LEARNING SHAPE CATEGORIES BY CLUSTERING SHOCK TREES

Bin Luo, A. Robles-Kelly[∗], *A. Torsello, R. C. Wilson and E. R. Hancock*

The University of York, Department of Computer Science
York, YO1 5DD, UK

ABSTRACT

This paper investigates whether meaningful shape categories can be identified in an unsupervised way by clustering shock-trees. We commence by computing weighted and unweighted edit distances between shock-trees extracted from the Hamilton-Jacobi skeleton of 2D binary shapes. Next we use an EM-like algorithm to locate pairwise clusters in the pattern of edit-distances. We show that when the tree edit distance is weighted using the geometry of the skeleton, then the clustering method returns meaningful shape categories.

1. INTRODUCTION

There has recently been considerable interest in the use of the reaction-diffusion equation as a means of representing and analysing both 2D and 3D shapes [1, 2, 3]. In a nutshell, the idea is to extract a skeletal representation by evolving the shape-boundary inwards until singularities appear. Through the analysis of the differential properities of the singularities, a structural abstraction of the skeleton known as the shock-graph may be extracted. Although this abstraction has been widely used for shape-matching and recognition [4, 2], its use as a means of learning shape categories has attracted less attention. The aim in this paper is to investigate whether graph-clustering can be used as a means of partitioning shock-trees into shape classes via unsupervised learning. Graph clustering is an important yet relatively under-researched topic in machine learning [5]. The process can be used to structure large data-bases of relational models [6] or to learn equivalence classes. One of the reasons for limited progress in the area has been the lack of algorithms suitable for clustering relational structures. In particular, the problem has proved elusive to conventional central clustering techniques. The reason for this is that it has proved difficult to define what is meant by the mean or representative graph for each cluster. A more fruitful avenue of investigation may be to pose the problem as pairwise clustering. This requires only that a set of pairwise distances between graphs be supplied. The clusters are located by identifying sets of graphs that have strong mutual pairwise affinities. There is therefore no need to explicitly identify an representative (mean, mode or median) graph for each cluster.

[∗] Supported by CONACYT, under grant No. 146475/151752.

Our approach is as follows. Commencing from a database of silhouettes, we extract the Hamilton-Jacobi skeleton and locate the shocks which correspond to singularities in the evolution of the object boundary under the eikonal equation. We compute the similarity of the shapes using weighted and un-weighted tree edit distance. With the set of pairwise edit-distances between the shock-graphs to hand, we use a maximum-likelihood method for pairwise clustering. Our experiments show that when used in conjunction with the weighted tree edit distance, the pairwise clustering process locates meaningful shape categories.

2. SHOCK TREE EDIT DISTANCE

The practical problem tackled in this paper is the clustering of 2D binary shapes based on the similarity of their shock-trees. The idea of characterizing boundary shape using the differential singularities of the reaction equation was first introduced into the computer vision literature by Kimia, Tannenbaum and Zucker [3]. The idea is to evolve the boundary of an object to a canonical skeletal form using the reaction-diffusion equation. The skeleton represents the singularities in the curve evolution, where inward moving boundaries collide. The reaction component of the boundary motion corresponds to morphological erosion of the boundary, while the diffusion component introduces curvature dependent boundary smoothing. In practice, the skeleton can be computed in a number of ways, here we use a variant of the method Siddiqi, Tannenbaum and Zucker, which solves the eikonal equation which underpins the reaction-diffusion analysis using the Hamilton-Jacobi formalism of classical mechanics [1]. Once the skeleton is to hand, the next step is to devise ways of using it to characterize the shape of the original boundary. Here we follow Zucker, Siddiqi, and others, by labeling points on the skeleton using so-called shock-classes [2]. According to this taxonomy of local differential structure, there are different classes associated with behavior of the radius of the osculating circle from the skeleton to the nearest pair of boundary points. The so-called shocks distinguish between the cases where the local osculating circle has maximum radius, minimum radius, con-

