Transactions on Machine Learning
and Data Mining
Vol. 4, No. 2 (2011) 75-95

ISSN:1865-6781,
ISBN: 978-3-942952-07-1

ibai Publishing
www.ibai-publishing.org

Spherical Classification of Remote Sensing Data

Dalton Lunga[1,2] and Okan Ersoy[1]

[1]Purdue University, West Lafayette, USA.
[2]CSIR-Meraka Institute, Brummeria, Pretoria, South Africa
{dlunga,ersoy}@purdue.edu

Abstract. Real data are often characterized by high dimensional feature vectors. However, such data contain redundant information that may not be beneficial for analysis algorithms. As such, feature transformation arises in related fields of study, including geoscientific applications, as a means to capture the few characteristics that are useful for pattern analysis algorithms. In this study, we investigate the transformation of remote sensing images to a coordinate system that preserves local pixel relationships on a constant curvature space. The transformation is performed using the spherical embedding method. Based on the properties of spherical surfaces and their relationship with local tangent spaces, we further propose two geometrical spherical nearest neighbor metrics for classification. As part of experimental validation, results on modeling multi-class multispectral and hyperspectral data using the proposed spherical Mahalanobis nearest neighbor rule and the spherical discriminant adaptive nearest neighbor rule are presented. The results indicate that the proposed metrics yield better classification accuracies on lower dimensional spherical surfaces. This promising outcome serves as a motivation for further development of new models to analyze remote sensing images in spherical manifolds.

Keywords: Spherical Embedding, Diffusion Maps, Locality Preserving Projections, Remote Sensing Imagery, Spherical Nearest Neighbor Rules, Classification, Riemannian Manifolds, Exponential Maps, Tangent Spaces

1 Introduction

The acquisition of high resolution remote sensing images is increasingly providing additional details that are significant in many studies including disaster

monitoring, land cover studies, scene understanding in computer vision, monitoring by coastal guards, etc. The acquired additional details make it practical to reach better classification decisions because of the increased discriminative details captured by the sensors. However, an increase in acquired high-resolution details has a tendency of increasing redundant information which degrades the performance of analysis algorithms. A remedial approach to such challenges is to map the high dimensional feature vectors to a lower dimensional space for further analysis or better visualization. Often, this is achieved by implementing linear feature extraction methods such as principal component analysis (PCA) [1]. However, the inherent increase in spectral resolution with remote sensing imaging sensors makes complex nonlinear dependencies of details in objects to be easier to capture. Then, approaches that use linear methods for feature extraction are not very effective. Such an observation calls for nonlinear feature extraction methods or transformations that preserve useful nonlinear dependencies that are relevant for data discrimination.

In this paper, we exploit the nonlinear structure of remote sensing imagery by using a nonlinear feature transformation method to enable better visualization and classification of remote sensing data. The approach embeds data onto a constant curvature coordinate system which preserves local neighborhood relations from the high-dimensional feature space. A curvature coordinate system is an example of a Riemannian manifold [2]. We propose spherical nearest neighbor rules for classification on a Riemannian manifold on this basis. Our approach falls into the realm of ongoing work on manifold learning as currently being explored in various research areas including the machine learning community. Manifold learning methods are commonly becoming a standard to embed data onto lower dimensional spaces [3].

The nonlinear transformation of data onto a lower dimensional representation can be accomplished by making use of techniques such as multidimensional scaling (MDS) [4], `diffusion maps` [5], locally linear embedding (LLE) [6] and locality preserving projections (`LPP`) [7]. The amount of distortion in manifold representations varies across each method due to the nature of the metric space used and how well the intrinsic geometry of the space captures the characteristics of the data. The widely used PCA and MDS methods involve the modeling of linear variabilities in multidimensional data. In PCA, one computes the linear projections of greatest variance of the eigenvectors of the data covariance matrix. In MDS, one computes the low dimensional embedding that best preserves pairwise distances between data points. If the pairwise relations are based on the Euclidean distances, as in the case of classical scaling, then the results of MDS are equivalent to PCA (up to a linear transformation). Both approaches do not seem to have any built-in mechanism for determining the geometry of the manifold. However, the methods are simple to implement, and their optimizations are well understood and not prone to local minima. These virtues account for the widespread use of PCA and MDS, despite their inherent limitations as linear methods. On the other hand LLE, `diffusion maps` and `LPP` are capable of generating highly nonlinear embedding. Like PCA and MDS, solutions to these

problems are obtained by solving an eigenvalue problem that scales well with large high dimensional data sets.

Recently, a new eigenvector method was proposed - the spherical embedding (`SE`) for mapping of dissimilarity matrices onto constant curvature manifolds [8]. The spherical embedding approach maps the dissimilarity of feature vectors onto a spherical manifold. `SE` embeds data onto a metric space while optimizing over the kernel distance matrix of positional vectors. This approach is suitable for feature vectors whose magnitude is irrelevant for analysis. Thus, the discrimination of feature vectors is achieved by considering the angle information contained for each transformed feature. We note that use of angle information could also be achieved by simply normalizing each vector by its norm, However such an approach would not be suitable for manifold embedding since the neighborhood structure of points is not preserved (*e.g.* consider two example vectors $\boldsymbol{x}_1 = [2, 0]^T$ and $\boldsymbol{x}_2 = [5, 0]^T$. Applying the L_2 normalization the resulting images of these vectors have the same angle and unit magnitude thus are mapped to the same point on a circle - resulting in loss of identity and any neighborhood relations).

Each of the above approaches represents an attempt to derive a coordinate system that resides on the manifold. The nonlinear methods represent a well studied new class of algorithms that can be brought to bear on many high dimensional applications that exhibit nonlinear structure, e.g., the analysis of remote sensing imagery. It is often the case that researchers would like to carry out some analysis on the embedded data, e.g. perform classification. Due to its simplicity, the one-nearest neighbor classifier is a widely used method. Beyond the one-nearest neighbor classifier, other methods would have to incorporate the geometry of the manifold. The main question that is still a challenge remains - how do we design manifold based classifiers? We make use of simple tools from Lie algebras to design classification methods that incorporates the underlying structure of the manifold. The tools enable us to propose two geometrically based classifiers: spherical Mahalanobis nearest neighbor `(sphMahalanobis)` and the spherical discriminant adaptive nearest neighbor `(sphDann)` methods. The formulation of these classifiers is based on relationships that spherical manifolds have with their local tangent spaces [2]. Such metrics and hence the resulting neighborhoods, depend on the test point locations on the spherical manifold. We compare the results of the spherical lower dimensional representation and `sphDann` classification results to both the `diffusion maps` [5] and the `LPP` embedding approach [7].

The paper is structured as follows: Table 1 presents the description of the notations used in this paper. Section 2 presents the procedure for the spherical embedding method. Section 3 presents the proposed spherical nearest neighbor metrics. Section 4 gives a summary of the two related techniques that we used to compare our work. Section 5 gives the summary of experimental results and ideas related to future work. Section 6 presents our conclusions.

