

A Structured Learning Approach to Attributed Graph Embedding

Haifeng Zhao¹ Jun Zhou^{2,3} Antonio Robles-Kelly^{2,3}

¹School of Comp. Sci. & Tech., Nanjing Univ. of Sci. & Tech., Nanjing 210094, China

²RSISE, Bldg. 115, Australian National University, Canberra ACT 0200, Australia

³National ICT Australia (NICTA) *, Locked Bag 8001, Canberra ACT 2601, Australia

Abstract. In this paper, we describe the use of concepts from structural and statistical pattern recognition for recovering a mapping which can be viewed as an operator on the graph attribute-set. This mapping can be used to embed graphs into spaces where tasks such as categorisation and relational matching can be effected. We depart from concepts in graph theory to introduce mappings as operators over graph spaces. This treatment leads to the recovery of a mapping based upon the graph attributes which is related to the edge-space of the graphs under study. As a result, this mapping is a linear operator over the attribute set which is associated with the graph topology. Here, we employ an optimisation approach whose cost function is related to the target function used in discrete Markov Random Field approaches. Thus, the proposed method provides a link between concepts in graph theory, statistical inference and linear operators. We illustrate the utility of the recovered embedding for shape matching and categorisation on MPEG7 CE-Shape-1 dataset. We also compare our results to those yielded by alternatives.

1 Introduction

In the pattern analysis community, there has recently been renewed interests in the embedding methods motivated by graph theory. One of the best known of these is ISOMAP [1]. Related algorithms include locally linear embedding which is a variant of PCA that restricts the complexity of the input data using a nearest neighbor graph [2], and the Laplacian eigenmap that constructs an adjacency weight matrix for the data-points and projects the data onto the principal eigenvectors of the associated Laplacian matrix [3]. Collectively, these methods are sometimes referred to as manifold learning theory.

Embedding methods can also be used to transform the relational-matching problem into a point-pattern matching problem in a high-dimensional space. The idea is to find matches between pairs of point sets when there are noises, geometric distortion and structural corruption. This problem arises in shape analysis, motion analysis and stereo reconstruction. The main challenge in graph matching is how to deal with differences in node and edge structure. One of the most elegant approaches to the graph matching problem has been to use graph spectral methods [4], and exploit information conveyed

* National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

by the eigenvalues and eigenvectors of the adjacency matrix. More recently, Sebastian and Kimia [5] have used a distance metric analogous to the string edit distance to perform object recognition from a dataset of shock graphs.

The main argument levelled against the methods mentioned above is that they adopt a heuristic approach to the relational matching problem by using a goal-directed graph similarity measure. To overcome this problem, several authors have proposed more general approaches using ideas from information and probability theory. For instance, Wong and You [6] defined an entropic graph-distance for structural graph matching. Christmas, Kittler and Petrou [7] have shown how a relaxation labeling approach can be employed to perform matching using pairwise attributes whose distribution is modeled by a Gaussian. Wilson and Hancock [8] have used a MAP (maximum *a posteriori*) estimation framework to accomplish purely structural graph matching. Recently, Caetano *et al.* have proposed a method to estimate the compatibility functions for purposes of learning graph matching [9].

In this paper, we aim at estimating a linear mapping so as to embed a graph into a high-dimensional space where distances between nodes correspond to the structural differences between graphs. This can be viewed as a statistical learning process in which the goal of computation is the recovery of a linear operator which maps the attribute-set of a graph onto an embedding space in order to minimise a cost function arising from a Markovian formulation. In this manner, the recovered mapping is related to the space defined by the graph edge-set while being an operator on the attribute-set. Such an embedding permits the use of metrics in the target space for relational matching and shape categorisation tasks.

Thus, the motivation here is to recover a statistically optimal solution for the graph embedding problem. The bulk of the work elsewhere in the literature hinges in the use of dimensionality reduction techniques or relational similarity and matching algorithms. Here we take a more general view of the problem through learning. This learning approach leads to the statistical methods, where, for Graphical Models, MRFs are the ideal choice due to their use of pairwise potentials. Moreover, the method provides a link between structural and statistical pattern recognition techniques through the algebraic graph theory [10], graph spectral methods [4] and Markov Random Fields [11].

