Summer 2014

High Dimensional Data Set Analysis Using a Large-Scale Manifold Learning Approach

Loc Tran
Old Dominion University

HIGH DIMENSIONAL DATA SET ANALYSIS USING A LARGE-SCALE MANIFOLD LEARNING APPROACH

by

Loc Tran
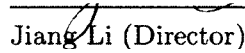B.S. May 2008, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
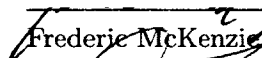Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
August 2014

Approved by:

Jiang Li (Director)

Frederic McKenzie (Member)

Dean Krusienski (Member)

Yaohang Li (Member)

ABSTRACT

HIGH DIMENSIONAL DATA SET ANALYSIS USING A LARGE-SCALE MANIFOLD LEARNING APPROACH

Loc Tran
Old Dominion University, 2014
Director: Dr. Jiang Li

Because of technological advances, a trend occurs for data sets increasing in size and dimensionality. Processing these large scale data sets is challenging for conventional computers due to computational limitations. A framework for nonlinear dimensionality reduction on large databases is presented that alleviates the issue of large data sets through sampling, graph construction, manifold learning, and embedding. Neighborhood selection is a key step in this framework and a potential area of improvement. The standard approach to neighborhood selection is setting a fixed neighborhood. This could be a fixed number of neighbors or a fixed neighborhood size. Each of these has its limitations due to variations in data density. A novel adaptive neighbor-selection algorithm is presented to enhance performance by incorporating sparse ℓ_1-norm based optimization. These enhancements are applied to the graph construction and embedding modules of the original framework. As validation of the proposed ℓ_1-based enhancement, experiments are conducted on these modules using publicly available benchmark data sets. The two approaches are then applied to a large scale magnetic resonance imaging (MRI) data set for brain tumor progression prediction. Results showed that the proposed approach outperformed linear methods and other traditional manifold learning algorithms.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

xii

Chapter 1

INTRODUCTION

It can be argued that human progression has always been linked to our ability to understand data. With the advancement of technology for data acquisition and data storage, the size of data sets are increasing exponentially [2]. In general, the hope is that if large-scale data could be exploited effectively, science would extend its reach, and technology would become more advanced and robust. Thus, there is a growing need for algorithms that are able to scale to larger and larger data sets. While there is great potential in deciphering large data sets, the challenge of processing large-scale data sets exists. The amount of data a single user c'an process is often limited to the power offered by a personal computer. In this dissertation, we focus on the nonlinear dimensionality reduction of a large data set using conventional hardware.

Nonlinear dimensionality reduction has become a popular topic in pattern recognition literature. For high dimensional data sets with continuous variables, it is often the case that the data points are arranged close to a manifold of much lower dimensionality than the original data set. This manifold is the intrinsic structure of the data. Thus, dimensionality reduction can also be referred to as manifold learning. Conventional methods such as principal component analysis are effective in finding linear manifolds such as a linear line or a flat hyper-plane. However, it is often the case that the manifold is nonlinear and have some sort of intrinsic curvature or irregular shape. For example, this occurs in applications for face pose recognition, handwriting recognition, and speech signals. The goal for nonlinear manifold learning methods is to identify the intrinsic shape of low dimensional data structures embedded in high dimensional data and to represent these nonlinear structures with a low dimensional linear model. Obviously, the nonlinear manifolds are much harder to compute than the linear manifolds. For many nonlinear methods, the local neighborhood of each data sample needs to be computed and a connectivity/similarity graph needs to be calculated which is difficult for large-scale data sets. Generally, manifold learning approaches require a spectral decomposition of the similarity matrix. This step is also computationally expensive.

Besides the computational complexity problem, another problem of traditional manifold learning methods is that the neighborhood selection is often rigid. Since local geometry is a large part of manifold learning, neighborhood selection is a critical part of many nonlinear dimensionality reduction techniques. Traditional methods use simple k-nearest neighbors as the neighborhood selection method. This would connect each point to a fixed number k of its nearest neighbors. An optimal value for k is oftentimes different at various parts of the manifold. An incorrect selection of k could cause the resulting manifold to incorrectly represent the intrinsic structure of the data.

In this dissertation, we address the challenge of applying manifold learning on a large data set. First, we present the Incremental Landmark approach which reduces the data set size through sampling and embedding. Sampling allows for exact eigen-decomposition of nonlinear manifold techniques on the smaller sampled set. We present an adaptive sampling method based upon local curvature information. Remaining points are reintroduced into the manifold using a neighborhood based embedding step. Also, since many nonlinear manifold approaches require neighborhood selection, we present an adaptive neighborhood selection method that is applied to the graph construction and embedding steps of the Incremental Landmark approach. The proposed neighborhood selection is based on sparse optimization of the ℓ_1-norm. The ℓ_1-norm based approaches replace the graph construction and embedding modules of the Incremental Landmark approach to give the final proposed method.

