SIGNAL DIFFUSION FEATURES FOR AUTOMATIC TARGET RECOGNITION IN SYNTHETIC APERTURE SONAR

Jason C. Isaacs and James D. Tucker

Advanced Signal Processing and ATR
Naval Surface Warfare Center
Panama City, FL

ABSTRACT

Given a high dimensional dataset, one would like to be able to represent this data using fewer parameters while preserving relevant signal information, previously this was done with principal component analysis, factor analysis, or basis pursuit. However, if we assume the original data actually exists on a lower dimensional manifold embedded in a high dimensional feature space, then recently popularized approaches based in graph-theory and differential geometry allow us to learn the underlying manifold that generates the data. One such manifold-learning technique, called Diffusion Maps, is said to preserve the local proximity between data points by first constructing a representation for the underlying manifold. This work examines binary target classification problems using Diffusion Maps to embed inverse imaged synthetic aperture sonar signal data with various diffusion kernel representations for automatic target recognition. Results over three sonar datasets demonstrate that the resulting diffusion maps capture suitable discriminating information from the signals to improve target recognition and drastically lower the false alarm rate.

1. INTRODUCTION

The central problem in high-dimensional data analysis is the trade-off between computational complexity and the resolution gained with either more features or pixels. Therefore, a typical first step in analyzing high-dimensional data is to find a lower-dimensional representation and the concise description of its underlying geometry and density. This is usually done however, with global dimension reducing techniques such as principal component analysis, and Multidimensional Scaling. These techniques in general work well with well behaved maximally variant data. What if the data is only locally correlated? Then these techniques do not provide informative embedded data. Alternatively, graph based manifold learning techniques offer to embed the data based on local relationship preservation, i.e., they generally preserve the neighborhood structure. Such techniques are Diffusion Maps [2] and [3], Local linear Embedding [8], Laplacian Eigenmaps [1], Hessian Eigenmaps [4], and Local Tangent Space Alignment [10].

In this paper we consider the manifold learning technique Diffusion Maps of Coifman et al. [2], [3] and analyze the time-signal information extraction ability of Diffusion Maps. This is done by looking at the classification results for each binary target dataset using a boosted decision tree.

2. SYNTHETIC APERTURE SONAR

Sythetic aperture sonar (SAS) [6], is similar to SAR in that the aperture is formed artificially from received signals to give the appearance of a real aperture several times the size of the transmit/receive pair. SAS is performed by collecting a set of time domain signals and match filtering the signals to eliminate any coherence with the transmitted pulse. SAS images are generated by beamforming the time domain signals using various techniques, e.g. time-delay, chirp scaling, or ω-k beamforming. In this work the wave-number technique (ω-k) is used. The goal here is to classify target and non-target object signals that have been detected in sonar imagery. The data that will be analyzed is the inverse imaging data from an extracted region of interest around a detected object. For a given a sonar image snippet this involves inverse beamforming the snippet to resolve the target area time-domain signal, or stave, data. This method allows for the elimination of non-target scatterer interference by using only the contributing complex-image target information to derive the time-domain data, see Figure 1.

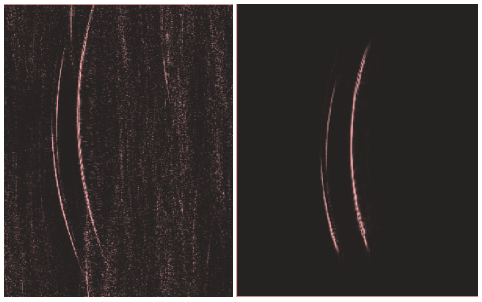

Fig. 1. Images of a sequence of sonar pings that comprise time-domain signals about a target object. The original stave data is on the left and the inverse image data is on the right.

As can be seen in Figure 1, the inverse image data is devoid of the interference that is seen in the original stave data. Figure 2 below shows the inverse image processes as used for the experimental data in this work.