2 Graph Theory and Statistical Learning

Here we work with a data set Γ of attributed graphs. As mentioned earlier, we aim at learning a linear mapping \mathcal{T} that can be used to embed the attributes of the graph-vertices into a space of dimensionality Ω whose basis is the optimal transformation of a linear map from the vertex to the edge space. In this manner, the embedding will reflect the structure of the edge-space of the graph while being based upon its attribute-set. This has two main advantages. Firstly, the target space for the learned mapping will reflect the structure of the graphs under study. Since similar graphs should have akin edge-spaces, this provides an embedding that is inherently related to a graph topology common to the set Γ . Secondly, note that the mapping \mathcal{T} embeds the vertex-attributes into the graph edge-space according to a linear operator drawn from spectral geometry.

This is not only practically useful but theoretically important since it provides a link between the spectra of graphs and linear operators.

2.1 Structured Learning

To commence, we require some formalism. Let $G_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{A}_i)$ denote the graph indexed i in Γ with node-set $\mathcal{V}_i = \{V_{i,1}, \dots, V_{i,|\mathcal{V}_i|}\}$, edge-set $\mathcal{E}_i = \{e | V_{i,a}, V_{i,c} \in \mathcal{V}_i\}$ and attribute-set $\mathcal{A}_i = \{A_{i,1}, \dots, A_{i,|\mathcal{V}_i|}\}$. Here, we aim at learning a global mapping \mathcal{T} which is a matrix whose dimensionality is $\Omega \times |\mathcal{A}_i|$. In other words, we aim at recovering an operator which can embed any of the attributes for a given $G_i \in \Gamma$ into a space \mathfrak{R}^Ω .

In this manner, the aim of computation is the recovery of the optimal transformation matrix over the field of attributes for the graphs in Γ . To recover this transformation matrix, we provide a link to Markov Random Field (MRF) models so as to abstract the problem into a graphical setting which takes profit of the inherent strength of Markovian approaches as related to pairwise potentials. To commence, we associate each $V_{i,a} \in \mathcal{V}_i$ with a hidden variable X_a in the state space Λ . The probability distribution represented by the MRF is given by

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{G_i \in \Gamma} \prod_{V_{i,a} \in \mathcal{V}_i} \zeta_i(X_a) \prod_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \prod_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} \varphi_{i,k}(X_a, X_b) \quad (1)$$

where $\mathcal{X} = \{X_a\}_{a=1, \dots, |\mathcal{V}_i|}$ is the set of hidden variables, $\zeta_i(X_a)$ and $\varphi_{i,k}(X_a, X_b)$ are unitary and binary potential functions which determine the likelihood of the graphs in the data set corresponding to the state $\varrho \in \Lambda$ and $Z = \int_{\Lambda} P(\mathcal{X})$ is the normalisation factor.

Since this normalisation factor is invariant with respect to X_a , the inference of the above MRF model can be recast as an Maximum A Posteriori (MAP) estimation problem to maximise the probability $P(\mathcal{X})$ over the state space Λ . Moreover, we can consider X_a as a vector of continuous variables whose elements can be viewed as the linear product such that $X_a = \mathcal{T} \mathbf{A}_i(a)$, where $\mathbf{A}_i(a)$ is the row indexed a of the matrix \mathbf{A}_i , whose rows correspond to the node attribute set \mathcal{A}_i for the graph G_i . In other words, the hidden variables correspond to the embeddings of the graph-vertex attributes onto the space defined by the linear mapping $\mathcal{T} : \mathcal{A}_i \mapsto \mathfrak{R}^\Omega$. From an alternative viewpoint, we can consider X_a to be the weighted analogue of the attribute-vector for the a^{th} vertex in the graph indexed i in Γ .