This dissertation is arranged as follows. Related work is discussed in Chapter 2. The related work will focus first on linear dimensionality reduction techniques before reviewing modern nonlinear dimensionality reduction approaches. This chapter contains a literature review on some techniques to address dimensionality reduction on large data sets and also the application of sparse learning on dimensionality reduction. In Chapter 3, the proposed approach for large scale nonlinear dimensionality reduction is introduced. The basic structure is presented as the Incremental Landmark method. A novel approach in neighborhood selection using the ℓ_1-norm is introduced to enhance the Incremental Landmark method. An analysis on the benefits of using the ℓ_1-based optimization is also discussed. The ℓ_1-norm enhanced method is validated on benchmark data sets in Chapter 4. Here, two components of the original proposed method are modified using ℓ_1-based neighbor selection and are tested individually with quantitative data sets from the UCI data set repository

and the results are compared with the standard Incremental Landmark method. The two modifications are combined to form the final proposed method and is also tested with benchmark image data sets used for face and object recognition. In Chapter 5, the Incremental Landmark method is applied to an MRI brain tumor data set for tumor identification and progression prediction. In Chapter 6, the ℓ_1-norm enhanced proposed method is applied to the same data MRI data set. An analysis of using the proposed approach versus using randomized singular value decomposition for large scale manifold learning is presented in Chapter 7. Finally, conclusions are summarized in Chapter 8.

1.1 CONTRIBUTIONS

The contributions of this work includes a novel, adaptive, neighbor-selection algorithm using the ℓ_1-norm. Also, the Isomap method is enhanced using the ℓ_1-based neighbor selection for graph construction. Likewise, local linear embedding is enhanced using the ℓ_1-based method. Local curvature variation sampling is introduced to adaptively sample data sets. Another contribution is an identification of a bridge between abnormal tumor tissue and normal tissue in MRI brain scans as discussed in Chapter 5.

Chapter 2

RELATED WORK

2.1 MANIFOLD LEARNING

The focus of this dissertation is on large-scale manifold learning. Before getting into large-scale application, it is important to talk about what is manifold learning. To go by the formal definition, manifold learning is an approach to non-linear dimensionality reduction that takes advantage of a geometric structure formed in a data set that exists in a lower dimensional space than the original data set. This lower dimensional structure is referred to as a manifold, thus the name manifold learning. Conceptually, this concept may be difficult to realize. For one, we live in a three dimensional world. Spaces greater than three or four dimensions is difficult to imagine physically. Here, two example manifolds are discussed to get a better understanding of what manifolds are.

First, imagine two towns, Town A and Town B, separated by a river. Now consider the distance by car between the two towns. Going in a straight line across the river would only be one mile but cars can't drive on water. The true driving distance is the length of roads between the two towns. Consider a bridge one mile away from both towns. The distance would be one mile to the bridge, one mile crossing the bridge, and one mile from the bridge to the opposite town resulting in a three mile driving distance. In this example, the setting is a two dimensional roadway. But the road is one dimensional since cars can only go back and forth. Thus, the manifold would be the one-dimensional road between the two towns in a two dimensional setting.