	0.981	0.844	0.434	0.604	0.562	0.600	0.497	0.554	0.422	0.511	0.557	0.484	0.427	0.428	0.415	0.402
	0.844	0.981	0.548	0.685	0.683	0.720	0.559	0.509	0.381	0.621	0.629	0.534	0.447	0.474	0.517	0.434
	0.446	0.543	1.000	0.569	0.554	0.643	0.637	0.475	0.651	0.436	0.559	0.428	0.395	0.406	0.414	0.373
	0.604	0.685	0.809	1.000	0.663	0.731	0.713	0.592	0.496	0.507	0.624	0.404	0.418	0.443	0.449	0.404
	0.562	0.684	0.531	0.666	1.000	0.857	0.714	0.459	0.385	0.588	0.622	0.456	0.497	0.524	0.541	0.521
	0.571	0.650	0.643	0.750	0.857	1.000	0.793	0.584	0.505	0.580	0.670	0.456	0.515	0.514	0.480	0.503
	0.502	0.559	0.648	0.727	0.703	0.796	1.000	0.533	0.503	0.438	0.562	0.433	0.457	0.467	0.506	0.482
	0.554	0.606	0.475	0.592	0.459	0.531	0.554	1.000	0.415	0.434	0.405	0.443	0.459	0.419	0.400	0.408
	0.441	0.356	0.676	0.506	0.388	0.472	0.530	0.394	0.981	0.384	0.438	0.396	0.386	0.310	0.353	0.310
	0.516	0.626	0.479	0.507	0.593	0.586	0.487	0.434	0.379	1.000	0.825	0.556	0.520	0.634	0.630	0.555
	0.556	0.636	0.559	0.626	0.622	0.670	0.578	0.405	0.428	0.820	1.000	0.449	0.590	0.542	0.532	0.518
	0.496	0.449	0.417	0.388	0.462	0.429	0.475	0.397	0.403	0.492	0.449	1.000	0.627	0.496	0.519	0.560
	0.395	0.447	0.395	0.441	0.495	0.488	0.445	0.497	0.371	0.602	0.573	0.693	0.992	0.711	0.687	0.699
	0.443	0.520	0.334	0.436	0.570	0.536	0.465	0.422	0.335	0.632	0.554	0.494	0.696	0.976	0.895	0.840
	0.431	0.450	0.409	0.425	0.520	0.520	0.480	0.445	0.378	0.629	0.543	0.572	0.719	0.847	0.982	0.771
	0.441	0.447	0.380	0.447	0.560	0.520	0.493	0.434	0.346	0.563	0.558	0.541	0.765	0.864	0.817	1.000

Fig. 1. Pairwise edit distances computed using un-weighted trees.

	1.000	1.000	0.774	1.000	0.889	0.889	0.706	0.643	0.850	0.818	0.889	0.635	0.640	0.706	0.694	0.684
	1.000	1.000	0.774	1.000	0.889	0.889	0.706	0.643	0.850	0.818	0.889	0.635	0.640	0.706	0.694	0.684
	0.774	0.774	1.000	0.774	0.833	0.833	0.676	0.714	0.800	0.773	0.694	0.615	0.620	0.676	0.667	0.658
	1.000	1.000	0.774	1.000	0.889	0.889	0.706	0.643	0.850	0.818	0.889	0.635	0.640	0.706	0.694	0.684
	0.889	0.889	0.694	0.889	1.000	1.000	0.765	0.730	0.950	0.808	0.778	0.673	0.680	0.765	0.750	0.737
	0.889	0.889	0.833	0.889	1.000	1.000	0.765	0.730	0.950	0.808	0.778	0.673	0.680	0.765	0.750	0.737
	0.706	0.706	0.676	0.706	0.765	0.765	1.000	0.782	0.794	0.674	0.680	0.730	0.692	0.765	0.801	0.669
	0.643	0.643	0.714	0.643	0.730	0.730	0.847	1.000	0.771	0.649	0.639	0.769	0.724	0.651	0.698	0.682
	0.850	0.850	0.800	0.850	0.950	0.950	0.794	0.857	1.000	0.764	0.739	0.692	0.700	0.794	0.778	0.763
	0.818	0.818	0.773	0.818	0.808	0.808	0.599	0.731	0.764	1.000	0.909	0.647	0.720	0.749	0.806	0.789
	0.889	0.889	0.694	0.889	0.778	0.778	0.680	0.548	0.739	0.909	0.778	0.598	0.680	0.595	0.667	0.737
	0.635	0.635	0.615	0.635	0.673	0.673	0.778	0.769	0.692	0.647	0.598	1.000	0.785	0.730	0.752	0.729
	0.640	0.640	0.620	0.640	0.680	0.680	0.741	0.724	0.700	0.655	0.680	0.824	1.000	0.840	0.764	0.834
	0.706	0.706	0.676	0.706	0.765	0.765	0.706	0.651	0.715	0.824	0.765	0.730	0.840	1.000	0.972	0.947
	0.694	0.694	0.667	0.694	0.750	0.750	0.801	0.698	0.778	0.659	0.583	0.752	0.812	0.915	1.000	0.920
	0.684	0.684	0.658	0.684	0.737	0.737	0.669	0.682	0.763	0.718	0.573	0.729	0.741	0.947	0.920	0.947

