , the larger the cluster formed by the matching process. In the experiments presented here we set $\epsilon = 0.0026$, a value which we found to be near-optimal for the trees at hand.

We ranked the results using a score given by the quantity:

$$W \times \frac{1}{2} \left\{ \sum_{u \in H_1} (m(u)/M_1) + \sum_{v \in H_2} (m(v)/M_2) \right\}, \quad (5)$$

where H_1 and H_2 are the sets of matched nodes in the first and in the second tree, respectively, W is the overall similarity between matched nodes (i.e., the weight of the maximal clique found), M_1 and M_2 the total mass associated with the nodes of each tree, and $m(u), m(v)$ the masses of nodes u and v , respectively. The score represents the weight of the maximal clique scaled by the average of the total (relative) mass of nodes in each tree that participates in the match. The top 8 matches are shown for each query shape, in Table 1. It is evident that the best matches are to instances from the same object class. These results represent an improvement with respect to those obtained for the isomorphism case [4], which corresponds to having $\epsilon = 0$, since nodes with negligible significance can now be clustered. Specifically, we have observed that our many-to-one matching algorithm tends to provide a sharper distinction between object classes.

5. Conclusions

We have expanded our earlier work on matching hierarchical relational structures to handle many-to-one mappings. This has been done by introducing a notion of ϵ -homomorphism between attributed trees, and by constructing an association graph whose maximal weight cliques are shown to be in one-to-one correspondence with maximal similarity subtree homomorphisms. Computational examples of matching articulated and deformed shapes give improved results over our earlier work [4], because the framework now allows negligible nodes, which arise due to structural alterations in the underlying shapes, to be ignored during the matching process. More experiments and results can be found in [5].

References

[1] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone. A Versatile Computer-controlled Assembly System. In *Proc. 3rd Int. J. Conf. Artif. Intell.*, pages 298–307, Stanford, CA, 1973.

[2] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, N.J, 1982.

Prot.	Top 8 matches							
	1	2	3	4	5	6	7	8
	 8.00	 5.89	 5.33	 4.73	 4.73	 4.67	 4.66	 4.26
	 9.00	 6.78	 6.76	 4.99	 4.68	 3.89	 3.68	 3.67
	 12.00	 8.28	 8.20	 7.68	 5.59	 5.53	 5.47	 4.94
	 12.00	 8.96	 5.15	 4.61	 4.53	 4.52	 4.47	 4.28
	 8.00	 5.57	 4.68	 4.31	 4.21	 3.00	 2.78	 2.74
	 21.00	 19.23	 18.50	 7.37	 6.20	 6.20	 6.10	 5.77
	 12.00	 9.67	 5.92	 5.35	 5.28	 5.28	 4.70	 3.02

Table 1. A tabulation of the top 8 matches for each prototype shape, with $\epsilon = 0.0026$. The scores indicate the value of index (5).

[3] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The Maximum Clique Problem. In *Handbook of Combinatorial Optimization (Supplement Volume A)*, D.-Z. Du and P. M. Pardalos (Eds.), Kluwer Academic Publishers, Boston, MA, 1999.

[4] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching Hierarchical Structures Using Association Graphs. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(11), 1999, pages 1105–1120.

[5] M. Bartoli, M. Pelillo, K. Siddiqi, and S. W. Zucker. Attributed Tree Homomorphism Using Association Graphs. *In preparation*.

[6] I. M. Bomze, M. Pelillo, and V. Stix. Approximating the Maximum Weight Clique Using Replicator Dynamics, *Technical Report CS-99-13, Dipartimento di Informatica, Università Ca' Foscari di Venezia*, 1999.

[7] M. Pelillo. Replicator dynamics in combinatorial optimization, In *Encyclopedia of Optimization*, C. Floudas and P. M. Pardalos (Eds.). Kluwer Academic Publishers, Boston, MA (in press).

[8] K. Siddiqi, A. Shokoufandeh, S.J. Dickinson, and S.W. Zucker. Shock Graphs and Shape Matching, *Int'l J. Computer Vision*, 35(1), 1999, pages 13–32.