Taking the logarithm of Equation 1, we have

$$\log P(\mathcal{X}) = \sum_{G_i \in \Gamma} \sum_{V_{i,a} \in \mathcal{V}_i} X_a^T c_i(a) + \sum_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \sum_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} X_a^T w_{i,k}(a, b) X_b \quad (2)$$

where $\log \zeta_i(X_a) = X_a^T c_i(a)$ and $\log \varphi_{i,k}(X_a, X_b) = X_a^T w_{i,k}(a, b) X_b$ are determined by the potential functions. Note that, in the expression above, $c_i(a)$ is a vector and $w_{i,k}(a, b)$ is a matrix, respectively. Also, the normalisation factor has become an additive constant and, as a result, we have removed it from further consideration.

Maximising the above cost function is equivalent to solving the original MRF inference problem, as defined in Equation 1. The cost function is in quadratic form and, hence, it is a natural choice to apply quadratic programming techniques to solve the relaxation problem. However, the Hessian of Equation 2 is determined by the coefficients of the second order term $w_{i,k}(a, b)$ which are not necessarily convex. A number of techniques have been proposed to relax the discrete problem above and convert the MRF cost functional into more tractable forms. Along these lines, some examples are SDP [12], SOCP [13], and spectral relaxation [14].

Instead of finding a continuous relaxation for the original cost function of the MRF model, we propose an alternative cost function which is closely related to it. Notice that the first and the second terms on the right-hand-side of the cost function in Equation 2 can be treated as correlation terms. The first of them measures the correlation between the graph and the single node potential. The second term measures the compatibility between graphs and the pairwise node-potential $w_{i,k}(a, b)$. By thinking of correlation as a measure of similarity and viewing it as an inverse distance, we can transform the maximisation problem at hand into a minimisation one. To do this, the $L2$ norm is a natural choice. The corresponding cost function is hence defined as follows

$$\min f(x) = \sum_{G_i \in \Gamma} \sum_{V_{i,a} \in \mathcal{V}_i} \|c_i(a) - x_a\|^2 + \eta \sum_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \sum_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} \|w_{i,k}(a, b)\|^2 \|x_a - x_b\|^2 \quad (3)$$

where η is a regularisation constant. For the sake of consistency, we have used vector norms where appropriate.

The reformulation of the cost function as above has several appealing properties. First, it is closely related to the MRF model in terms of its physical meaning. Like the MRF, our cost function also accommodates two complementary terms, i.e. a term which measures the compatibility between the data and its transformed field variable and a smoothness term which can be used to enforce the consistency between the variables for those nodes corresponding to the same graph, i.e. $i = k$. The main difference in the cost functions above is the replacement of the inner product with squared distance. Secondly, the cost function defined above is convex. Thus, we can always attain globally optimal solutions for the relaxed problem on the continuous variables. Moreover, the problem can be reduced to that of solving a sparse linear system of equations with positive semidefinite Hessian.

2.2 The L2-norm

In this section, we explore the use of the L2-norm for purposes of recovering the mapping \mathcal{T} . We show how the extremisation of the cost function defined in Equation 3 can be reduced to that of solving a sparse linear system of equations. Recall that we have let $x_a = \mathcal{T} \mathbf{A}_i(a)$, then the cost function can be rewritten as follows

$$\begin{aligned} \operatorname{argmin}_{\mathcal{T}} f(\mathcal{X}) = & \sum_{G_i \in \Gamma} \sum_{V_{i,a} \in \mathcal{V}_i} \|c_i(a) - \mathcal{T} \mathbf{A}_i(a)\|^2 \\ & + \eta \sum_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \sum_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} \|w_{i,k}(a,b)\|^2 \|\mathcal{T} \mathbf{A}_i(a) - \mathcal{T} \mathbf{A}_k(b)\|^2 \end{aligned} \quad (4)$$

Further, by using the factorisation properties of matrix norms and enforcing consistency between those nodes corresponding to the same graph, i.e. $\|w_{i,k}(a,b)\| = 1$ iff $i = k$ and zero otherwise, we can greatly simplify the equation above as

$$\operatorname{argmin}_{\mathcal{T}} f(\mathcal{X}) = \sum_{G_i \in \Gamma} \sum_{V_{i,a} \in \mathcal{V}_i} \|c_i(a) - \mathcal{T} \mathbf{A}_i(a)\|^2 + \eta \gamma \|\mathcal{T}\|^2 \quad (5)$$

where

$$\gamma = \sum_{\substack{G_i \in \Gamma \\ G_k \in \Gamma}} \sum_{\substack{V_{i,a} \in \mathcal{V}_i \\ V_{k,b} \in \mathcal{V}_k}} \|w_{i,k}(a,b)\|^2 \|\mathbf{A}_i(a) - \mathbf{A}_k(b)\|^2 \quad (6)$$

Since γ does not depend on \mathcal{T} , and, hence, becomes a constant, from now on, and for the sake of convenience, we use the shorthand $\lambda = \eta \gamma$.