The second example is a set of face pose images of a single person shown in Figure 1. Now consider, the distance between the top left image and the bottom right image. These two images are shown side by side in Figure 2. In a computer vision standpoint, the two images are very dissimilar from each other. A difference image of these two would show that these images are very different. In this data set, however, a manifold exists between these two images. The manifold relates the images through rotation from a frontal face image to a profile view. Following
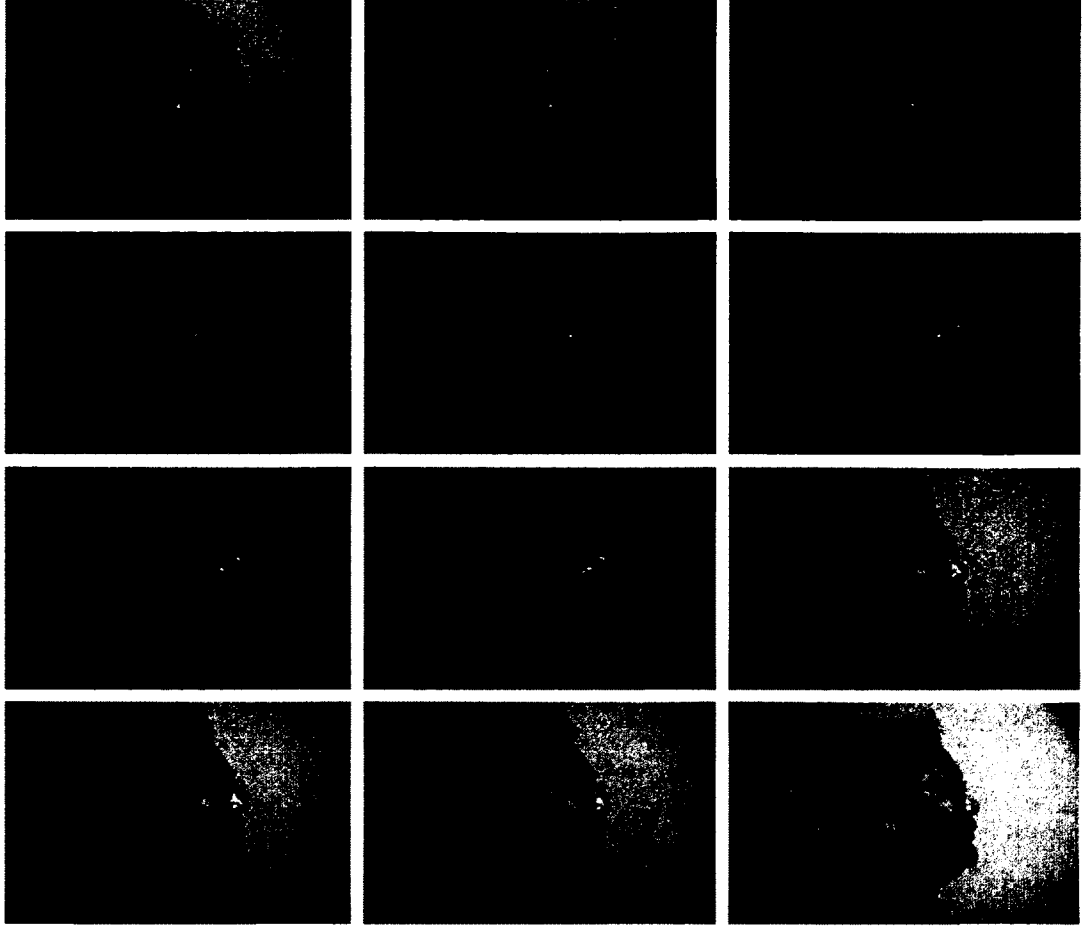
Figure 1: Example nonlinear face pose data set

the images in Figure 1 from left to right and top to bottom, each successive image is similar to the previous. Using this manifold, a connection could be made between the two images. The example shows a non-linear one-dimensional manifold embedded in a data set of 640×480 pixel images. The original data set was a 307,200 dimensional data set that was reduced to one. This example shows that large data sets could be reduced greatly if the manifold of the data could be found.

The goal of manifold learning is to produce a low dimensionality mapping of the data set while still retaining the relevant information found within the data. These techniques can be divided into two categories, linear and nonlinear. Linear techniques are much easier to compute but will fail if the data has a nonlinear manifold. On the other hand, nonlinear techniques are much more difficult to compute but will not always correctly unfold a nonlinear data set. As noted in the literature, there is
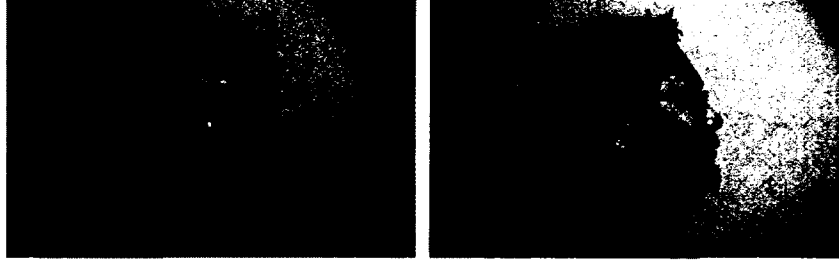
Figure 2: Example nonlinear data set

not one method that will outperform other methods in all cases [3]. In this section, various linear and nonlinear dimensionality reduction techniques will be reviewed.

2.1.1 LINEAR DIMENSIONALITY REDUCTION

Principle Component Analysis (PCA)

Principal component analysis (PCA) is a popular unsupervised linear dimensionality reduction technique [4, 5]. The method creates a low-dimensionality representation of the data that minimizes the variance in the data. In other words, PCA finds d orthogonal vectors that encompass the most variance in the data. Consider X is a $m \times n$ data matrix containing the m samples in n dimensions and M is an $m \times d$ matrix mapping that maximizes $M^T \text{cov} (X) M$ where $d \leq n$. Finding M can be done by solving the eigen-mapping problem shown below.

$$\text{cov} (X) M = \lambda M$$

The resultant M contains d orthogonal basis vectors spanning the data. These vectors are referred to in PCA as principal components, each with a corresponding eigenvalue $\lambda_{1...d}$. The first principal component will correspond to the largest eigenvalue. Similarly the second principal component will correspond to the second largest eigenvalue and so on such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. In this way, each successive principal component will cover less variance in the data set than the principal components before it. Dimensionality reduction can be achieved by keeping the first d principal components. The following principal components would have a less meaningful contribution to variations in the data and may be discarded. As this is a linear technique, it is unable to recognize a nonlinear structure.

