SIGNAL CLASSIFICATION BY MATCHING NODE CONNECTIVITIES

Linh Lieu and Naoki Saito

Department of Mathematics
University of California, Davis
One Shields Avenue, Davis, CA 95616 USA

ABSTRACT

We propose a simple and efficient way for pattern recognition and signal classification within the Diffusion Framework. Our proposed Node Connectivity Matching (NCM) method is derived from the diffusion distance. However, instead of computing the eigenvalues/eigenvectors of the normalized diffusion matrix on the graph constructed from the data, as required when approximating the diffusion distance, we treat each row of the normalized diffusion matrix as a training histogram of node connectivities. To classify an unlabeled data point, we compare its node connectivities to the training histograms using the L^2 norm as a bin-by-bin histogram discriminant measure. Through numerical examples we show that our NCM method is more accurate than using the diffusion distance.

Index Terms— Diffusion distance, normalized diffusion matrix, Markov transition probabilities, directed diffusion, histogram matching.

1. INTRODUCTION

In recent years, methods in the Diffusion Framework have been shown to possess vast applications in many different areas of science, particularly in data mining. When referring to the Diffusion Framework, we refer to a method or procedure that involves in some way utilization of the graph Laplacian on the weighted graph constructed from the data. These methods are in general robust and effective. M. Belkin et al. ([1] and the references therein) proposed to use the eigenvectors of the graph Laplacian for dimension reduction while preserving local geometry. Related to the graph Laplacian is the diffusion operator (defined in (2) below). R. R. Coifman and S. Lafon [2, 3, 5] gave an interpretation of the action of the diffusion operator via a Markov process on the graph and introduced the *diffusion maps* and *diffusion distance* to pattern recognition and many more applications. Following suit are methods for datasets matching [6, 7], diffusion on a graph for clustering and image denoising [4, 9], and many more that we cannot mention due to the limited space here.

Partially funded by the NSF grant DMS-0636297 and the ONR grants N00014-07-1-0166, N00014-09-1-0041, N00014-09-1-0318.

In this paper, we observe that the diffusion distance is essentially a weighted L^2 distance between the rows of the normalized diffusion matrix P. When we view row i of the matrix P as a probability distribution for a random walker to move from node i on the graph constructed from the data to all other nodes, then the diffusion distance can be viewed as the weighted L^2 distance between transition probability distributions (as described in [2, 5]). Furthermore, each of these probability distributions (i.e., rows of the diffusion matrix) can be viewed as a distribution of connectivities of a node to all other nodes in the graph. From this observation, we propose a simple method for classification by directly comparing distributions of node connectivities, instead of by the diffusion distance which requires computation of the eigenvalues and eigenfunctions of the normalized diffusion matrix P.

We shall describe our idea in details in Sec. 4. In Sec. 2 we give a brief review of the definition and properties of the diffusion maps and the diffusion distance. Then in Sec. 3 we briefly describe two known methods within the Diffusion Framework that are closely related to this work: classification using diffusion distance and directed diffusion. Finally we illustrate an application of our proposed Node Connectivity Matching (NCM) method in Sec. 5.

2. DIFFUSION DISTANCE AND DIFFUSION MAPS

Under the Diffusion Framework [2, 5, 6, 7], diffusion maps are employed to achieve spectral embedding of the data. Furthermore, the usual Euclidean distance in the embedding space approximates the diffusion distance defined in (4).

We assume in general that the data $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ lies in a space that possesses a natural dissimilarity measure δ. For example, if X is a database of image patches of size 32×32, then X can be treated as a subset of $\mathbb{R}^{1024}$ and δ the L^2 norm in $\mathbb{R}^{1024}$.