Table 1. Symbol definitions

Symbol	Description	Symbol	Description
$\mathbb{R}$	Real numbers	$\boldsymbol{P}$	Markov chain matrix
$\mathbb{N}$	Natural numbers	$w_{ij} \in \boldsymbol{W}$	The similarity weight
$\mathbb{R}^d$	d-dimensional Euclidean space	ϵ	Neighborhood strip parameter
$\langle \cdot, \cdot \rangle$	Inner product	θ_{ij}	Angle
$\| \cdot \|$	Euclidean norm	π_j	Class j probability
N	Number of observations	$k(\cdot, \cdot)$	Kernel function
C_j	Class j label	$p(\cdot, t \| \cdot)$	Posterior probability
N_j	Sample size in C_j	ψ	Diffusion eigenvector
G	Graph	λ	Eigenvalue
E	Edge set	$\boldsymbol{\psi}$	Diffusion map
V	Vertex set	$\boldsymbol{v}_i$	LPP Eigenvector
$\boldsymbol{X}$	$N \times d$ matrix	$\boldsymbol{x}_i \in \boldsymbol{X}$	Row vector
$\boldsymbol{Z}$	$N \times m$ matrix	$\boldsymbol{z}_i \in \boldsymbol{Z}$	Row vector
$\boldsymbol{W}$	Similarity matrix	M	Lower dimensional manifold
$\boldsymbol{U}$	Eigenvector matrix	$T_{\boldsymbol{z}}M$	Tangent space to M w.r.t $\boldsymbol{z}$
$\boldsymbol{\Lambda}$	Eigenvalue matrix	$Exp_{\boldsymbol{z}}$	Exponential mapping w.r.t $\boldsymbol{z}$
$\boldsymbol{\Sigma}$	Covariance matrix	$\mathcal{N}$	Normal distribution
$\boldsymbol{\Sigma}^{-1}$	Inverse covariance matrix	d^g	Geodesic distance
$\boldsymbol{S}_w$	Within-class covariance matrix	d^e_{ij}	Euclidean distance
$\boldsymbol{S}_B$	Between-class covariance matrix	d_m	Spherical mahalanobis distance
$\boldsymbol{I}$	Identity matrix	d_{sd}	Spherical-adaptive-distance
$\boldsymbol{X}_{\boldsymbol{x}_0}$	Neighborhood to $\boldsymbol{x}_0$	d_t	Diffusion distance
$\boldsymbol{D}$	Degree matrix		
$\boldsymbol{L}$	Laplacian matrix		
$\boldsymbol{L}_{diff}$	Diffusion Laplacian matrix		

2 Riemannian Manifold Geometry

In non-Euclidean spaces, computations are carried out by using different tools than the standard methods used in a Euclidean space. The geometry that exists in Riemannian manifolds dictates how these tools are formulated [2]. On a spherical manifold, a convenient way to measure the distance between two points is no longer the straight line between the points as in the Euclidean space. Distances on spherical surfaces are defined as the length of the shortest curve between a pair of points (this defines the notion of *geodesic*). Later, we revisit the notion of geodesic distance in the context of *Tangent Spaces*. In this section we simply discuss the spherical embedding (SE) formulation [8].

A d-dimensional Riemannian space is defined by its tensor g_{ij} in some local coordinate system $u_1, u_2, \ldots, u_d$. This is usually related to an infinitesimal distance element in the space by

$$ds^2 = \sum_{ij} g_{ij} du_i du_j \tag{1}$$

The metric must be positive definite, and any metric tensor defines a particular Riemannian space. A simple form of a Riemannian manifold that easily relates to directional data is the elliptic manifold of unit radius.

2.1 Spherical Manifolds

An elliptic manifold is an example of a constant curvature space that is defined as the geometry of a spherical surface. In some cases, a hypersphere can easily be embedded in the Euclidean space, for example, the embedding of a unit sphere in three dimensions is

$$\boldsymbol{z} = (\sin u \sin v, \cos u \sin v, \cos v)^T \tag{2}$$

A spherical embedding (elliptic) implies a metric tensor of the form

$$ds^2 = dx^2 + dy^2 + dz^2 \tag{3}$$

$$= sin^2 v du^2 + dv^2 \tag{4}$$

The embedding of a $d-1$ dimensional unit hypersphere in a d dimensional space follows from this equation. The surface of the unit hypersphere can be implicitly defined by the constraint

$$\|\boldsymbol{z}_i\|^2 = 1, \quad i = 1, \cdots, N \tag{5}$$

For elliptic manifolds, the arc length between two vectors defines a geodesic distance - a great circle that joins the points on the surface. The geodesic provides the means for computing distances on curved surfaces. We can simply derive its equation on a unit radius surface by noting that $\boldsymbol{z}_i^T \boldsymbol{z}_j = \cos\theta_{ij}$ which implies $\theta_{ij} = \arccos(\boldsymbol{z}_i^T \boldsymbol{z}_j)$. Therefore, the geodesic distance between vectors $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ is

$$d_{ij}^g = \theta_{ij} = \arccos(\boldsymbol{z}_i^T \boldsymbol{z}_j) \tag{6}$$

2.2 Spherical Dissimilarity Matrix

We begin by letting $\boldsymbol{X}$ to be the set of observations in a d-dimensional Euclidean space with vector elements $\{\boldsymbol{x}_i\}_{i=1}^N$. The main idea is to have an approach where the angle information between any pair of row vectors in $\boldsymbol{X}$ is easily extracted while preserving their neighborhood relations. Such neighborhood relations can be explained in terms of meaningful similarities of vectors in a space (meaningful in the sense that similar vectors when embedded, will remain close to each other while dissimilar vectors will be far apart). In the context of this paper, the similarity between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from $\boldsymbol{X}$ can be explained as follows: if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are sufficiently similar, the distance between them will be small. A suitable distance similarity measure can be found in the original space and an embedding algorithm can be proposed to map such relations onto a spherical surface. The algorithm can then compute a spherical distance similarity (herein the geodesic

distance) between the corresponding maps of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The corresponding maps are defined by the row vector elements of the spherical positional matrix $\boldsymbol{Z}$, meaning we would like to map $\boldsymbol{x}_i$ to $\boldsymbol{z}_i$ and $\boldsymbol{x}_j$ to $\boldsymbol{z}_j$ while minimizing the difference between Euclidean similarity and their corresponding spherical similarity which is a function of the unknown vectors $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$.

We define $\boldsymbol{W} = [d^e_{ij}]$ to be the high dimensional space similarity matrix, where d^e_{ij} is the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. We proceed to find an approximate spherical manifold similarity matrix $\boldsymbol{Z}\boldsymbol{Z}^T$ by finding the matrix $\boldsymbol{Z}^\star$ whose row vector elements $\{\boldsymbol{z}_i\}_{i=1}^N$ satisfy an optimization problem with the constraint in equation 5.

$$\boldsymbol{Z}^\star = \arg \min_{\boldsymbol{Z}, \|\boldsymbol{z}_i\|=1} |\boldsymbol{Z}\boldsymbol{Z}^T - \hat{\boldsymbol{W}}| \tag{7}$$

where $\hat{\boldsymbol{W}}$ is the normalized version of $\boldsymbol{W}$ with elements given by $w_{ij} = \cos(d^e_{ij})$. The solution to this problem can be efficiently derived from the eigendecomposition of $\hat{W}$, i.e. $\hat{\boldsymbol{W}} = \boldsymbol{\Lambda}\boldsymbol{U}\boldsymbol{\Lambda}^T$ and finding $\boldsymbol{Z}^\star$ to be

$$\boldsymbol{Z}^\star = U_{n\times m}\Lambda^{1/2}_{m\times m} \tag{8}$$

where m is experimentally chosen such that the elements of $U_{n\times m}$ corresponds to the largest m eigenvalues of $\Lambda_{m\times m}$. Algorithm 1 provides the corresponding steps of the spherical embedding approach. In the next section, we present some notation and definitions on the relationship between spherical surfaces and the local tangent planes.