Fig. 2. Pairwise edit distances computed using weighted trees.

stant radius or a radius which is strictly increasing or decreasing. We abstract the skeletons as trees in which the level in the tree is determined by their time of formation [4, 2]. The later the time of formation, and hence their proximity to the center of the shape, the higher the shock in the hierarchy. While this temporal notion of relevance can work well with isolated shocks (maxima and minima of the radius function), it fails on monotonically increasing or decreasing shock groups. To give an example, a protrusion that ends on a vertex will always have the earliest time of creation, regardless of its relative relevance to the shape.

To overcome this drawback, we augment the structural information given by the skeleton topology and the relative time of shock formation, with a measure of feature importance. We opt to use a shape-measure based on the rate of change of boundary length with distance along the skeleton. To compute the measure we construct the osculating circle to the two nearest boundary points at each location on the skeleton.

This measurement has previously been used in the literature to express *relevance* of a branch when extracting or pruning the skeleton, but is has recently been shown that its geometric and differential properties make it a good measure of shape similarity [7].

Given this representation we can cast the problem of computing distances between different shapes as that of finding the tree edit distance between the weighted graphs for their skeletons.

Tree edit distance is a generalization to trees of *String edit distance*. The edit distance is based on the existence of a set B of basic edit operation on a tree and a set C of costs, where $c_b \in C$ is the cost of performing the edit operation $b \in B$. The choice of the basic edit operations, as well as their cost, can be tailored to the problem, but common operations include leaf pruning, path merging, and, in case of an attributed tree, change of attribute. Given two trees T_1 and T_2, the set B of basic edit operations, and the cost of such operation $C = c_b, b \in N$, we call an *edit path* from T_1 to T_2 a sequence $b_1, \ldots, b_n$ of basic edit operations that transform T_1 into T_2. The length of such path is $l = c_{b_1} + \cdots + c_{b_n}$; the *minimum length edit path* from T_1 to T_2 is the path form T_1 to T_2 with minimum length. The length of the minimum length path is the tree edit distance.

With our measure assigned to each edge of the tree, we define the cost of matching two edges as the difference of the total length ratio measure along the branches. The cost of eliminating an edge is equivalent to the cost of matching it to an edge with zero weight, i.e. one along which the total length ratio is zero.

Using the edit distance of the shock trees we generate two similarity measures for a pair of shapes.

- The first measure is obtained weighting the nodes with the border length ratio normalized by the total length of the border of the shape. That is the length of the fraction of the border spanned by the shock group divided by the total length of the border. In this way the sum of the weights in a tree is 1 and the measure is scale invariant. The similarity of the shapes is computed by adding the minimum weight for each matched node, that is $d_{w,w'} = \sum_i min(w_i, w_i')$, where w_i and w_i' are the weight of the nodes that are matched together by our tree edit distance algorithm.

- The second measure of shape similarity is computed from the unweighted structure: We assign a uniform edit cost of 1 to each node and we compute the average ratio of matched nodes: $d_{w,w'} = \frac{1}{2}\left(\frac{\#\hat{T}}{\#T_1} + \frac{\#\hat{T}}{\#T_2}\right)$, where T_1 and T_2 are the two trees to be matched, $\hat{T}$ is the median of the two trees obtained through cut operations only, and $\#$ indicates the number of nodes in the tree.

Figures 1 and 2 show the un-weighted and weighted shock-tree edit distances for a set of 2D shapes.

3. GRAPH-CLUSTERING

We pose the problem of learning the set of shape-classes as that of finding pairwise clusters in the distribution of tree-edit distance. The process of pairwise clustering is somewhat different to the more familiar one of central clustering. Whereas central clustering aims to characterise cluster-membership using the cluster mean and variance, in pairwise clustering it is the relational similarity of pairs of objects which are used to establish cluster membership. Although less well studied than central clustering, there has recently been renewed interest in pairwise clustering aimed at placing the method on a more principled footing using techniques such as mean-field annealing [8].