Begin by constructing a weighted connected symmetric graph with points in X as nodes and weights $w(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$ on the edge connecting $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. A common practice is to use the Gaussian weights

$$w_\varepsilon(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq \mathrm{e}^{-(\delta(\boldsymbol{x}_i, \boldsymbol{x}_j)/\varepsilon)^2}, \ \varepsilon > 0. \tag{1}$$

Let W denote the (symmetric) weights matrix with $W_{ij} \triangleq w_\varepsilon(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The weights give a notion of local geometry to the dataset X. When the points in X are on a manifold in $\mathbb{R}^n$, the Gaussian weights W approximates the heat kernel on the manifold [1]. Let D be the diagonal matrix with $D_{ii} \triangleq \sum_j W_{ij}$ the degree of node i. The *normalized diffusion matrix* is defined as

$$P \triangleq D^{-1}W, \tag{2}$$

The matrix P is essentially the matrix of an averaging operator. It is non-negative and row-stochastic (i.e., its eigenvalues are $1 = \lambda_0 > \lambda_1 \geq \cdots \geq 0$, and the sum of each row is 1). Hence, it can be viewed as a transition matrix of a Markov process on X with P_{ij} representing the probability of moving from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ in one step. When the Markov chain is forwarded in time, the probability of moving from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ in t time steps is

$$P^t_{ij} = \sum_\ell \lambda^t_\ell \boldsymbol{\phi}_\ell(i)\boldsymbol{\psi}_\ell(j),\ t \in \mathbb{N}, \tag{3}$$

where $\{\boldsymbol{\phi}_\ell\}$ and $\{\boldsymbol{\psi}_\ell\}$ are (orthonormal) left and right eigenvectors of P, and $\boldsymbol{\phi}_\ell(i)$ means the i^{th} entry of $\boldsymbol{\phi}_\ell$.

Let $P^t_{i\cdot}$ denote row i of the matrix P^t. For t chosen a priori, the *diffusion distance* between two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as

$$D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 \triangleq \left\| P^t_{i\cdot} - P^t_{j\cdot} \right\|^2_{L^2(X, \frac{1}{\boldsymbol{\pi}})}, \tag{4}$$

where $\boldsymbol{\pi}$ is the stationary distribution of the Markov process dictated by P. Notice that this is simply the weighted L^2 distance between row i and row j of the matrix P^t.

The diffusion distance measures the difference in how $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected to all other nodes in the graph, that is, $D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)$ takes into account all incidences relating $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Consequently, it is robust to noise or small perturbations. It is a good tool for extracting the underlying geometry in the dataset X, especially when X lies on a low dimensional manifold in a high-dimensional space.

Using the spectral decomposition (3) of P^t and the mutual orthogonality of $\{\boldsymbol{\phi}_\ell\}$ and $\{\boldsymbol{\psi}_\ell\}$, it can be verified that (see [6])

$$D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = \sum_\ell \lambda^{2t}_\ell \left(\boldsymbol{\psi}_\ell(i) - \boldsymbol{\psi}_\ell(j)\right)^2. \tag{5}$$

Since the eigenvalues λ_ℓ's are non-increasing, the diffusion distance can be approximated to a prescribed relative accuracy $\tau > 0$ by

$$D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 \approx \sum_{\ell=0}^{s(\tau,t)} \lambda^{2t}_\ell \left(\boldsymbol{\psi}_\ell(i) - \boldsymbol{\psi}_\ell(j)\right)^2, \tag{6}$$

where

$$s(\tau, t) \triangleq \arg\max_{\ell \in \mathbb{N}} \{|\lambda_\ell|^t > \tau |\lambda_1|^t\}. \tag{7}$$

A *diffusion map* is defined as

$$\Psi_t : \boldsymbol{x}_i \mapsto \begin{pmatrix} \lambda^t_1 \boldsymbol{\psi}_1(i) \\ \lambda^t_2 \boldsymbol{\psi}_2(i) \\ \vdots \\ \lambda^t_{s(\tau,t)} \boldsymbol{\psi}_{s(\tau,t)}(i) \end{pmatrix}. \tag{8}$$

Ψ_t embeds all points in X into $\mathbb{R}^{s(\tau,t)}$ where the usual Euclidean distance is an approximation to the diffusion distance. In general, $s(\tau, t)$ is much smaller than the original dimension of the data points. The key point to note is that the diffusion map Ψ_t produces a low-dimensional representation of the data that highlights the underlying intrinsic local geometry in the data.