Algorithm 1: Spherical Embedding Procedure

Input: $\boldsymbol{X} = [\boldsymbol{x}_i],\ i = 1, \cdots, N$, define $\boldsymbol{W}_{N\times N} = [w_{ij}] = [d^e_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j)]$ to be a matrix of Euclidean pairwise relationships for N d-dimensional observations.

Output: $\boldsymbol{Z}^\star$ a $N \times m$ spherical embedding matrix of maps (images) corresponding to N observations.

Solve the optimization problem for matrix $\boldsymbol{Z}$:

$$\boldsymbol{Z}^\star = \arg \min_{\boldsymbol{Z}, \|\boldsymbol{z}\|=1} |\boldsymbol{Z}\boldsymbol{Z}^T - \hat{\boldsymbol{W}}|$$

Solution is obtained by eigendecomposition, that is let $\hat{\boldsymbol{W}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, and set

$$\boldsymbol{Z}^\star = \boldsymbol{U}_{n\times m}\boldsymbol{\Lambda}^{1/2}_{m\times m}$$

2.3 Tangent Spaces and Manifolds

In this section, we present some definitions from Lie algebra - an algebraic structure whose main use is in studying geometric objects such as Riemannian manifolds. A sphere is known to be a compact Riemannian manifold that has some extra structure, e.g. a Riemannian metric, [2]. A *Riemannian metric* on a manifold M is a smoothly varying inner product $< \cdot, \cdot >$ on the tangent plane $T_{\boldsymbol{z}}M$ at each point $\boldsymbol{z} \in M$. The Riemannian distance between two points $\boldsymbol{z}_i, \boldsymbol{z}_j \in M$, denoted by $d^g(\boldsymbol{z}_i, \boldsymbol{z}_j)$, is defined as the minimum geodesic curve over all possible smooth curves between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. The study of Riemannian manifolds under Lie algebras has been observed to exhibit special local properties that enables simplification of computations by mapping to tangent space [2]. Such properties are established under the exponential map which relates objects on a spherical surface to their tangent spaces, *i.e* for each point $\boldsymbol{z}_i \in M$, the exponential map $Exp_{\boldsymbol{z}_i}$ transforms tangent space points to their images on M, while preserving their pairwise distance with reference to the base point $\boldsymbol{z}_i$. The mapping simplifies the difficult computation of geodesic distances on the surface of the manifold. We exploit these relations in more detail in the next section.

3 Spherical Metrics

3.1 Spherical Mahalanobis Distance

In this section, we extend the problem of classification to spherical manifolds by simply extending the ideas behind the framework of a nearest neighbor classifier. The K-nearest neighbor method is a non-parametric classifier that is memory-based. Assuming that the embedded training data, $\{\boldsymbol{z}_i\}_{i=1}^N$, is labeled and one is presented with a new unlabeled test vector $\boldsymbol{z}_0$, the main idea behind such a classifier is to first compute the pairwise distance between the test vector and each training set vector. One then proceeds to selecting the k nearest neighbors to the test vector and then choosing the majority label to be the label to assign the test vector. If one assumes that the feature vectors are real-valued, then the geodesic distance can be used to compute the distance between a given test point $\boldsymbol{z}_0$ and the potential nearest neighbor $\boldsymbol{z}_i$.

Given its simplicity, the K-nearest neighbor method has been applied with success in many classification problems. It is often successful when each class has many possible prototypes, and the decision boundary is very irregular. It is a classifier whose properties can be easily extended to non-Euclidean geometries. Our goal is to extend the use of K-nearest neighbor methods to spherical manifolds. The proposed metric rules are based on spherical manifold geometry and will incorporate the structure of the distribution of data around a given point. Incorporating structure into a classifier in Riemannian manifolds is known to be a very difficult task and still remains an open research problem. However, by making use of the exponential maps and log maps to be briefly defined in the next section, we are able to propose intuitive and simple structured spherical K-nearest neighbor metrics.

Exponential and Log Maps originate from Lie algebra as inverse relations to denote the notation of mapping points from $T_{\boldsymbol{z}_i}M$ onto the manifold M and mapping points from the manifold to the tangent space at point $\boldsymbol{z}_i$. The notation does not imply the traditional Log and Exponential functions. The exponential map takes a point $\boldsymbol{x} \in T_{\boldsymbol{z}_i}M$ on the tangent space to a point $\boldsymbol{z}_j$ on the sphere as follows:

$$\boldsymbol{z}_j = \boldsymbol{z}_i \cos\theta + \frac{\sin\theta}{\theta}\boldsymbol{x} \tag{9}$$

while the $Log_{\boldsymbol{z}_i}$ map of a point $\boldsymbol{z}_j$ on the sphere in the tangent space is defined as

$$\boldsymbol{x} = \frac{\theta}{\sin\theta}(\boldsymbol{z}_j - \boldsymbol{z}_i \cos\theta) \tag{10}$$

Note that we are reusing $\boldsymbol{x} \in T_{\boldsymbol{z}_i}M$ to denote the Log map transformed $\boldsymbol{z}_j$ spherical unit vectors onto the tangent space and these tangent plane vectors should not be confused with the original row vectors of observations in matrix $\boldsymbol{X}$. We can now consider extending the Mahalanobis distance to what we will call the spherical Mahalanobis distance by using the notion of a covariance matrix extended to spherical spaces. A Mahalanobis measure of points on a manifold M can now be defines as a distance between a random point $\log_{\boldsymbol{z}_i}\boldsymbol{z}_j \sim \mathcal{N}(\overline{\log_p s}, \boldsymbol{\Sigma}_{(\log_{\boldsymbol{z}_i}\boldsymbol{z}_j, \log_{\boldsymbol{z}_i}\boldsymbol{z}_j)})$ and a (deterministic) point $\log_{\boldsymbol{z}_i}\boldsymbol{z}_i$. Keeping in mind that $\log_{\boldsymbol{z}_i}\boldsymbol{z} : M \to T_{\boldsymbol{z}}M$, this can be defined by

$$d_m(\boldsymbol{z}_i, \boldsymbol{z}_j) = (\log_{\boldsymbol{z}_i}\boldsymbol{z}_j - \log_{\boldsymbol{z}_i}\boldsymbol{z}_i)^T \boldsymbol{\Sigma}^{-1}(\log_{\boldsymbol{z}_i}\boldsymbol{z}_j - \log_{\boldsymbol{z}_i}\boldsymbol{z}_i) \tag{11}$$

$$= (\log_{\boldsymbol{z}_i}\boldsymbol{z}_j)^T \boldsymbol{\Sigma}^{-1}_{(\log_{\boldsymbol{z}_i}\boldsymbol{z}_j, \log_{\boldsymbol{z}_i}\boldsymbol{z}_j)}(\log_{\boldsymbol{z}_i}\boldsymbol{z}_j) \tag{12}$$

The base point $\boldsymbol{z}_i$ on the manifold maps to $\log_{\boldsymbol{z}_i}\boldsymbol{z}_i$ which is the origin of the tangent space and hence the simplification leading to equation (12). The choice of a class label is picked as the class with a majority presence on the set of K spherical nearest neighbors. $\boldsymbol{\Sigma}_{(\log_{\boldsymbol{z}_i}\boldsymbol{z}_j, \log_{\boldsymbol{z}_i}\boldsymbol{z}_j)}$ is simply computed in the local tangent space of the base point.