2.1.2 NONLINEAR DIMENSIONALITY REDUCTION

Kernel PCA

Kernal PCA (KPCA), is an extension to traditional linear PCA that allows it to compute non-linear manifolds [6]. Kernel PCA computes the principal eigenvectors of a kernel matrix rather than the covariance matrix.

Kernel PCA computes the kernel matrix K of the data points x_i. The entries in the kernel matrix are defined by:

$$k_{ij} = -\frac{1}{2}\kappa\left(x_i, x_j\right)$$

where κ is a kernel function, which may be any function that gives rise to a positive-semidefinite kernel K [7]. Subsequently, the kernel matrix K is double-centered using the following modification of the entries

$$k_{ij} = -\frac{1}{2}\kappa\left(k_{ij} - \frac{1}{n}\sum_l k_{il} - \frac{1}{n}\sum_l k_{jl} + \frac{1}{n^2}\sum_{lm} k_{lm}\right)$$

The centering operation corresponds to subtracting the mean of the features in traditional PCA. As a result, the data in the feature space defined by the kernel function is zero-mean. Thus, the principal d eigenvectors v_i of the centered kernel matrix are computed. Now similar to traditional PCA, the eigenvectors of the covariance matrix a_i can now be computed. These eigenvectors are related to the eigenvectors of the kernel matrix v_i through the following relation [8]

$$a_i = \frac{1}{\sqrt{\lambda_i}}v_i$$

In order to obtain the low-dimensional data representation, the data is projected onto the eigenvectors of the covariance matrix a_i. The result of the projection is the low-dimensional data representation Y given by:

$$y_i = \left\{\sum_{j=1}^{n} a_1^{(j)}\kappa(x_j, x_i), \ldots, \sum_{j=1}^{n} a_d^{(j)}\kappa(x_j, x_i)\right\}$$

A disadvantage of kernal PCA is that a suitable kernel is difficult to determine, especially for high dimensional data sets. Kernel PCA has been successfully applied to face recognition [9], speech recognition [10], and novelty detection [11].

Isomap

The Isomap method reduces a high dimensional data set while preserving the geodesic distance between data points [12]. If the d-dimensional original input space is defined as X and the d'-dimensional Euclidean space Y as the lower dimensional manifold, then the problem can be summarized as finding a mapping of $f : X \mapsto Y$. Ideally, geodesic distance pairs in X will be equivalent to Euclidean distance pairs in Y. The Isomap method is a widely used nonlinear dimensionality reduction technique in many applications including wood inspection [13], visualization of biomedical data [14], and head pose estimation [15].

Isomap first constructs an k-nearest neighborhood graph $G_{n \times n}$ by connecting x_i to its k-nearest neighbors in X, where n is the number of data points in X. Based on G, Isomap then computes a similarity graph $D_G = \{d_G(i,j)\}$, where $d_G(i,j)$ is the geodesic distance between data points i and j computed from G. Geodesic distance can be obtained efficiently by using the Dijkstra's algorithm [3]. Once the similarity matrix D_G is established, the Isomap utilizes the traditional multidimensional scaling (MDS) technique to reduce dimensionality by minimizing the following cost function:

$$E = \|\tau (D_G) - \tau (D_Y)\|_{\ell_2}$$

Here, D_Y represents a matrix of Euclidean distances $\{d_Y (i,j) = \|y_i - y_j\|\}$ and $\|A\|_{\ell_2}$ represents the ℓ_2 matrix norm $\sqrt{\Sigma_{i,j} A_{i,j}^2}$. Also, $\tau (\cdot)$ is the second-order variation of the geodesic distances that converts the geodesic distance into an inner product.

$$\tau = -\frac{1}{2} H^T S H$$

where

$$S (i,j) = (d (i,j))^2$$

$$H (i,j) = \delta (i,j) - \frac{1}{n}$$

S is a matrix of squared distances, H is a centering matrix and δ is the identity matrix. The minimum of the cost function E can be found by setting the first d' eigenvectors of the matrix $\tau (D_G)$ as the coordinates for y_i. The residual variance of the data is decreased with each successive eigenvector. Once the number of eigenvectors reaches the intrinsic dimensionality of the manifold, the residual variance that is reduced from

successive addition of eigenvectors bottoms out. The result of performing nonlinear dimensionality reduction is a d'-dimensional representation of the sampled points where $d' < d$. In other words, a mapping is made such that $X_{\text{samp}} \rightarrow Y_{\text{samp}}$ and $y_{\text{samp}} = [y_1, y_2, \ldots, y_{d'}]$ where the distances between points in Y follow the manifold geometry.