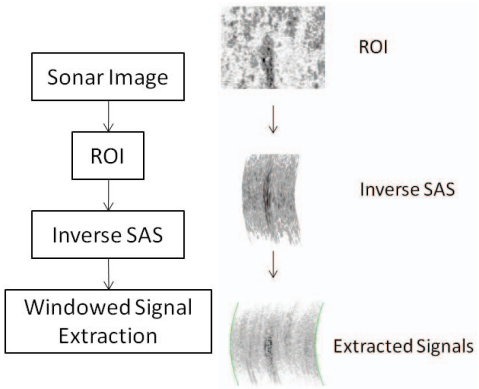

Fig. 2. Example extraction of time-signals from the inverse SAS imaging of a SAS image ROI.

An example of the targets is shown in the image of Figure 2. There are also other non-target objects that will typically be picked-up by a detector. Therefore, the task at hand after detection is to classify these objects appropriately as targets or not.

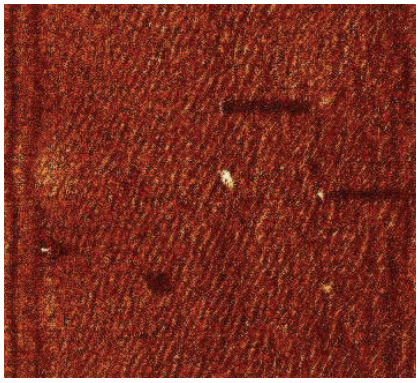

Fig. 3. SWAT SAS Image with targets and non-targets.

3. DIFFUSION MAPS

3.1. Overview

Diffusion Maps are defined as the embedding of complex data onto a low dimensional Euclidian space, via the eigenvectors of suitably normalized random walks over the given dataset. It has been shown, both theoretically in [2] and by examples in [3] how this embedding can be used for dimensionality reduction, manifold learning, geometric analysis of complex data sets and fast simulations of stochastic dynamical systems.

Diffusion Maps are said to preserve the local proximity between data points by first constructing a graph representation for the underlying manifold. The vertices, or nodes of this graph, represent the data points, and the edges connecting the vertices, represent the similarities between adjacent nodes. If properly normalized, these edge weights can be interpreted as transition probabilities for a random walk on the graph. After representing the graph with a matrix, the spectral properties of this matrix are used to embed the data points into a lower dimensional space, and gain insight into the geometry of the dataset. It has been shown in [2] and [3] that the eigenfunctions of Markov matrices can be used to construct coordinates called Diffusion Maps that generate these efficient representations of the complex geometric structures and the associated family of diffusion distances, obtained by iterating the Markov matrix, defines the multi-scale geometries that prove to be useful in the context of data parameterization and dimensionality reduction. The process of constructing these Diffusion Maps as described in [2] and [3] is discussed in sections 3.1 through 3.5.

3.2. Construction of a Random Walk on the Data

Given a data set Ω with a distribution μ of the points on Ω and a kernel k: $\Omega \times \Omega \rightarrow \mathbb{R}$ that satisfies the following properties:

- k is symmetric: $k(x, y) = k(y, x)$,
- k is positivity preserving: $k(x, y) \geq 0$.

This kernel represents some notion of affinity or similarity between points of Ω as it describes the relationship between pairs of points in this set and in this sense, one can think of the data points as being the nodes of a symmetric graph whose weight function is specified by k. The kernel constitutes an a priori presumption of the local geometry of Ω, and since a given kernel will capture a specific feature of the data set, its choice should be guided by the application that one has in mind; this will be discussed later.

It is known that to any reversible Markov process, one can associate a symmetric graph. In addition, the converse is also true, i.e., from the graph defined by (Ω, k), one can construct a reversible Markov chain on Ω. This technique is known as the normalized graph Laplacian construction. The steps are as follows: define

$$d(x) = \int_{\Omega} k(x, y) d\mu(y) \tag{1}$$

to be a local measure of the degree of node x in this graph and define P^t to be an n × n matrix whose entries are given by

$$p_t(x, y) = \frac{k(x, y)}{d(x)} \tag{2}$$

which is the probability of transition from x to y in one time step. For $t = 1$ this can be interpreted as the first-order neighborhood structure of the graph.