The spherical Mahalanobis distance does incorporate structure into the metric. However, in many applications when the nearest-neighbor classification is carried out in a high dimensional feature space (i.e. hyperspherical surfaces), the nearest neighbors of a point can be very far away, causing bias and degrading the performance of the voting rule [9]. Such challenges call for considering an adaptive metric to be used in spherical nearest neighbor classification so that the resulting neighborhoods around a test point are able to stretch out in directions for which the class probabilities don't change significantly. An extension of such an approach to spherical manifolds is presented in the next section.

3.2 Spherical Discriminant Adaptive Nearest-Neighbor

In many high-dimensional problems, the nearest neighbor of a point can be very far away, causing bias and degrading the performance of the classification rule.

This problem was addressed for Euclidean spaces in Tibshirani and Hastie [10], where a *discriminant adaptive nearest-neighbor*(Dann) metric was presented. Accordingly, at each test point, a neighborhood of say 50 points is formed, and the class distribution among the points is used to decide how to deform the neighborhood, meaning to adapt the rule or the metric. The adapted metric is then used in a nearest-neighbor rule at the query point. This process results in potentially different metrics for each query point based on the distribution of label boundaries near the test point. This locally discriminative procedure only demands that information contained in the local within-and between-class covariance matrices is all that is needed to determine the optimal shape of the neighborhood.

An extension of this metric to spherical manifolds is simplified by taking advantage of the log-exponential mappings introduced earlier. Using the log-exponential mappings, we again choose two points $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, on the spherical manifold and define their tangent space positions as

$$\boldsymbol{x} = \log_{\boldsymbol{z}_i} \boldsymbol{z}_j, \quad \boldsymbol{x}_0 = \log_{\boldsymbol{z}_i} \boldsymbol{z}_i \tag{13}$$

Using the definition from the previous section, we know that any tangent space point $\boldsymbol{x}$ and tangent origin $\boldsymbol{x}_0$ take on coordinates on the spherical curved manifold as

$$\boldsymbol{z}_j = Exp_{\boldsymbol{z}_i}\boldsymbol{x}, \quad \boldsymbol{z}_i = Exp_{\boldsymbol{z}_i}\boldsymbol{x}_0 \tag{14}$$

The tangent space is locally defined around $\boldsymbol{x}_0$, as such computing of the distance from the tangent space origin to any other vector $\boldsymbol{x}$, entails computing for the norm of $\boldsymbol{x}$.

The *spherical discriminant adaptive nearest-neighbor* metric at a query point $\log_p p$ is defined by

$$d_{sd}(\boldsymbol{z}_i, \boldsymbol{z}_j) = (\log_{\boldsymbol{z}_i} \boldsymbol{z}_j - \log_{\boldsymbol{z}_i} \boldsymbol{z}_i)^T \boldsymbol{\Sigma} (\log_{\boldsymbol{z}_i} \boldsymbol{z}_j - \log_{\boldsymbol{z}_i} \boldsymbol{z}_i) \tag{15}$$

The expression in (15) can be rewritten using the mappings in equation (13) as

$$\begin{aligned} d_{sd}(\boldsymbol{x}, \boldsymbol{x}_0) &= (\boldsymbol{x} - \boldsymbol{x}_0)^T \boldsymbol{\Sigma}_{(\boldsymbol{x}_0, \boldsymbol{x}_0)} (\boldsymbol{x} - \boldsymbol{x}_0) \\ &= \boldsymbol{x}^T \boldsymbol{\Sigma}_{(\boldsymbol{x}_0, \boldsymbol{x}_0)} \boldsymbol{x} \end{aligned} \tag{16}$$

This simplification comes as a result of $\boldsymbol{x}_0$ being the origin of the tangent space $T_{M_{\boldsymbol{z}_i}}(\boldsymbol{z}_j)$. $\boldsymbol{\Sigma}_{\boldsymbol{x}_0, \boldsymbol{x}_0}$ in equation (16) is defined by

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_0, \boldsymbol{x}_0} = S_w^{-1/2} \{ S_w^{-1/2} S_B S_w^{-1/2} + \epsilon I \} S_w^{-1/2} \tag{17}$$

S_w is the pooled within-class covariance matrix

$$\boldsymbol{S}_w = \sum_{j=1}^{J} \pi_j S_{wj}$$

and S_B is the between class covariance matrix

$$\boldsymbol{S}_B = \sum_{j=1}^{J} \pi_j (\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})^T$$

with $\pi_j = \frac{N_j}{N}$ and $\bar{\boldsymbol{x}}_j = \frac{1}{N_j} \sum_{\boldsymbol{x} \in \mathcal{C}_j} \boldsymbol{x}$, where N_j is the number of samples in class C_j so that the within class C_j covariance matrix is computed as

$$\boldsymbol{S}_{wj} = \frac{1}{N_j} \sum_{n=1}^{N_j} (\boldsymbol{x} - \bar{\boldsymbol{x}}_j)(\boldsymbol{x} - \bar{\boldsymbol{x}}_j)^T \tag{18}$$

where $\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i$,

The parameter ϵ with value set to 1 rounds the neighborhood from an infinite strip to an ellipsoid, so as to avoid using points far away from the query point. `sphDann` involves choosing initial $k_{\boldsymbol{x}_0}$ nearest points to $\boldsymbol{x}_0$ to form the set $\boldsymbol{X}_{\boldsymbol{x}_0}$ in the local tangent space $T_{\boldsymbol{z}}M$. This is carried out by applying the spherical Mahalanobis distance measure of equation (12) (a geodesic distance measure can be used as well). The $k_{\boldsymbol{x}_0}$ points are chosen to determine the distribution of class labels around the test point $\boldsymbol{x}_0$. We experimentally observed that $k_{\boldsymbol{x}_0} = 60$ nearest points present enough samples for adapting the metric in the neighborhood of the query point. We could also consider this number to be a parameter obtained by cross validation methods. With the set $\boldsymbol{X}_{\boldsymbol{x}_0}$ determined, equation (17) can be computed. The second part involves using the `sphDann` metric in a K-nearest neighbor rule at $\boldsymbol{x}_0$. Note that the aim is to have the neighborhood of a query point stretched in the direction that coincides with the linear discriminant boundary of the classes. It is the direction in which class probabilities change the least.

4 Related Work

For comparison purposes, we consider two methods that compute the lower dimensional representation of data based on the eigenvectors of the embedding space. The lower dimensional space by both techniques are based on weighted graph algorithms. A graph is defined by $G = (V, E, \boldsymbol{W})$ with V being the set of vertices or nodes to represent observed samples in $\boldsymbol{X}$, E is the set of edges connecting nearby points to each other while $\boldsymbol{W}$ denotes the symmetric weight matrix between the connected vertices. Thus $w_{ij} \in \boldsymbol{W}$ is constrained to be a positive value weighting the edge between vertex i and j. Elements of $\boldsymbol{W}$ are usually computed through a pairwise distance function or kernel function as will be defined shortly. If $w_{ij} = 0$, then vertices i and j are not connected.