Local linear embedding (LLE)

Another popular technique is local linear embedding (LLE) [16]. Similar to Isomap, it constructs a graphical representation of the data samples. But unlike Isomap, the method takes into account only the local neighborhood properties while Isomap preserves geodesic distance. In LLE, local properties of the data are represented as a linear combination of their nearest neighbors. The reconstruction weights, w_i to minimize the following cost function:

$$E\left(W\right) = \sum_i \left\| x_i - \sum_j^k w_{ij} x_j \right\|^2$$

The second summation represents the reconstruction of x_i from the k nearest neighbors of x_i. Thus, the norm can be seen as the reconstruction error of all points in X. In order to exclude the trivial solution, a constraint is put on the weights so that they are non-zero and also standardized to one.

$$\sum W_{ij} = 1$$

It has been shown by [16] that a low dimensional representation could be found by computing the eigenvectors corresponding to the smallest d non-zero eigenvalues of the inproduct $(I - W)^T (I - W)$. Here, W is the $n \times n$ reconstruction weight matrix where values are 0 if i and j are not neighbors and I is an $n \times n$ identity matrix. Compared to Isomap, the neighborhood features in LLE are less susceptible to short circuiting. A disadvantage of LLE is that it tends to collapse large distances to small distances in the low dimensional space. This occurs frequently when a manifold contains holes. LLE has been successfully applied to super-resolution [17] and sound source localization [18].

Local Tangent Space Analysis (LTSA)

Local tangent space analysis is a technique that maps data points to a low dimensional space by aligning neighboring tangent space information. The local tangent space Θ_i of data points x_i are found by applying PCA on the k nearest neighbors of data points x_i. A mapping M_i can thus be be made of each neighborhood to the local tangent space space Θ_i. A linear mapping L_i exists between the tangent space Θ_i to the low-dimensional representation y_i from the properties of local tangent space. Thus LTSA aims to minimize the following cost function:

$$\min_{Y_i, L_i} \sum_i \|Y_i J_k - L_i \Theta_i\|^2$$

where $J_k = I_k - \frac{1}{k} 1 \cdot 1^T$ is the centering matrix of size k. The minimization of this cost function was shown by Zhang and Zha [19] to be the eigenvectors corresponding to the smallest non-zero eigenvalues of an alignment matrix B. The elements of the alignment matrix B are obtained by iterative summations (for all matrices V_i and starting from $b_{ij} = 0$) for $\forall ij$

$$B_{N_i N_i} = B_{N_i N_i} + J_k \left(I - V_i V_i^T \right) J_k$$

where N_i is a selection matrix that contains the indices of the nearest neighbors of data point x_i. The low-dimensional representation Y is obtained by computation of the eigenvectors corresponding to the d smallest non-zero eigenvectors of the symmetric matrix $\frac{1}{2} \left(B + B^T \right)$.

LTSA has been successfully applied to microarray data [20] and anomaly detection [21].

Diffusion Maps

Diffusion maps takes a probabilistic approach to nonlinear dimensionality reduction [22]. First, a graph $G = (\Omega, W)$ is defined that contains n nodes. The weight matrix for this graph is defined as $W = \{w(x, y)\}_{x, y \in \Omega}$ and must satisfy two conditions. The weight matrix must be symmetric: $W = W^T$. This would mean that the graph is bidirectional where the edge from nodes x to y would equal the edge from nodes y to x. The second condition is that the graph must have point-wise positivity: $w(x, y) \geq 0$ for all $x, y \in \Omega$. This would just ensure that the weight matrix is

positive semidefinite. The actual weight matrix should be chosen dependent upon the application. A weight matrix could be something like a Gaussian kernel but the only requirement for an adequate weight matrix is that it must represent a similarity of x and y. The Gaussian kernel is defined as $w_\epsilon = \exp\left(-\|x-y\|^2/\epsilon\right)$ which can be applied when the data is defined by a set of discrete data points.