To minimise the cost function above, in practice, we can treat the problem as a continuous relaxation one which leads to a convex quadratic optimisation problem. To this end, we constraint the rows of the transformation matrix to add up to unity and introduce the vector of lagrange multipliers \mathbf{N} . The cost function becomes

$$g = f(\mathcal{X}) - \mathbf{N}^T (\mathcal{T}^T \mathbf{e} - \mathbf{e}) \quad (7)$$

where \mathbf{e} is a vector of ones whose dimensionality is given by the context.

With these ingredients, we compute the partial derivative with respect to \mathcal{T} . We get

$$\frac{\partial g}{\partial \mathcal{T}} = 2(\mathcal{T} \mathbf{A}_i^T - \mathbf{C}_i^T) \mathbf{A}_i^T + 2\lambda \mathcal{T} \quad (8)$$

where \mathbf{C}_i is a matrix whose a^{th} row corresponds to the vector $c_i(a)$ for the node indexed a in the graph G_i and \mathbf{A}_i is the matrix defined earlier.

We now introduce the shorthands $\mathbf{F}_i = 2\mathbf{A}_i^T \mathbf{A}_i$ and $\mathbf{G}_i = 2\mathbf{C}_i^T \mathbf{A}_i^T$. As a result, we can now write the partial derivative above in the following manner

$$\frac{\partial g}{\partial \mathcal{T}} = \mathcal{T} \mathbf{F}_i - \mathbf{G}_i + 2\lambda \mathcal{T} \quad (9)$$

Following a similar approach, we can compute the partial of the function g with respect to the Lagrange multipliers in \mathbf{N} . By equating both partial derivatives to zero, we can write the solution as a the linear equation. This linear equation can be written using matrix notation in a straightforward manner by adding over the graphs in Γ .

3 Implementation Issues

Based on the above components, we now turn our attention to the implementation and application of our embedding method. Here, training can be viewed as the stage where the linear mapping \mathcal{T} is learned from the graph vertex-attributes and the PCA of the incidence mapping. The testing stage then becomes the use of the mapped graph attributes into the target space for different purposes, for example, categorisation and relational matching.

The training stage starts from constructing the attributes of the graph-vertices. The attribute-set is application dependent. Here, we view, in general, the vertex-attributes $\mathbf{A}_i(a)$ as vectors, where each of these has a one-to-one correspondence to a graph vertex. This also permits the computation of the weight matrix \mathcal{W} with elements $\mathcal{W}(a, c)$ for the graph G_i . The weight matrix \mathcal{W} can be related to the un-normalised Laplacian through the relationship $\mathcal{L} = \mathbf{D} - \mathcal{W}$, where \mathbf{D} is a diagonal matrix such that $\mathbf{D} = \text{diag}(\text{deg}(1), \text{deg}(2), \dots, \text{deg}(|\mathcal{V}_i|))$ and $\text{deg}(c) = \sum_{a=1}^{|\mathcal{V}_i|} \mathcal{W}(a, c)$ is the degree of the node indexed c in the graph [4].