3.3. Powers of P and multi-scale geometric analysis of Ω

The matrix P contains geometric information about the data set Ω. The transitions that it defines directly reflect the local geometry defined by the immediate neighbors of each node

in the graph of the data. In other words, $p_1(x, y)$ represents the probability of transition in one time step from node x to node y and it is proportional to the edge-weight $k(x, y)$. For $t \geq 0$, the probability of transition from x to y in t time steps is given by $p_t(x, y)$, the kernel of the t^{th} power P^t of P. Larger powers of P, allows the integration of the local geometry and therefore will reveal relevant geometric structures of Ω at different scales, i.e., larger neighborhoods.

3.4. Spectral Analysis of the Markov Chain

Powers of P constitute an object of interest for the study of the geometric structures of Ω at various scales. A classical way to describe the powers of an operator is to employ the language of spectral theory, namely eigenvectors and eigenvalues. Although for general transition matrices of Markov chains, the existence of a spectral theory is not guaranteed, the random walk constructed here exhibits very particular mathematical properties, i.e., if the graph is connected, which we now assume, then the stationary distribution is unique and we have

$$\lim_{t \to +\infty} p_t(x, y) = \phi_0(y) \tag{3}$$

where the Markov chain has a stationary distribution given by

$$\phi_0(y) = \frac{d(y)}{\sum_{z \in \Omega} d(z)}. \tag{4}$$

The chain is reversible, i.e., it follows the detailed balance condition:

$$\phi_0(x) p_1(x, y) = \phi_0(y) p_1(y, x). \tag{5}$$

The vector $_0$ is the top left eigenvector of P. The spectral analysis of the Markov chain is governed by the following eigen-decomposition

$$p_t(x, y) = \sum_{l \geq 0} \lambda_l^t \psi_l(x) \phi_l(y), \tag{6}$$

where $\{\lambda_l\}$ is the sequence of *eigenvalues* of P (with $|\lambda_0| \geq |\lambda_1| \geq |\lambda_2| \geq \cdots$) and $\{\psi_l\}$ and $\{ _l\}$ are the corresponding biorthogonal right and left eigenvectors.

3.5. Diffusion Distances and Diffusion Maps

The spectral properties of the Markov chain can now be linked to the geometry of the data set Ω. As previously mentioned, the idea of defining a random walk on the data set relies on the following principle: the kernel k specifies the local geometry of the data and captures some geometric feature of interest. The Markov chain defines fast and slow directions of propagation, based on the values taken by the kernel, and as one runs the walk forward, the local geometry information is being propagated and accumulated the same way local transitions of a system can be integrated in order to obtain a global characterization of this system.

Running the chain forward is equivalent to computing the powers of the operator P. For this computation, we could, in theory, use the eigenvectors and eigenvalues of P. Therefore, we are going to directly employ these objects in order to characterize the geometry of the data set Ω. The family of diffusion distances $\{D_t\}_{t \in N}$ is given by

$$D_t^2(x, z) = \sum_{y \in \Omega} \frac{\left(p_t(x, y) - p_t(z, y)\right)^2}{\phi_0(y)}. \tag{7}$$

In other words, $D_t(x, z)$ is a functional weighted l_2 distance between the two posterior distributions $p_t(x, \cdot)$ and $p_t(z, \cdot)$. For a fixed value of t, D_t defines a distance on the set Ω. By definition, the notion of proximity that it defines reflects the connectivity in the graph of the data. Indeed, $D_t(x, z)$ will be small if there is a large number of short paths connecting x and z, that is, if there is a large probability of transition from x to z and vice versa. The main interesting features of diffusion distance are: 1) the points are closer if they are highly connected, 2) $D_t(x, z)$ involves summing over all paths and is therefore robust to noise perturbations, 3) the distance takes into account all evidence relating x and z. $D_t(x, z)$ does not have to be computed explicitly. It can be computed using the eigenvectors and eigenvalues of P:

$$D_t^2(x, z) = \sum_{l \geq 1} \lambda_l^{2t} (\psi_l(x) - \psi_l(z))^2. \tag{8}$$

As previously mentioned, the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ tend to 0 and have a modulus strictly less than 1. As a consequence, the above sum can be computed to a preset accuracy $\delta > 0$ with a finite number of terms: if we define as the number of elements retained to meet this accuracy. Then, up to relative precision δ, we have

$$D_t(x, z) = \left(\sum_{l \geq 1}^{s(\delta, t)} \lambda_l^{2t} (\psi_l(x) - \psi_l(z))^2 \right)^{\frac{1}{2}}. \tag{9}$$