The graph G with weights W represent the known local geometry of the set. Now, a Markov random walk is defined on the graph. A degree $d(x)$ is defined for each node x as

$$d(x) = \sum_{z \in \Omega} w(x, z)$$

The degree can be seen as the sum of all the weights from x to all other nodes in the graph. Next, an $n \times n$ matrix P is defined whose elements are given by:

$$p_1(x, y) = \frac{w(x, y)}{d(x)}$$

Here, $p_1(x, y)$ can be seen as the probability of transitioning from x to y in 1 time step. Thus the matrix P can be interpreted as a transition matrix containing probabilities for transitioning from any one node to all other nodes. Since this matrix represents one transition, the values reflect the first-order neighborhood of the graph. This local information is extended to other points in the graph by taking powers of the matrix P. This is equivalent to running the random walk for additional iterations. In other words, if P^t is the tth power of P, then the value $p_t(x, y)$ represents the probability of moving from node x to y in t time steps. In diffusion maps, t is a free parameter that can be adjusted for each manifold. Increasing the value for t can be seen as extending the local information of the point to neighbors that are further away. Therefore, P^t reflects the connectivity of the graph while preserving the intrinsic geometry of the data set that is present in the local interconnections.

If the graph is connected, this is, if all the nodes in the graph can be visited by traveling along edges in the graph, we can assert that:

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

where ϕ_0 is the unique stationary distribution given by:

$$\phi_0\left(x\right) = \frac{d\left(x\right)}{\sum_{z\in\Omega} d\left(z\right)}$$

This value is proportional to the relative degree of x in the graph. The value is the degree of the node x relative to the sum of the degrees of all nodes in graph. This can be seen as a way to measure the density of points. A node with many neighbors will have a high degree and thus also a high unique stationary distribution signifying that the node is located in a dense area. Also, since the graph was defined to be bidirectional, the Markov chain is reversible. This would mean that the following balance condition holds true:

$$\phi_0\left(x\right)p_1\left(x, y\right) = \phi_0\left(y\right)p_1\left(y, x\right)$$

A distance metric between points in Ω can be extracted which will follow the manifold's intrinsic geometry. Two points x and z are considered close if their conditional probabilities $p_t\left(x, \cdot\right)$ and $p_t\left(z, \cdot\right)$ are close. This notion was used in a previous paper [23] to define distances on a Markov random walk. The advantage of using this scheme is that distances in a random walk can be evaluated with existing spectral techniques. This will later allow parametrization of the data using eigen-maps and eventually a method for reducing dimensionality. Thus, the "diffusion distance" D_t between x and y is defined as the weighted ℓ_2 distance:

$$D_t^2\left(x, z\right) = \|p_t\left(x, \cdot\right) - p_t\left(z, \cdot\right)\|_{1/\phi_0}^2 = \sum_{y\in\Omega} \frac{\left(p_t\left(x, y\right) - p_t\left(z, y\right)\right)^2}{\phi_0\left(y\right)}$$

where the "weights" $\frac{1}{\phi_0(x)}$ penalize points of low density more than those with high density.

Points that are in high density areas will have many short paths connecting each other in the graph resulting in a small diffusion distance. One way to picture this phenomenon is that one short path connecting two points is "evidence" that the points are close. If the points are indeed close, other short paths also exist, especially if the points are located in a high density area. Since all paths in the graph are considered, the diffusion distance is more robust to noise than if only the shortest path available is used, which can be prone to short circuiting.

Finally, the data set can be reduced to a lower dimensional manifold space. Since the definition of diffusion distances is connected to the spectral theory of random walk, eigen-map decomposition can be applied to the data. From spectral theory, the transition matrix P has a set of left and right eigenvectors and a corresponding set of eigenvalues $|\lambda_0| \geq |\lambda_1| \geq \cdots \geq |\lambda_{n-1}| \geq 0$:

$$\phi_j^T P = \lambda_j \phi_j^T$$

and

$$P \psi_j = \lambda_j \psi_j$$

It can be shown that $\phi_k^T \psi_l = \delta_{kl}$, or in other words, the product of a transposed left eigenvector and a right eigenvector corresponding to the same eigenvector will equal 1 and all other combinations will equal 0. Also, since the matrix is a transition matrix, it can be shown that $\lambda_0 = 1$ and $\psi_0 \equiv 1$. This occurs because each row of a transition matrix should sum to 1. Thus, the vector $\psi_0 \equiv 1$ right hand multiplied with P will always result in another vector of 1's. Therefore this eigenvector is trivial and is not considered in the final calculations. The two sets of eigenvectors can be related according to the following function.