The use of the graph Laplacian is important, since it permits the computation of the unary potentials $c_i(a)$. Consider the mapping \mathcal{I} of the functions $g(e)$ over the set of edges \mathcal{E}_i to all the functions $h(\cdot)$ over the set of vertices \mathcal{V}_i . The incidence mapping \mathcal{I} is then an operator such that $\mathcal{I}g(e) = h(e_+) - h(e_-)$, where the nodes $V_{i,a} = e_+$ and $V_{i,c} = e_-$ are the head and tail, respectively, of the edge $e \in \mathcal{E}_i$. As a result, \mathcal{I} is a $|\mathcal{V}_i| \times |\mathcal{E}_i|$ matrix which satisfies

$$\mathcal{L} = \mathcal{I}\mathcal{I}^T \quad (10)$$

Note that the incidence mapping \mathcal{I} is independent of the orientation of the edges in \mathcal{E}_i . Moreover, it is an operator, so it is independent of the vertex-basis, i.e. it is permutation invariant [10]. Further, the incidence mapping is recovered via a Young-Householder [15] decomposition on \mathcal{L} . With these ingredients, we proceed to define the unary potential $\zeta_i(\mathcal{X}_a)$ as an exponential family over the optimal linear transformation of the incidence mapping \mathcal{I} into a space in \mathbb{R}^Ω . Thus, in practice, we can recover the potential $\zeta_i(\mathcal{X}_a)$ using the Principal Component Analysis (PCA) of the matrix \mathcal{I} . This is, we perform an eigendecomposition on \mathcal{I} so as to select the leading Ω eigenvectors of the incidence mapping. This yields $c_i(a) = [\phi_1(a), \phi_2(a), \dots, \phi_\Omega(a)]^T$, where $\phi_k(a)$ is the a^{th} coefficient of the k^{th} eigenvector of the incidence mapping \mathcal{I} for the graph indexed i in Γ , such that $\phi_k = [\phi_k(1), \phi_k(2), \dots, \phi_k(|\mathcal{V}_i|)]^T$.

Our choice of unary potential hinges in the developments in [16]. It can be shown that the space spanned by the PCA analysis is equivalent to the vertex-to-node scatter for the graph. Thus, we can view the terms $c_i(a)\mathcal{X}_a^T \mathbf{e}$ as the projections of the vectors \mathcal{X}_a onto the subspace defined by the principal directions of the covariance for the mapping between the sets \mathcal{V}_i and \mathcal{E}_i in G_i . With $c_i(a)$ at hand, the linear mapping matrix \mathcal{T} can be solved by extremising $f(\mathcal{X})$ as described in the previous sections.

As related to computational complexity, note that the embedding recovery is effected via Quadratic Programming and, therefore can be solved in polynomial time. The embedding operation, in practice, is a matrix multiplication, which can also be rendered computationally efficient. In summary, the step sequence of the method is as follows:

1. For every graph in Γ , compute the corresponding incidence mapping \mathcal{I} via the Young-Householder decomposition of the Laplacian \mathcal{L} .
2. Compute the vectors $c_i(a)$ via PCA on the incidence mappings for the graphs in the data set.
3. Compute the linear mapping \mathcal{T} by extremising the cost function in Equation 5.

Using the the linear mapping matrix \mathcal{T} , we can transform any $\mathbf{A}_i(a)$ into a target space, where each graph is represented as a matrix whose a^{th} row corresponds to the coordinates associated to the attribute indexed a in the i^{th} graph in Γ . As a result, relational matching between graphs can be performed by comparing the distances between the transformed attributes. This is due to the fact that there is a known one-to-one relationship between vertices and attributes in the graph. Further, these row vectors can be used to represent each graph as a probability distribution of pairwise vertex distance in the target space. In practice, these can be done via a histogram of distance frequencies whose bin-centres in the embedding space can be recovered using a clustering method such as k -means or maximum-likelihood estimation (MLE) approaches. This can be viewed as a codebook in the target space. In this way, we transfer the structural representation of a graph into a statistical representation that can be used for categorisation or relational matching tasks.

4 Experimental Results

Now, we turn our attention to the applicability of the embedding \mathcal{T} to shape categorisation and relational matching settings. We use the MPEG7 CE-Shape-1 shape database, which contains 1400 binary shapes of 70 different classes with 20 images in each category. Fig. 1 shows some examples in the dataset. We have represented each shape as a graph whose vertices correspond to contour pixels sampled in regular intervals. Here, we have sampled 1 in every 10 pixels on the shape contours. With the sample contour pixels, we build a fully connected graph whose edge-weights are given by the Euclidean distances on the image plane between each pair of pixel-sites. Thus, the entries of the weight matrix for the graph correspond to the pairwise distances between the image-coordinates for every pair of vertices in the graph. The weigh matrix is then normalised to unity so as to have every weight in the graph in the interval $[0, 1]$. The attribute set is given by the frequency histogram of these distances for every clique. That is, for the a^{th} vertex in G_i , $\mathbf{A}_i(a)$ is given by the histogram for the edge-weights for the clique centered at the node indexed a . In our experiments, we have used 12 bins for the frequency histogram computation.