3. CLASSIFICATION UNDER THE DIFFUSION FRAMEWORK

Numerous approaches within the Diffusion Framework have been proposed for classification or pattern recognition applications. We shall describe only two approaches that are most closely related to the topic of this paper.

One approach is to use the diffusion distance as a discriminant measure [2, 5]. In its simplest form this approach involves computing a diffusion map using the training data. Then the diffusion map is extended to the unlabeled (test) data points by Nyström's method or its variants [6, 3]. As described in Sec. 2, the diffusion map embeds the training data into a low-dimensional Euclidean space $\mathbb{R}^{s(\tau,t)}$ in which the usual Euclidean distance is an approximation of the diffusion distance. The extension of the diffusion map embeds the unlabeled data into the same space $\mathbb{R}^{s(\tau,t)}$. The final step is to analyze or classify the data using the Euclidean distance in this space, such as the k-Nearest Neighbor method.

A second approach is known as directed-diffusion or regularized diffusion on a graph [4, 9]. Here, a graph G is constructed using all data points (both training and unlabeled). Then the normalized diffusion matrix P computed on G is used as an operator to *diffuse* the labels from the labeled nodes to the unlabeled nodes. For a K-class classification problem, this is achieved by applying P^t with some $t \in \mathbb{N}$ to a $N \times K$ matrix V whose rows are indexed by the nodes in G. Initially, $V(i, j) = 1$ if node i is known to have label j, and $V(i, j) = 0$ otherwise. After the *label-diffusion* process, the test data point indexed by row i of V is assigned the label ℓ if $\ell = \arg\max_{1 \leq j \leq K} V(i, j)$.

4. CLASSIFICATION VIA NODE CONNECTIVITY MATCHING

In this section, we describe a new discriminant measure that can be utilized for pattern recognition and signal classification. Our idea of Node Connectivity Matching (NCM) is derived directly from (4), the definition of the diffusion distance.

First, we construct a connected weighted graph G with Gaussian weights on the training data. Let the nodes be indexed by the training data points (i.e., if $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N_1}\}$ is the training set, then the nodes in G are labeled by $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N_1}$). Next, we consider each row i of the normalized diffusion matrix on G as a probability distribution (or histogram) of the connectivities of the node $\boldsymbol{x}_i$ to all other nodes in G. Suppose $Y = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{N_2}\}$ is the set of unlabeled data. We add N_2 nodes to G (indexed by $\boldsymbol{y}_j$). For each $\boldsymbol{y}_j$, we construct a probability distribution H_j of connectivities of $\boldsymbol{y}_j$ to all $\boldsymbol{x}_i$. More precisely, H_j is an N_1-bin histogram given by

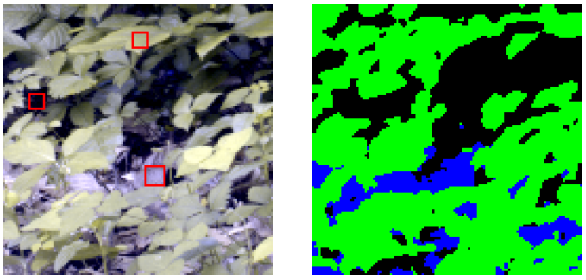
$$H_j(i) \triangleq \frac{w_\varepsilon(\boldsymbol{y}_j, \boldsymbol{x}_i)}{\sum_{i=1}^{N_1} w_\varepsilon(\boldsymbol{y}_j, \boldsymbol{x}_i)}, \tag{9}$$

where w_ε denotes the Gaussian weights defined in (1) and $i = 1, \cdots, N_1$, $j = 1, \cdots, N_2$.