$$\psi_l(x) = \frac{\phi_l(x)}{\phi_0(x)} \text{ for all } x \in \Omega$$

Thus, each set of eigenvectors can be normalized for ease of notation. The left eigenvectors of P are normalized with respect to $1/\phi_0$:

$$\|\phi_l\|_{1/\phi_0}^2 = \sum_x \frac{\phi_l^2(x)}{\phi_0(x)} = 1$$

The right eigenvectors are normalized with respect to ϕ_0:

$$\|\psi_l\|_{\phi_0}^2 = \sum_x \psi_l^2(x)\,\phi_0(x) = 1$$

Next, consider $p_t(x,y)$ which is the kernel of the tth iterate of the transition matrix P. The biorthogonal spectral decomposition will thus become:

$$p_t(x,y) = \sum_{j \geq 0} \lambda_j^t \psi_j(x)\,\phi_j(y)$$

This is analogous to performing a weighted principal component analysis on the matrix P^t since the process reduces the matrix into essentially a diagonalized covariance matrix. Following from properties of principal component analysis, the first k terms provide the best rank-k approximation of P^t that would minimize the variance of the data set. The metric for this minimization is shown below:

$$\|A\|^2 = \sum_x \sum_y \phi_0(x) \, a(x,y)^2 \frac{1}{\phi_o(y)}$$

Finally, by inserting the spectral decomposition from $p_t(x,y)$ into the diffusion distance $D_t^2(x,z)$, we can come to the following representation for distance in diffusion maps:

$$D_t^2(x,z) = \sum_{j=1}^{n-1} \lambda_j^{2t} (\psi_j(x) - \psi_j(z))^2$$

Once again, the trivial eigenvector $\psi_0 \equiv 1$ does not provide useful information and is excluded from this calculation. In summary, the diffusion distance $D_t^2(x,z)$ can now be represented by its eigenvectors and eigenvalues. Now, because of the decay of the eigenvalues, the diffusion distance can be approximated to a certain degree of accuracy using only the first few eigenvectors similar to how principal component analysis reduces linear matrices. To represent this as a formula, let $q(t)$ be the largest index j such that $|\lambda_j|^t > \delta |\lambda_1|^t$. The approximation for diffusion distance can thus be represented as the first $q(t)$ non-trivial eigenvectors and eigenvalues as follows:

$$D_t^2(x,z) \simeq \sum_{j=1}^{q(t)} \lambda_j^{2t} (\psi_j(x) - \psi_j(z))^2$$

The above formulation can be seen as representing the manifold as a Euclidean distance in $\mathbb{R}^{q(t)}$. The right eigenvectors ψ_j weighted with their corresponding eigenvalues λ_j are the coordinates of the data in the lower dimensional embedding. This can be represented as the following mapping:

$$\Psi : x \mapsto \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{q(t)}^t \psi_{q(t)}(x) \end{pmatrix}$$

We can then apply this to the diffusion distance calculation.

$$D_t^2\left(x, z\right) \simeq \sum_{j=1}^{q(t)} \lambda_j^{2t} \left(\psi_j\left(x\right) - \psi_j\left(z\right)\right)^2 = \left\| \Psi_t\left(x\right) - \Psi_t\left(z\right) \right\|^2$$

The mapping $\Psi : \Omega \mapsto \mathbb{R}^{q(t)}$ can be seen as parametrization of the original feature space into a lower-dimensional space $\mathbb{R}^{q(t)}$, where the weighted eigenvectors are the coordinates. In doing so, the data from G can be projected along a cloud of points in the lower dimensional manifold. The final equation $D_t^2\left(x, z\right)$ can be interpreted as a Euclidean distance approximation for the diffusion distance.

Diffusion maps has been used in applications such as image completion [24] and multiscale anomaly detection [25].

Discriminative orthogonal neighborhood-preserving projections

Discriminative orthogonal neighborhood-preserving projections (DONPP) is an extension to orthogonal neighborhood preserving projections that takes into account the labeled information of data [26, 27]. The general idea of this supervised manifold learning method is to use labeled information to cluster together same-class labels and separate different-class labels.

Consider a data set, X, DONPP assumes each data sample $\vec{x}_i$ can be reconstructed by the other same-class samples:

$$\vec{x}_i = \left(\vec{c}_i\right)_1 x_{i^1} + \left(\vec{c}_i\right)_2 x_{i^2} + \cdots + \left(\vec{c}_i\right)_{n_1-1} x_{i^{n_i-1}} + \vec{\epsilon}_i$$

Here, the subscript denotes the class label, $\vec{\epsilon}$ refers to the reconstruction error, and $\vec{c}$ are the reconstruction weights. To explain the subscript in x_{ij}, j refers to the class label while i refers to an arbitrary sample with class label j. As with other reconstruction methods, the goal is to minimize the error:

$$\arg\min_{\vec{c}_i} \left\| \vec{\epsilon}_i \right\|^2 = \arg\min_{\vec{c}_i} \left\| \vec{x}_i - \sum_{j=1}^{n_i-1} \left(\vec{c}_i\right)_j \vec{x}_{ij} \right\|$$