To commence, we require some formalism. We are interested in grouping a set of graphs $\mathcal{G} = \{G_1, ..., G_{|M|}\}$ whose index set is M. The set of graphs is characterised using a matrix of pairwise similarity weights. The elements of this weight matrix are computed using tree-edit distance $d_{i,j}$ between the graphs indexed i and j. Here we use the exponential similarity function

$$W_{i,j} = \begin{cases} \exp[-kd_{i,j}] & \text{if } i \neq j \\ 0 & otherwise \end{cases} \quad (1)$$

to generate the elements of the weight-matrix, where k is a constant which is heuristically set. The aim in graph-clustering is to locate the updated set of similarity weights which partition the set of graphs into disjoint subsets. To be more formal, suppose that Ω is the set of graph-clusters and let S_ω represent the set of the graphs belonging to the cluster indexed ω. Further, let $s_{i\omega}^{(n)}$ represent the probability that the graph indexed i belongs to the cluster indexed ω at iteration n of the algorithm. We are interested in posing the clustering problem in a maximum likelihood setting. Under the assumption that the cluster memberships of the graphs follow a Bernoulli distribution with the link-weights as parameters, the likelihood-function for the weight matrix W is given by

$$P(W) = \prod_{\omega \in \Omega} \prod_{(i,j)\in M\times M} W_{i,j}^{s_{i\omega} s_{j\omega}} (1 - W_{i,j})^{1 - s_{i\omega} s_{j\omega}} \quad (2)$$

The corresponding log-likelihood function is

$$\mathcal{L} = \sum_{\omega\in\Omega} \sum_{(i,j)\in M\times M} \left\{ s_{i\omega} s_{j\omega} \ln W_{ij} + (1 - s_{i\omega} s_{j\omega}) \ln(1 - W_{i,j}) \right\} \quad (3)$$

We have recently, shown how this log-likeihood function can be iteratively optmised using an EM-like process. In the E (expectation) step, the cluster membership probabilities are updated according to the formula

$$s_{i\omega}^{(n+1)} = \frac{\prod_{j\in M} \left\{ \frac{W_{i,j}^{(n)}}{1 - W_{ij}^{(n)}} \right\}^{s_{j\omega}^{(n)}}}{\sum_{i\in M} \prod_{j\in M} \left\{ \frac{W_{ij}^{(n)}}{1 - W_{ij}^{(n)}} \right\}^{s_{j\omega}^{(n)}}} \quad (4)$$

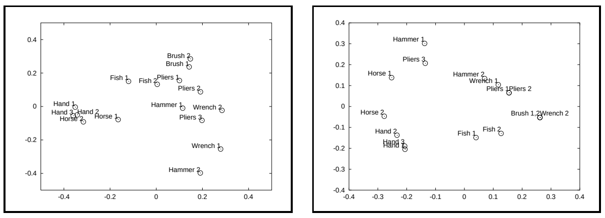
Once the revised cluster membership variables are to hand then we apply the M (maximisation) step of the algorithm to update the similarity-weight matrix. The updated similarity-weights are given by

$$W_{ij}^{(n+1)} = \sum_{\omega\in\Omega} s_{i\omega}^{(n)} s_{j\omega}^{(n)} \quad (5)$$

These two steps are interleaved and iterated to convergence.

4. EXPERIMENTS

The 16 shapes used in our study are shown in Figures 1 and 2. In Figure 1 we show the pattern of weighted edit distances between the shock-trees for the shapes, while Figure 2 shows the corresponding weighted tree edit distances.



(a) (b)

Fig. 3. (a)Multidimensional scaling applied to the weighted tree edit distances; (b)Multidimensional scaling applied to the weighted tree edit distances.

To visualise the data, we have applied multi-dimensional scaling to the two sets of distances [9]. Figures 3a and 3b respectively show the leading two components of the eigenvectors of the distance matrix for the weighted and unweighted trees. In the case of the weighted shock trees, the cluster structure is somewhat clearer than for the unweighted shock trees. However, in both cases the cluster structure is far from clear. In other words, the clustering of the graphs is not a problem that could be solved by simply applying central clustering to the eigenvectors delivered by MDS.