We can therefore introduce a family of diffusion maps $\{\Psi_t\}_{t \in \mathbb{N}}$ given by

$$\psi_t : x \to \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{s(\delta,t)}^t \psi_{s(\delta,t)}(x) \end{pmatrix} \tag{10}$$

Each component of $\Psi_t(x)$ is termed diffusion coordinate. The map Ψ_t: $\Omega \to R^{s(\delta, t)}$ embeds the data set into a Euclidean space of $s(\delta, t)$ dimensions. This method constitutes a universal and data driven way to represent a graph, or any generic data set, as a cloud of points in a Euclidean space. Moreover, $s(\delta, t)$ depends on the properties of the random walk and not on the number of features of the original representation.

4. KERNEL FUNCTIONS

The kernel constitutes our prior definition of the local geometry of Ω, and since a given kernel will capture a specific feature of the data set, its choice should be guided by the application. For comparisons sake this work will use the following selection of kernels:

- Laplacian Kernel: $k(x, y) = \exp\left(-\left\|x - y\right\| - \mu \middle/ b\right) / 2b$,

- Gaussian Kernel: $k(x,y) = \exp(-\|x-y\|^2 / 2\ ^2)$,
- Rayleigh Kernel: $k(x,y) = \dfrac{\|x-y\|\exp(-\|x-y\|^2 / 2\ ^2)}{\sigma^2}$,
- Polynomial Kernel: $k(x,y) = (1+\langle x,y\rangle)^d$

where the Gaussian and Polynomial kernels are most familiar from support vector machines. The Laplacian and Rayleigh were used previously in [7].This list is by no means exhaustive and was not constituted in any optimal way.

5. EXPERIMENTS

5.1. Experimental Setup

The problem here is to analyze the discriminating feature embeddings of the resultant diffusion maps for the classification of target and non-target objects in SAS data. Each SAS dataset is divided into five groups that are as equal as possible, 5-fold cross validation. One group is set aside for the training set and four groups for the dedicated testing set. This procedure is continued until all groups have represented as a testing set. The average performance overall five-folds is presented as the probability of classification (P_C), or sensitivity, and the probability of false alarm (P_{FA}), or specificity. This is done to demonstrate the trade-off between correctly classifying true cases versus incorrectly classifying false cases. Each kernel uses the same groups for each data set so that the possibility of poor individual performance due to the distribution of the draw is reduced as best as possible.

An Adaboosted decision tree is used to classify the datasets. Adaboost is a machine learning algorithm, formulated by Yoav Freund and Robert Schapire [5]. It is a meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is somewhat sensitive to noisy data and outliers. Otherwise, it is less susceptible to the overfitting problem than most learning algorithms. AdaBoost calls a weak classifier repeatedly in a series of rounds. For each call a distribution of weights is updated that indicates the importance of examples in the data set for the classification, i.e., the difficulty of each sample. For each round, the weights of each incorrectly classified example are increased (or alternatively, the weights of each correctly classified example are decreased), so that the new classifier focuses more on those difficult examples. Experiments here are performed with 200 iterations of boosting a simple decision tree. The experimental variable values are listed below.

Experimental Variables
δ= 1e-7, α = 1, b = 2, μ = 1, σ^2 =3, d = 3.

Where δ is the diffusion threshold, α is the diffusion probability distribution scaling, b is the Laplacian kernel scaling parameter, μ is the mean for the Laplacian kernel, σ^2 is the variance for the Gaussian kernel and the square of the mode for the Rayleigh kernel, and d is the polynomial kernel degree.

5.2. Data Sets

The experiment discussed above tests the diffusion features for target classification enhancement on the following three SAS data sets [9]:

- Sonar1: Shallow Water Acoustic Toolset [9] low frequency 1.
- Sonar2: Shallow Water Acoustic Toolset [9] low frequency 2.
- Sonar3: Shallow Water Acoustic Toolset [9] high frequency 2.

For each data set listed above, Table 1 below includes the number of samples, the class distribution, and the number of features, or attributes.