Definition 1. *(Kernel): A kernel* $k : \boldsymbol{X} \times \boldsymbol{X} \rightarrow \mathbb{R}$ *on a data set* $\boldsymbol{X}$ *is a function that defines edge weights for matrix* W *in the weighted graph. It has the following properties:*

- *symmetric:* $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i)$
- *positivity preserving:* $k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$
- *represents similarity between points in* $\boldsymbol{X}$

A common kernel function for computing weights w_{ij}'s that capture neighborhood relations (a very important property to preserve under embedding data) is the unnormalized Gaussian function

$$w_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\{\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\} \tag{19}$$

We can also define a diagonal matrix $\boldsymbol{D}$, whose elements d_{ij}'s are computed as $d_{ii} = \sum_{j=1}^{N} w_{ij}$ denoting the sum of edge weights corresponding to vertices that are incident to vertex i in the graph. To study the neighborhood relationship of embedded data points, graph based embedding algorithms differ mostly in how they exploit the spectral properties of the graph Laplacian $\boldsymbol{L}$, defined by

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \tag{20}$$

with $\boldsymbol{D}$ and $\boldsymbol{W}$ as defined above. We briefly discuss the main ideas behind two widely used graph based techniques: the diffusion maps and the locally preserving projections.

4.1 Diffusion Maps

The diffusion maps approach is based on the graph Laplacian which is a central tool in spectral graph theory. The method is capable of finding meaningful geometric descriptions of data sets when the observed samples are non-uniformly distributed and based on the notion of kernels [5]. The approach provides a new motivation for normalized graph Laplacian which relates to diffusion distances. Diffusion distances give different multiscale geometries depending on how often the random walk matrix is iterated. The normalized Laplacian is constructed as follows:

$$\begin{aligned} \boldsymbol{L}_{diff} &= \boldsymbol{D}^{-1}\boldsymbol{L} \\ &= \boldsymbol{I} - \boldsymbol{P} \end{aligned} \tag{21}$$

where $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}$ is defined to be the diffusion operator and each entry $p_{ij} = w_{ij}/d_{ii}$ is viewed as the transition kernel of the Markov chain on the graph G. Thus p_{ij} defines the transition probability of going from state i to j in one time step. Therefore $\boldsymbol{P}$ defines the entire Markov chain matrix. The main goal of this approach is such that the transition probabilities defined by $\boldsymbol{P}$ reflect the local geometry of the data. Thus, we need to establish the spectral properties of the resulting Markov chain (more precisely its matrix eigenvalues and eigenvectors). This is achieved by defining the diffusion distance as follows:

Definition 2. *A family of diffusion distances* $\{d_t\}_{t\in\mathbb{N}}$ *at time* t *is defined as:* $d_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq \|p(\boldsymbol{x}_k, t|\boldsymbol{x}_i) - p(\boldsymbol{x}_k, t|\boldsymbol{x}_j)\|_w^2 = \sum_{\boldsymbol{x}_k}(p(\boldsymbol{x}_k, t|\boldsymbol{x}_i) - p(\boldsymbol{x}_k, t|\boldsymbol{x}_j))^2 w(\boldsymbol{x}_k)$ *where* $p(\boldsymbol{x}_k, t|\boldsymbol{x}_i)$ *is the probability that the random walk that started at* $\boldsymbol{x}_i$ *arrived at* $\boldsymbol{x}_k$ *after* t *steps.*

The above definition is intuitive in the sense that if two embedded points are closer then there should be more paths of larger weights w_{ij}'s connecting them. From this definition, taking the difference of the two posterior values inside the brackets and squaring the result gives a probabilistic distance measure.

Another important discovery from the diffusion maps approach (as presented in [5]) is that $d_t(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be computed from the eigenvectors ψ_l and eigenvalues λ_l of the probability matrix $\boldsymbol{P}$ as follows:

$$d_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\sum_{l \geq 1} \lambda_l^{2t} (\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j))^2)^{1/2} \tag{22}$$

This important result is derived from spectral theory in the Hilbert space and also makes use of the fact that the eigenfunctions of the probability matrix $\boldsymbol{P}$ are orthonormal. Thus, the diffusion map $\boldsymbol{\psi}_t(\boldsymbol{x}) : \boldsymbol{X} \rightarrow \boldsymbol{Z}$ defined by

$$\boldsymbol{\psi}_t \triangleq \begin{pmatrix} \lambda_1^t \psi_1(\boldsymbol{x}) \\ \lambda_2^t \psi_2(\boldsymbol{x}) \\ \vdots \\ \lambda_m^t \psi_m(\boldsymbol{x}) \end{pmatrix} \tag{23}$$

embeds the data into the Euclidean space $\boldsymbol{Z} = \mathbb{R}^m$ in which the distance is approximately the diffusion distance

$$\|\psi_t(\boldsymbol{x}_i) - \psi_t(\boldsymbol{x}_j)\| = d_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) + \mathcal{O}(t, m) \tag{24}$$

where the big $\mathcal{O}$-notation denotes that the embedding space is approximated to a relative accuracy. The space is described by the eigenvectors of the embedding while the smoothness of the mapping is simply achieved by scaling each eigenvector by its corresponding eigenvalue as shown in equation (23). More details on the diffusion maps can be found in Coifman and Lafon [5].

4.2 Locality Preserving Projections

Locality preserving projections (LPP) is a linear approximation of the nonlinear Laplacian embedding method [7]. The algorithm is based on constructing the adjacency graph of the input features with an edge between feature $\boldsymbol{x}_i$ and feature $\boldsymbol{x}_j$ if $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 < \xi$, with $\xi \in \mathbb{R}$, a small number defining the neighborhood to point $\boldsymbol{x}_i$. The weight w_{ij}, if node i is connected to node j, is computed using equation 19. The embedding space is then obtained by solving for the eigenvectors and eigenvalues of

$$\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{z} = \lambda \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{z} \tag{25}$$

where both the diagonal matrix $\boldsymbol{D}$ and the graph Laplacian $\boldsymbol{L}$ are as defined in equation 20. In this formulation, the i^{th} column of $\boldsymbol{X}$ defines the vector $\boldsymbol{x}_i$. Let $\boldsymbol{v}_0, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_{m-1}$, be the solution to the eigenvector problem in equation (25),

ordered according to the eigenvalues $\lambda_0 < \lambda_1 < \cdots < \lambda_{k-1}$. Then, the embedding space can be defined as the mapping of each observation $\boldsymbol{x}_i$ as follows

$$\boldsymbol{x}_i \rightarrow \boldsymbol{z}_i = \boldsymbol{V}^T \boldsymbol{x}_i \tag{26}$$

with $\boldsymbol{V} = (\boldsymbol{v}_0, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_m)$, where $\boldsymbol{z}_i$ denotes the lower dimensional embedding position corresponding to $\boldsymbol{x}_i$.

5 Experimental Data

5.1 Colorado

The Colorado data set consists of the following four data sources [11] : (1) *Landsat MSS data* (four spectral data channels), (2) *Elevation data* (one data channel), (3) *Slope data* (one data channel), (4) *Aspect data* (one data channel). Each channel comprises an image of 135 rows and 131 columns, and all channels are spatially co-registered. There are ten ground-cover classes listed in Table 2. One class is water; the others are forest types. It is very difficult to distinguish among the forest types using Landsat MSS data alone since the forest classes show very similar spectral responses. Details on the sample size for each class are provided in Table 2.