4.1 Relational Matching

Firstly, we illustrate the applicability of the embedding for relational matching making use of sample shapes in the dataset. We have learned the embedding \mathcal{T} for the MPEG7 CE-Shape-1 database so as to embed the set of graphs corresponding to the shapes into a space whose $\Omega = 70$. Once the embedding is at hand, relational matching was performed by viewing the node matching task as a point-matching one in the embedding space. We have used the coordinates $\mathcal{X}_a = \mathcal{T}\mathbf{A}_i(a)$ in the target space in order to

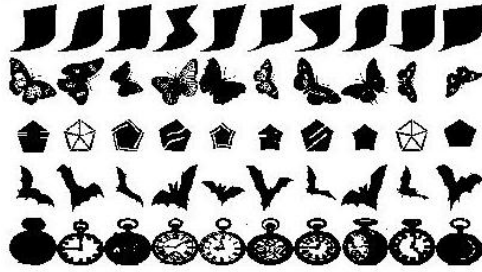


Fig. 1. Samples images from the MPEG7 CE-Shape-1 dataset

compute the distances between nodes in the graph-vertices to be matched. The correspondences are deemed to be the nearest neighbours for the vertex embeddings for each of the graphs under study. That is, the vertex a in the data graph is a match to the node b in the model graph iff the Euclidean distance between the corresponding embedded coordinate vectors χ_a and χ_b is minimum for all the nodes in the graph pair.

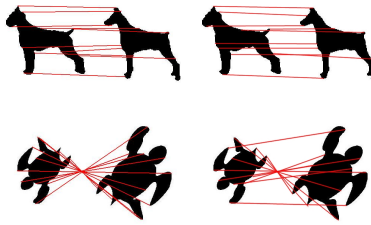


Fig. 2. Example matching results for our embedding (left-hand column) and graduated assignment (right-hand column)

In the the right-hand panels of Fig. 2, we show the results obtained using the recovered embedding \mathcal{T} . In the left-hand panel are the results obtained via graduated assignment [17]. Note that, from the panels, its clear that the distances in the target space provide a means to relational matching. Moreover, qualitatively, the results obtained making use of the embedding \mathcal{T} show less mis-assignments than those recovered using the alternative.

4.2 Shape Categorisation

As mentioned earlier, one of the advantages of the embedding strategy adopted here is that it provides a means to connect structural pattern recognition with statistical pattern recognition. Here, we employ the histogram of pairwise distances in the embedding space for the coordinates $\chi_a = \mathcal{T} \mathbf{A}_i(a)$ and construct a frequency histogram as a graph feature vector that can be used to obtain a “codeword” for each graph. To this end,

Method	Proposed Method	Skeletal Contexts [18]	Shape Contexts [19]	Structured Learning [20]
Accuracy	91.8%	88.4%	76.51%	87.3%

Table 1. Shape categorisation result comparison on the MPEG7-CE-Shape-1 dataset

we have used the frequency histograms for the distances between pairs of embeddings $\chi_a = \mathcal{T}\mathbf{A}_i(a)$ for those attributes in the same graph. These distance histograms have been used to recover a codebook which is computed using k -means clustering, where $k = 200$. Using the pairwise distances for the histogram representation of graphs, we can construct a codebook for all shapes, which we have performed categorisation using a linear SVM. This enables us to perform supervised learning and, thus, the proposed method can take advantage of the recent progresses in machine learning.

For our shape categorisation experiments, we divided the graphs in the MPEG-7 dataset into a training and a testing set. Each of these contains half of the graphs in each dataset. This is, we have used 700 graphs for training and 700 for testing. In contrast to our relational matching examples, here we have recovered the embedding matrix \mathcal{T} making use of those graphs in the training set only. We have tuned the SVM parameters using ten-fold cross validation. The experiments were done on a server with Xeon 2.33GHz CPU and 16G memory. In our experiments, the main computational burden was at training time, where graph generation took approximately 10 minutes, whereas the k -means application and SVM training took 50s.