To classify an unlabeled data point $\boldsymbol{y}_j$, we compare its histogram of node connectivities to the training histograms using a bin-by-bin histogram discriminant measure (such as the usual L^2 norm, the Hellinger distance, Jeffrey's divergence, and one-dimensional Earth Mover's Distance). Then use the nearest neighbor classifier to infer a label for $\boldsymbol{y}_j$.

Note that when we use the weighted L^2 norm given in (4) to measure the difference between any two node connectivity histograms of any two training data points, we get exactly the diffusion distance. In general, it is not common practice to compute the exact diffusion distance between the training data and the unlabeled data because that requires computing the normalized diffusion matrix on the union of the training and unlabeled sets. This is infeasible if the size of the unlabeled set is large. Instead, an approximation of the diffusion distance is used, as in the first approach described in Sec. 3. This process requires computing an extension of the diffusion map to the unlabeled data. In the approximation and extension process some error is admitted, which results in less accuracy. Our NCM approach mirrors the diffusion distance in that it takes into account all incidences relating the unlabeled data to the training data. This makes it robust to noise. In addition, we compare the histograms of connectivities directly, instead of performing spectral embedding. This saves time and improves accuracy.

A related method within the Diffusion Framework that does not perform spectral embedding is the directed-diffusion approach described in Sec. 3. Our proposed approach and the directed-diffusion approach are also similar in that both requires a small number of training data. On the other hand, directed diffusion requires the normalized diffusion matrix P to be computed using both training and unlabeled data and a procedure to determine the stopping time for the diffusion process (i.e., the number of time steps t for the operator P^t). Our approach involves computing the matrix P using only the training data and a straightforward comparison between the training histograms and the histograms of the unlabeled data.



(a) Seeds (b) Result

Fig. 1. (a) Three regions selected for training. (b) The result of segmentation by NCM algorithm (Green: leaf pixels, Blue: flower pixels, Black: background pixels).

5. NUMERICAL EXPERIMENTS

We apply our NCM algorithm to classify pixels in a collection of hyperspectral images of natural scenes. For example, in an image of a flowering shrub, we identify a pixel as leaf, flower, or background pixel. One application of such task is segmentation of the image into regions of same pixel types.

Each hyperspectral image is of size 128×128 pixels, and each pixel consists of 43 reflectance values at different wavelengths. In other words, each image is a data cube of size $128 \times 128 \times 43$. Details on all technicalities in the acquisition of the images can be found in [8].

We first extract a 3-by-3 window around each hyperspectral pixel and define this as the feature vector for the pixel. In other words, we run our algorithm on the set of feature vectors associated with the pixels, and each feature vector has length $387 (= 9 \times 43)$ in this case. To compute the Gaussian weights we treat the feature vectors as points in $\mathbb{R}^{387}$ and use the Euclidean distance for δ. The readers are referred to [7] for a detailed description on how to select a value for the scale parameter ε. To measure the dissimilarity between the histograms of node connectivities, we use Jeffrey's divergence, Hellinger distance, one-dimensional Earth Mover's Distance, and L^2 distance. However, these histogram discriminant measures give similar classification results. Therefore, we will report only the results from using the L^2 distance.

We perform our first numerical experiment on a hyperspectral image of a flowering shrub shown in Fig.1. The image consists of three types of pixels (leaf, flowers, and dark background). We identified three small regions (Fig.1a) corresponding to each type for training. The remaining pixels are set as unlabeled pixels. The classification result is shown in Fig.1b.

In order to numerically evaluate the accuracy of our NCM algorithm, we perform more controlled experiments. We segment by hand some leaf, trunk, and rocks regions from four different hyperspectral images. This gives us a three-class recognition problem. We run our experiments on the set of feature vectors associated to these hand picked pixels. For

comparison, we also perform classification via diffusion distance (Diff Dist) and directed-diffusion (Dir Diff) approach described in Sec. 3. Furthermore, to study the importance of the node connectivity histograms, we also try classification via nearest neighbor (NN) in L^2 distance between the feature vectors, i.e., we skip the computation of the node connectivity histograms and move directly to the classification phase.