5.2 Botswana Hyperion

Hyperion data with 9 identified classes of complex natural vegetation were acquired over the Okavango Delta, Botswana, in May 2001 [3]. The general class groupings include seasonal swamps, occasional swamps, and woodlands. Signatures of several classes are spectrally overlapped, typically resulting in poor classification accuracies. After removing water absorption, noisy, and overlapping spectral bands, 145 bands were used for classification experiments. The embedding and classification results are reported for all 9 classes. Table 2 provides the sample sizes for each class.

5.3 Kennedy Space Center (KSC)

Airborne hyperspectral data were acquired by the NASA AVIRIS sensor at 18-m spatial resolution over Kennedy Space Center during March 1996. Noisy and water absorption bands were removed, leaving 176 features for 13 wetland and upland classes of interest. Cabbage Palm Hammock (Class 3) and Broad Leaf/Oak Hammock (Class 6) are upland trees; Willow Swamp (Class 2), Hardwood Swamp (Class 7), Graminoid Marsh (Class 8) and Spartina Marsh (Class 9) are trees and grasses in wetlands. Their spectral signatures are mixed and often exhibit only subtle differences. Results for all 13 classes and for these "difficult" classes are reported for the spherical embedding and classification experiments. Samples sizes for each class are presented in Table 2.

Table 2. Experimental Data: class labels and number of labeled samples

Colorado	Botswana	Kennedy Space Center
c1 Water (603)	c1 Water (158)	c1 Scrub (761)
c2 Colorado Blue Sprice (112)	c2 Floodplain (228)	c2 Willow swamp (243)
c3 Mountane/Subalpine meadow (87)	c3 Riparian (237)	c3 Cabbage hamm (256)
c4 Aspen (140)	c4 Firescar (178)	c4 Cabbage palm(252)
c5 Ponderosa Pine (224)	c5 Island interior (183)	c5 Slash pine (161)
c6 Ponderosa Pine/Douglas Fir (314)	c6 Woodlands (199)	c6 Oak (229)
c7 Engelmann Spruce (294)	c7 Savanna (162)	c7 Hardwood swamp (105)
c8 Douglas Fir/White Fir (76)	c8 Short mopane (124)	c8 Graminoid marsh (431)
c9 Douglas Fir/Ponderosa Pine/Aspen (50)	c9 Exposed soils(111)	c9 Spartina marsh (520)
c10 Douglas Fir/White Fir/Aspen (99)		c10 Cattail marsh (404)
		c11 Salt marsh (419)
		c12 Mud flats (503)
		c13 Water (927)

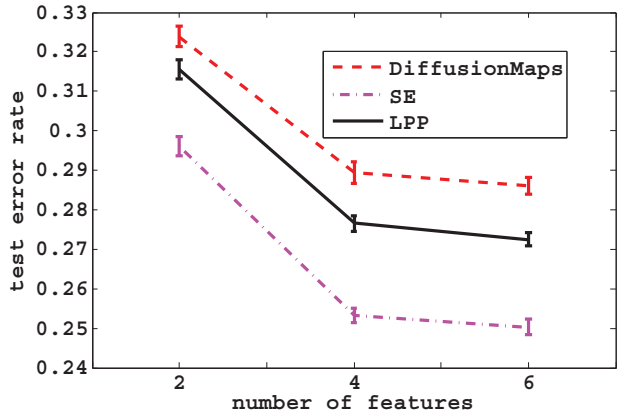

Fig. 1. Mean ± one standard error of misclassification error for `SE-sphDann`, `diffusion maps-Dann`, `LPP-Dann` 5-nearest neighbors on the Colorado data's embedded space

6 Experimental Results

To evaluate the spherical mapping, we consider `SE` as a preprocessing phase for pattern classification and also as a visualization tool. In many hyperspectral data applications, dimensionality reduction is considered useful as a feature extraction mechanism to speed up other algorithms. In this study, we compare the coordinates of features that were obtained from `SE` to those of `diffusion maps` and `LPP` for benchmark classification problems in land cover studies. All labeled image pixels were randomly sampled to provide 60% training and 40% testing samples, with a repetition over 20 runs. Each random sample is fixed and used in all three manifold learning methods so as to eliminate any variation when comparing output results. Due to space limitation, we only report on results that we obtained from using our proposed `sphDann` classifier in conjunction with `SE`. For `LPP` and `diffusion maps` embedding, we make use of the original `Dann` nearest neighbor classifier, [10]. For both `sphDann` and `Dann` we set $k_{\boldsymbol{x}_0} = 60$ nearest neighbor to define the structure of the metric and then chose the class label based on $K = 5$ nearest neighbors to the test point. The embedding parameters for `LPP` are set as $K = 12$ for the set of points to localize the neighborhood for each point and $\sigma = 1$ for the variance in the kernel function. The parameters for the `diffusion maps` approach were set to $t = 4$ for the time step and $\sigma = 1$. More experimental results can be found in the shorter release of this manuscript [12].

6.1 Manifold Classification

Figure 1 shows the results of `SE-sphDann`, `LPP-Dann` and `diffusion maps-Dann` when applied to a multispectral image. It is evident that for fewer number of features considered, `SE` achieves significantly lower error rates than `diffusion maps` and `LPP` on the test queries. The results suggest that the spherically extended `Dann` classifier effectively provides means to evaluate spherically embedded data.

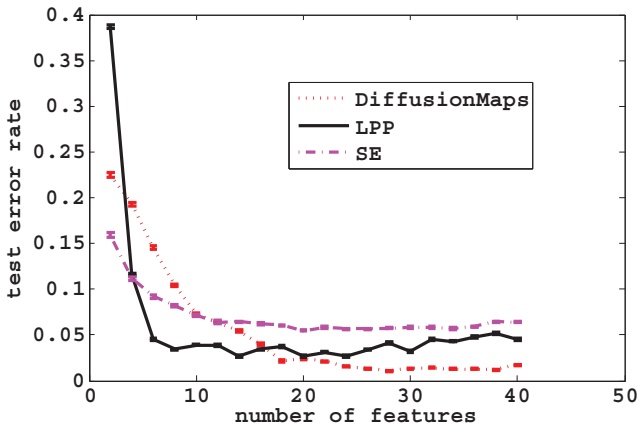

Fig. 2. Mean ± one standard error of misclassification error for `SE-sphDann`, `diffusion maps-Dann`, `LPP-Dann` 5-nearest neighbor on the Botswana data's embedded space

Figure 2 shows the results of `SE-sphDann`, `LPP-Dann` and `diffusion maps-Dann` when applied to the Botswana hyperspectral data. It is evident that for fewer number of features considered, `SE` achieves significantly lower error rates than `diffusion maps` on the test queries. However, there is a crossover in error rates as the number of features increases. For twelve features and beyond, it seems that `SE` lacks the ability to extract further useful information. Its performance tends to saturate while `diffusion maps` continues to achieve lower error rates due to the random walk properties of the diffusion matrix that continues to extract important information as more and more features are made available. On the other hand, `LPP` does well in line with `SE` up to a point where an increase in features does not seem to change the error rates. The behavior emanating from `SE` and `LPP` can be attributed to the locality preserving properties of the methods. Once the neighborhood structure has been constructed, addition of features that are further away does not seem to influence locality structure.