The categorisation results are shown in Table 1. For purposes of comparing our results with alternatives elsewhere in the literature, we show recognition rates for the skeletal matching method by Demirci *et al.* [18], the shape context method by Belongie *et al.* [19] and the structured learning method by Chen *et al.* [20]. The former two methods are unsupervised categorisation ones, while the last one is a supervised learning method. As shown in Table 1, our method shows a margin of improvement over the alternatives. Note that the alternative methods above have been specifically designed to provide optimum performance on binary shapes. Our method, on the other hand, makes a very simple abstraction of the shape in hand and can be naturally adapted to any shape whose structure can be captured by a relational structure. Moreover, our method is quite general in nature, permitting different tasks, such as the shape matching and categorisation, to be effected in a single computational framework.

5 Conclusions

In this paper, we have proposed a method to recover a mapping which is based upon the graph attribute-set and, at the same time, is inherently related to the graph topology. We have done this by drawing a link between the incidence mapping and a linear operator over the graph-vertex attributes. This linear operator is, in fact, a mapping that can be used for purposes of embedding graphs in a space where matching and categorisation tasks can be effected. We recover this embedding using a Markovian formulation which can be viewed as a learning process over a common topology for the set of graphs under

study. This learning process is based upon a cost function which is convex in nature. We exemplify the utility of our method for shape categorisation and matching on MPEG7 CE-Shape-1 dataset.

6 Acknowledgement

This work was done while Haifeng Zhao was a visiting scholar at the Canberra Research Laboratory of NICTA. He is supported by the National Natural Science Foundation of China (NSFC) under No.60775015.

References

1. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500) (2000) 2319–2323
2. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290** (2000) 2323–2326
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: NIPS. Number 14 (2002) 634–640
4. Chung, F.R.K.: *Spectral Graph Theory*. American Mathematical Society (1997)
5. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Shock-based indexing into large shape databases. In: ECCV. Volume 3. (2002) 731–746
6. Wong, A.K.C., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. *IEEE TPAMI* **7** (1985) 599–609
7. Christmas, W.J., Kittler, J., Petrou, M.: Structural matching in computer vision using probabilistic relaxation. *IEEE TPAMI* **17**(8) (1995) 749–764
8. Wilson, R., Hancock, E.R.: Structural matching by discrete relaxation. *IEEE TPAMI* **19**(6) (June 1997) 634–648
9. Caetano, T., Cheng, L., Le, Q., Smola, A.: Learning graph matching. In: ICCV. (2007) 14–21
10. Biggs, N.L.: *Algebraic Graph Theory*. Cambridge University Press (1993)
11. Bremaud, P.: *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues*. Springer (2001)
12. Keuchel, J.: Multiclass image labeling with semidefinite programming. In: ECCV. (2006) 454–467
13. Kumar, M., Torr, P., Zisserman, A.: Solving markov random fields using second order cone programming relaxations. In: CVPR. (2006) 1045–1052
14. Cour, T., Shi, J.: Solving markov random fields with spectral relaxation. In: Intl. Conf. on Artificial Intelligence and Statistics. (2007)
15. Young, G., Householder, A.S.: Discussion of a set of points in terms of their mutual distances. *Psychometrika* **3** (1938) 19–22
16. Ding, C., He, X.: K-means clustering via principal component analysis. In: ICML. (2004) 225–232
17. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. *IEEE TPAMI* **18**(4) (April 1996) 377–388
18. Demirci, M.F., Shokoufandeh, A., Dickinson, S.J.: Skeletal shape abstraction from examples. *IEEE TPAMI* **31**(5) (May 2009) 944–952
19. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE TPAMI* **24**(24) (April 2002) 509–522
20. Chen, L., McAuley, J.J., Feris, R.S., Caetano, T.S., Turk, M.: Shape classification through structured learning of matching measures. In: CVPR. (2009)