6. RESULTS AND CONCLUSIONS

The experimental results for the diffusion map time-signal feature extraction are shown below in Tables 2 through Table 4. The tables are listed per sonar dataset with each kernel given a column. The rows correspond to the diffusion map and original signal results for each set. Table 2 shows that for the sonar1 low frequency dataset that the diffusion map results are far better than the original signal classifications results, especially noted is the improvement in the false alarm rate. There is no apparent affect of kernel choice on the diffusion map feature extraction algorithm for this particular application. It is apparent; however that the diffusion map features allow for a much improved classification of targets vs. non-targets for these three SAS datasets over using the original inverse image time-signals.

As the experiments demonstrate, here the choice of kernel does not affect the resultant diffusion map. Overall, the four kernels performed rather well with the polynomial-3 kernel having a better average false alarm rate reduction. Regardless, for enhanced target recognition capability and an acceptable P_{FA} diffusion maps appear to extract the relevant information from these sonar object signals that best discriminates between targets and non-targets for ATR.

REFERENCES

[1] M. Belkin; P. Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation," Neural Computation, v.15 n.6, p.1373-1396, June 2003.

[2] R. Coifman; S. Lafon, "Diffusion Maps," Applied and Computational Harmonic Analysis, special issue on diffusion maps and wavelets, vol. 21, pp. 5-30, July 2006.

[3] R. Coifman; S. Lafon; A. Lee; M. Maggioni; B. Nadler; F. Warner; S. Zucker, “Geometric Diffusions as a Tool for Harmonics Analysis and Structure Definition of Data: Multiscale Methods,” Proc. Nat"l Academy of Sciences, vol. 102, no. 21, pp. 7432-7437, May 2005.

[4] D. Donoho; C. Grimes, "Hessian Eigenmaps: New Locally Linear Embedding Techniques for High-Dimensional Data," Proc. Nat'l Academy of Sciences, vol. 100, no. 10, pp. 5591-5596, May 2003.

[5] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt '95*, pages 23–37. Springer-Verlag, 1995.

[6] Gough, P.T., A synthetic aperture sonar system capable of operating at high speed and in turbulent media. IEEE Jour. Oceanic Eng., 11(2), pp333, 1986.

[7] Isaacs, J.C.; Foo, S.Y.; Meyer-Baese, A., "Novel Kernels and Kernel PCA for Pattern Recognition," Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on , vol., no., pp.438-443, 20-23 June 2007

[8] S. Roweis; L. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” Science, vol. 290, pp. 2323-2326, 2000.

[9] G. Sammelmann; J. Christoff; J. Lathrop; “Synthetic Images of Proud Targets”, Proc. IEEE/MTS OCEANS 2004, pp. 266-271.

[10] Z. Zhang; H. Zha, "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," Technical Report CSE-02-019, Dept. of Computer Science and Eng., Pennsylvania State Univ., 2002.

TABLE 1. EXPERIMENTAL DATA SETS

Data Set	# Samples	# Class 0	# Class 1	Signal Length
Sonar1	8600	7750	850	224
Sonar2	1240	930	310	204
Sonar3	1240	930	310	203

TABLE 2. EXPERIMENTAL RESULTS FOR SONAR1

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(224,40)	PC: 0.99 FA: 0.094	PC: 0.99 FA: 0.089	PC: 0.99 FA: 0.073	PC: 0.99 FA: 0.085
(224,40)	Inverse Image Signals without Diffusion Embedding PC: 0.984 FA: 0.235			

TABLE 3. EXPERIMENTAL RESULTS FOR SONAR2

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(204,40)	PC: 0.99 FA: 0.0172	PC: 0.993 FA:0.0215	PC: 0.99 FA:0.0247	PC: 0.99 FA:0.013
	Inverse Image Signals without Diffusion Embedding PC: 0.923 FA: 0.134			

TABLE 4. EXPERIMENTAL RESULTS FOR SONAR3

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(203, 40)	PC: 0.965 FA: 0.112	PC: 0.968 FA: 0.1183	PC: 0.965 FA: 0.109	PC: 0.96 FA: 0.1065
(203, 40)	Inverse Image Signals without Diffusion Embedding PC: 0.877 FA: 0.153			