Figure 3 shows the results of `SE-sphDann`, `LPP-Dann` and `diffusion maps-Dann` when applied to the KSC hyperspectral data. It is evident that for fewer number of features considered, `SE` achieves significantly lower error rates than `diffusion maps` and performs closely to `LPP` on the test queries. However, there are crossovers in error rates as the number of features is increased. `SE`'s performance tends to saturate while `diffusion maps` continues to achieve lower error rates. On the other hand, `LPP` does well in line with `SE` up to a point where an increase in features does not seem to change the error rates. The behavior emanating from `SE` and `LPP` can be attributed to the locality preserving properties of the methods. Once the neighborhood structure has been constructed, the addition of features that are further away does not seem to influence locality structure.

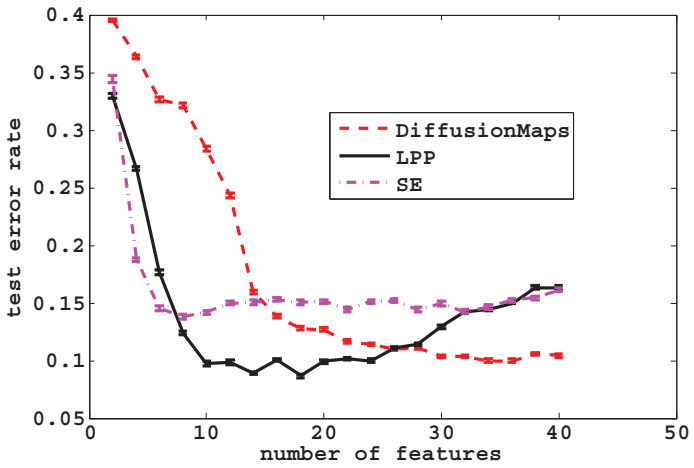

Fig. 3. Mean ± one standard error of misclassification error for `SE-sphDann`, `diffusion maps-Dann`, `LPP-Dann` 5-nearest neighbor on the KSC data's embedded space

6.2 Manifold Visualizations

A human assessment on the structure of the embedded data can be achieved by plotting the lower dimensional coordinates of data so as to validate if any local and global characteristics are present (e.g. grouping of similar points and the emerging clusters of data). A comparison of the manifold representation obtained from the three methods discussed in this study are shown in Figure 4, Figure 5 and Figure 6, respectively. In all plots what we observed (including experiments not shown here) is that `diffusion maps` and `LPP` are very sensitive to the parameter values assigned for each data set. The diffusion maps method seemed to work well with large values of sigma. `LPP` tends to compress similar

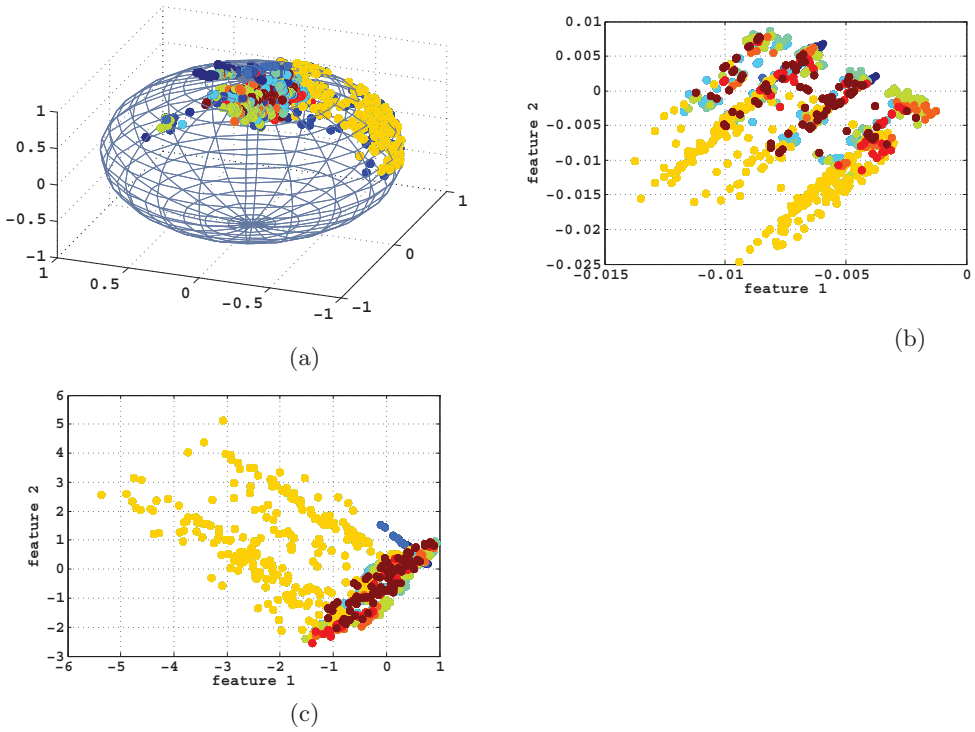

Fig. 4. Colorado Data. (a) 2D Spherical Embedding, (b) 2D `LPP` Embedding and (c) 2D `diffusion maps` Embedding. Each color coding denotes a different class label.

clusters towards a single point. On the hand, `SE` tend to recover a better local and global structure of data on few dimensions. This makes `SE` a suitable candidate for visualization tasks. The qualitative results from the classification plots confirm what the visualization plots are showing - that is, `SE` achieved lower error rates with fewer features than `LPP` and `diffusion maps`. However, we note that when the observed data has many classes, the resulting embedding under `SE` tend to crowd all embedding positions to the center of the lower dimensional space. This causes a degradation of cluster formation and it affects the classification accuracy. Our future work intends to substitute the matrix difference objective function of `SE` with a probability distribution distortion measure. This can be achieved by making use of the Kullback Leibler divergence measure - an information theoretic approach for determining the difference between two distributions P_h and Q_l. The idea is to assume a high dimensional distribution P_h over the observation space and set the goal of finding a lower dimension space distribution Q_l as a function of the embedding positions $\boldsymbol{z}$'s. Such an approach may allow us to impose prior knowledge onto the embedding positions in the form of a prior distribution which may address the crowding problem observed