In Figure 4a we show the initial matrix of pairwise similarity weights for the unweighted trees for the different shapes. Here the redder the entries, the stronger the similarity; the bluer the entries, the weaker the similarity. The order of the entries in the matrix is the same as the order of the shapes in Figures 1 and 2. After six iterations of the clustering algorithm the similarity weight matrix shown in Figure 4b is obtained. There are six clusters (brush (1) + brush (2) + wrench (4); spanner (3) + horse (13) ; pliers (5) + pliers (6) + hammer (9) ;pliers (7) +hammer (8) + horse (12); fish (10) + fish (12); hand (14) + hand (15) + hand (16). Clearly there is merging and leakage between the different shape categories. In Figures 5a and 5b we show the initial and final similarity matrices when weighted trees are used. The entries in the initial similarity matrix are better grouped than those obtained when the unweighted tree edit distance is used. There are now seven clusters. brush (1) + brush (2) ; spanner (3) + spanner (4); pliers (5) + pliars (6) + pliers (7); hammer (8) + hammer (9); fish (10) + fish (11); horse (12) + horse (13); hand (14) + hand (15) + hand (16)). These correspond exactly to the shape categories in the data-base.

5. CONCLUSIONS

This paper has presented a study of the problem of learning shape-categories by clustering shock-trees. We gauge the similarity of the trees using weighted and unweighted edit distance. To idetify distinct groups of trees, we use a maixmum likelihood algorithm for pairwise clustering. This takes as its input a matrix of pairwise similarities between shock-trees computed from the edit distances. The algorithm is reminiscent of the EM algorithm and has interleaved iterative steps for computing cluster-memberships and for updating the pairwise similarity matrix. Experimental evaluation of the method shows that it is capable of extracting clusters of trees which correspond closely to the shape-categories present.

6. REFERENCES

[1] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker, "The hamilton-jacobi skeleton," in *Seventh International Conference on Computer Vision*. IEEE, Sept. 1999, pp. 828–834, IEEE Computer Society.

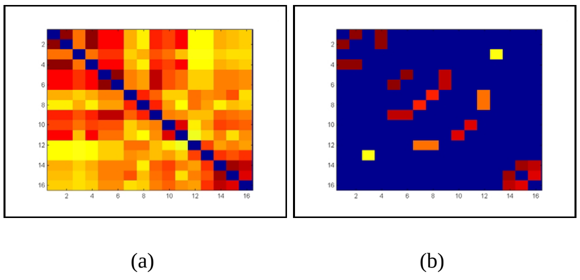


Fig. 4. (a) Initial similarity matrix for the unweighted tree edit distances; (b)Final similarity matrix for the unweighted tree edit distances.

[2] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, "Shock graphs and shape matching," *International Journal of Computer Vision*, vol. 35, no. 1, pp. 13–32, 1999.

[3] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Shapes, shocks, and deforamtions i," *International Journal of Computer Vision*, vol. 15, pp. 189–224, 1995.

[4] A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker, "Indexing using a spectral encoding of topological structure," in *Conference on Computer Vision and Pattern Recognition*, June 1999.

[5] S. Rizzi, "Genetic operators for hierarchical graph clustering," *Pattern Recognition Letters*, vol. 19, pp. 1293–1300, 1998.

[6] K. Sengupta and K. L. Boyer, "Organizing large structural modelbases," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, 1995.

[7] A. Torsello and E.R. Hancock, "A skeletal measure of 2d shape similarity," Submited, 2000.

[8] T. Hofmann and M. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Tansactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, 1997.

[9] C. Chatfield and A.J. Colins, *Introduction to multivariate analysis*, Chapman & Hall/CRC, 1980.

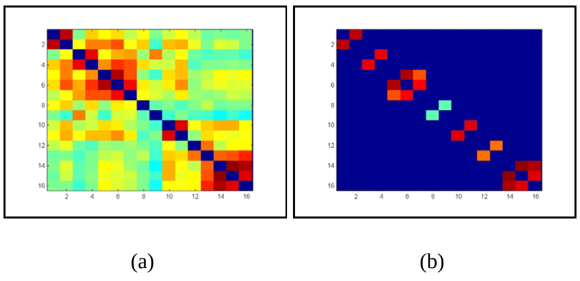


Fig. 5. (a) Initial similarity matrix for the weighted tree edit distances; (b) Final similarity matrix for the weighted tree edit distances.