First, we focus on the feature vectors extracted from the same hyperspectral image (that is, both training and test data come from the same image). For each class, we randomly selected 200 out of approximately 800 feature vectors to use as training data (i.e., we have 600 training points and approximately 1800 test points in total). We repeated this process three times. The average recognition errors over three trials are shown in the first row of Table 1.

Next, we enforce that the training data and the unlabeled data come from four different images. Due to different illumination, the reflectance values in any two leaf, trunk, or rock pixels belonging to different images can be very different. Our goal is to compare the practical applicability of these different classification algorithms. As before, we randomly selected 25% of feature vectors from each class to use as training data. The total sizes are $1500 (= 500 \times 3)$ training points and approximately 4500 test points. The average recognition errors over three randomly trials are listed in the second row of Table 1. We see that NCM performs 11% better in error rate than NN. In other words, the extra computation time spent on constructing node connectivity histograms improves recognition by 11%. The Dir Diff method performs best in our experiments because we computed the (full) diffusion matrix P using both training and test data. It contains all statistics between all data points. Therefore, the performance is best. However, the matrix P in this case is 16 times larger than the one in the NCM method, since the size of training set is only one fourth of the whole.

One advantage of classification under the Diffusion Framework is the ability to handle more effectively the variations in the data caused by different sensing processes, such as handling the difference in illumination in the example above. We can adjust the scale parameter ε in the Gaussian kernel to accommodate these differences. Although, the high error rate admitted by the diffusion distance approach in the example above argues the opposite. However, this high rate is mostly due to approximation error, since we do not compute the diffusion distance exactly. Our NCM algorithm bypasses the approximation process, hence improves accuracy, while maintaining the advantages of classification under the Diffusion Framework.

6. CONCLUSIONS

We have proposed a simple approach based on the diffusion distance for pattern recognition and signal classification. Via numerical experiments on hyperspectral images, we see evidence that our NCM algorithm improves over the approach that uses the diffusion distance for classification. Furthermore, we see that our NCM algorithm can handle variations in images caused by illumination better than directly comparing local feature patches.

Method	NCM	Diff Dist	Dir Diff	NN
Single image	2.23	3.13	1.19	1.37
Multiple images	20.00	57.23	10.57	31.26

Table 1. Classification Error Rates (%). Row 1: training and test data are from same image. Row 2: training and test data are from different images. Column 1: NCM with L^2 as histogram discriminant measure. Columns 2, 3: diffusion distance and directed diffusion approach as described in Sec. 3. Column 4: nearest neighbor in L^2 distance between feature vectors.

7. REFERENCES

[1] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation", *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[2] R. R. Coifman and S. Lafon, "Diffusion maps", *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 5–30, July 2006.

[3] R. R. Coifman and S. Lafon, "Geometric harmonics", *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 31–52, July 2006.

[4] Y. Keller, S. Lafon, and M. Krauthammer "Protein cluster analysis via directed diffusion", Fifth Georgia Tech International Conference on Bioinformatics, Georgia, USA, November 2005.

[5] S. Lafon, "Diffusion Maps and Geometric Harmonics", Ph.D. Dissertation, Yale University, May 2004.

[6] S. Lafon, Y. Keller, R.R. Coifman, "Data fusion and multicue data matching by diffusion maps", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 11, pp. 1784–1797, 2006.

[7] L. Lieu and N. Saito, "High dimensional pattern recognition using diffusion maps and Earth Mover's Distance", submitted to *Signal Processing*, preprint available at http://www.math.ucdavis.edu/˜llieu/pub.html, 2008.

[8] D. L. Ruderman, "Statistics of cone responses to natural images: implications for visual coding", *J. Opt. Soc. Am.*, vol. 15, no. 8, pp. 2036–2045, August 1998.

[9] A. D. Szlam, M. Maggioni, R. R. Coifman, "Regularization on graphs with function adapted diffusion processes", *Journal of Machine Learning Research*, vol. 9, pp. 1711–1739, 2008.