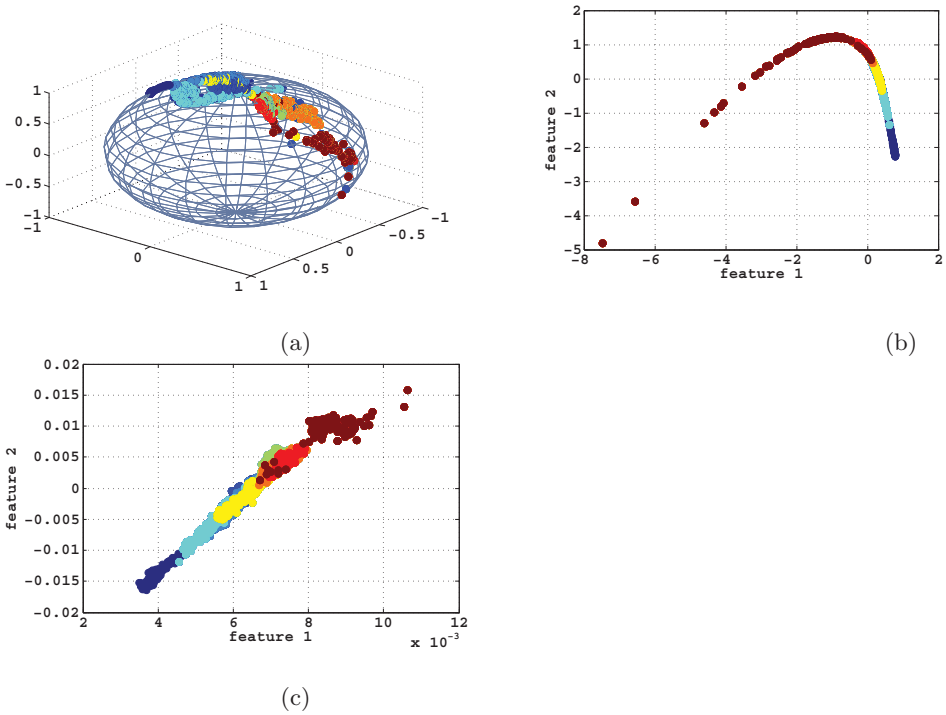

Fig. 5. Botswana Data. (a) 2D Spherical Embedding, (b) 2D `LPP` Embedding and (c) 2D `diffusion maps` Embedding. Each color coding denotes a different class label.

in `SE`. A second challenge that is still an open research problem is the investigation on approaches to reduce the complexity of `SE` from $\mathcal{O}(n^2)$ to a log linear time $\mathcal{O}(n \log n)$, which would be a speed in the optimization algorithm.

7 Conclusions

In this paper, we introduced the idea of mapping remote sensing image pixels onto a nonlinear spherical coordinate system. We cited data that was captured by high resolution sensors and as a result contains a high degree of nonlinearities. The direct result of such nonlinearities is a fundamental limit on the ability to discriminate different classes, for instance, data with spectrally similar vegetation. We demonstrated that a lower dimensional spherical space coordinate representation coupled with a spherical nearest neighbor classifier provides a system that achieves reduced error rates for remote sensing data. For few features, the proposed approach obtained better discriminative accuracy when compared to `diffusion maps` and also performs competitively with `LPP`. Indications from this study suggest that multispectral features when treated as embedding on a nonlinear surface do provide a comparative platform for land cover classification.

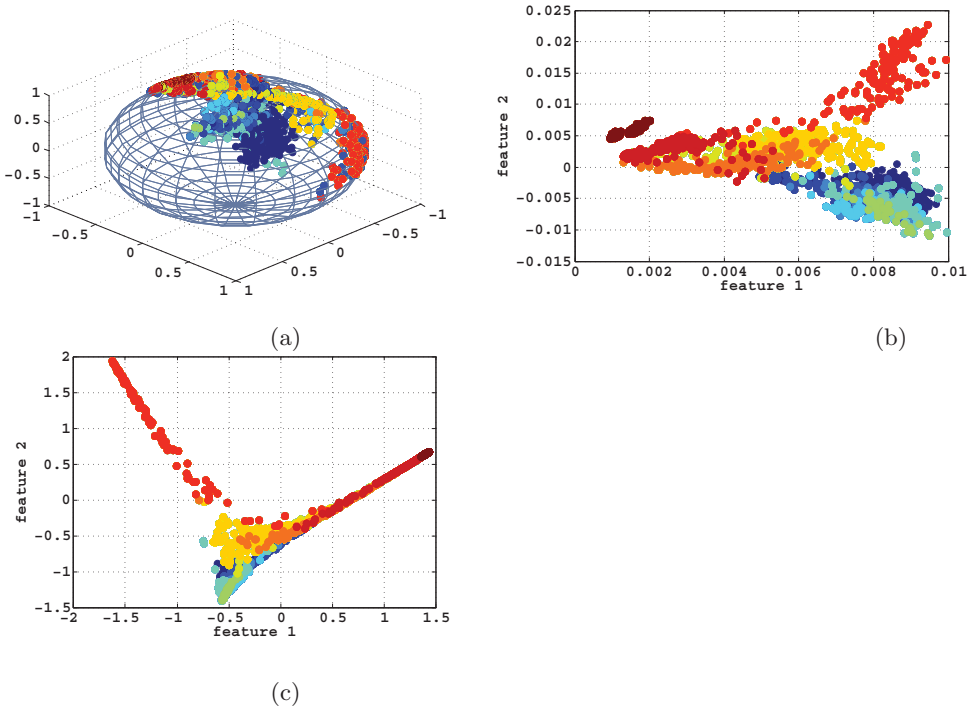

Fig. 6. Kennedy Space Center Data. (a) 2D Spherical Embedding, (b) 2D `LPP` Embedding and (c) 2D `diffusion maps` Embedding. Each color coding denotes a different class label.

References

1. Jolliffe, I. T.: Principal Component Analysis, Springer-Verlag (1986)
2. Kirillov, A.: Introduction to Lie Groups and Lie Algebras. Cambridge University Press (2008)
3. Crawford, M. M., Kim, W., Li, M.: Exploring Nonlinear Manifold Learning for Classification of Hyperspectral Data. Transactions on Optical Remote Sensing (2011)
4. Cox, M. A. A., Cox, T. F.: Multidimensional Scaling. Chapman and Hall (2001)
5. Coifman, R., Lafon, S.: Diffusion maps. Applied and Computational Harmonic Analysis: Special issue on Diffusion Maps and Wavelets 21, 5–30 (2006)
6. Roweis, S. T., Saul, L. K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
7. He, X., Niyogi, P.: Locality Preserving Projections. In: Advances in Neural Information Processing Systems (2003)
8. Wilson, R. C., Duin,R. P. W., Hancock, E. R., Pekalska,E.: Spherical embeddings for non-euclidean dissimilarities.In: Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 1903–1910. (2010)
9. Hastie, T., Friedman, J., Tibshirani,R. : The Elements of Statistical Learning: Data Mining,Inference, and Prediction. Springer, New York (2009)

10. Tibshirani R., Hastie, T.: Discriminant adaptive nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(6) (1996)
11. Benediktsson, J., Ersoy, O. K., Swain, P.: Neural network approaches versus statistical methods in classification of multisource remote sensing data. IEEE Transactions on Geoscience and Remote Sensing 28(4) (1990)
12. Lunga, D. , Ersoy, O. K.: Spherical nearest neighbor classification: Application to hyperspectral data. In: P. Perner (Ed.) MLDM2011, LNAI, vol. 6871, pp. 170-184, Springer Verlag, Heidelberg (2011)

Vitae

Dalton Lunga received his B.Eng. and M.S. degrees in electrical engineering from the University of Johannesburg and the University of Witwatersrand, South Africa, in 2004 and 2006, respectively. He is currently pursuing a PhD in Electrical and Computer Engineering at Purdue University. His research interests include manifold learning and differential geometry, information retrieval for remote sensing images, and statistical signal processing.

Okan K. Ersoy (M86-SM90-F00) received a B.S.E.E. degree from Robert College, Istanbul, Turkey, in 1967, and M.S. Certificate of Engineering, M.S., and Ph.D. degrees from the University of California, Los Angeles, 1971, and 1972, respectively. He is currently a Professor in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. His research interests include remote sensing, statistical and computational intelligence, digital signal/image processing and recognition, imaging, bioinformatics, diffractive optics and phased array systems. He has published over 250 papers in his research. He is a holder of four patents. Dr. Ersoy is a Fellow of the Optical Society of America and a fellow of IEEE.