There are two parts to the solution. DONPP minimizes the reconstruction error of same-class samples:

$$\arg\min_{\vec{y_i}} \left\| \vec{y_i} - \sum_{j=1}^{n_i-1} (\vec{c_i})_j \, \vec{y_{ij}} \right\|$$

where $\vec{y_i}$ is the low dimensional representation of $\vec{x_i}$. Secondly, the algorithm maximizes the distance of different-class samples in the lower dimensional space.

$$\arg\min_{\vec{y_i}} \left\| \vec{y_i} - \sum_{j=1}^{n_i-1} (\vec{c_i})_j \, \vec{y_{ij}} \right\|$$

The two functions can be combined as follows:

$$\arg\min_{\vec{y_i}} \left(\left\| \vec{y_i} - \sum_{j=1}^{n_i-1} (\vec{c_i})_j \, \vec{y_{ij}} \right\|^2 - \beta \sum_{p=1}^{k} \|\vec{y_i} - \vec{y_{ip}}\|^2 \right) = \arg\min_{Y_i} \operatorname{tr} \left(Y_i L_i Y_i^T \right)$$

where

$$L_i = \begin{bmatrix} 1 - \beta k & -\vec{c_i}^T & -\beta \vec{e_k} \\ -\vec{c_i} & \vec{c_i}\vec{c_i}^T & \vec{z}_{n_i-1}\vec{z}_k^T \\ -\beta \vec{e_k} & \vec{z}_k \vec{z}_{n_i-1}^T & -\beta I_k \end{bmatrix}$$

Here, β is a trade-off parameter between the inter- and intra-class functions and $\vec{z}$ represents a vector the same length of x whose entries are all 0. The solution is an eigen-decomposition problem:

$$X L X^T \vec{u} = \lambda \vec{u}$$

[27] shows that DONPP is an improvement on orthogonal neighborhood-preserving projection which doesn't take into account the label information. The intra- and inter-class dynamics of this algorithm are taken into account when creating the supervised learning model in the proposed method.

2.2 LARGE-SCALE MANIFOLD LEARNING

A natural extension to linear and nonlinear algorithms is the application on very large databases. This can be attributed to the explosion of data in recent years [2]. Notably, this is because of the advances in the acquisition of data along with the increase in storage capacity. For example, conventional digital camera resolutions

have increased dramatically over the past couple decades. The cost per gigabyte of memory storage has also dropped dramatically thus allowing data sets to grow. Many conventional algorithms are inept at scaling to very large data sets due to constraints such as memory and computational strength. The problem that occurs in particular with nonlinear dimensionality reduction is the complexity of eigen-decomposition of a large matrix along with calculating the similarity matrix. In this section, some of the approaches to tackle the problem of large scale dimensionality reduction are discussed.

2.2.1 COLUMN SAMPLING APPROACH

A current method to handle a large data set is to approximate large matrices with column sampling [28, 29]. The column sampling approach considers a large matrix, G, where the number of rows, n, represent a sample and the number of columns n

This method approximates the singular value decomposition of a large matrix by calculating a simpler matrix comprised of columns sampled from the original. The original spectral decomposition is as $G = U\Sigma V^T$ where U and V are the left and right singular vectors respectively and Σ are the eigenvalues. Columns of G are sampled to produce a smaller matrix C using various techniques including random sampling or an adaptive approach. In [28], a probabilistic model is used to adaptively sample columns. The probability of a column to be sampled is:

$$P_i^j = \frac{\left\| E_j^{(i)} \right\|^2}{\|E_j\|^2}$$

where $E_1 = G$ and $E_j = G - \pi_{S_1 \cup \cdots \cup S_{j-1}}(G)$. Here, $pi_S(G)$ denote the matrix whose rows are projections of G to the span of S. In other words, $\pi_{S_1 \cup \cdots \cup S_{j-1}}$ are the projection of G onto the space of columns already sampled. Thus, the columns of G that are orthogonal to columns already selected have a probability of being selected in this approach. Let C be the matrix formed from k sampled rows, $C = \pi_{S_1 \cup \cdots \cup S_k}(G)$, the spectral decomposition of this matrix is computed directly.

$$C = U_C \Sigma_C V_C^T$$

where $C \in \mathbb{R}^{m \times n}$. U_C and V_C are the left and right singular vectors respectively and Σ_C are the eigenvalues. The equation can be rearranged so that the left hand

singular vectors can be given by:

$$\tilde{U} = U_C = CV_C\Sigma_C^+$$

where $\tilde{}$ represents an approximation. The approximated eigenvalues are proportional to the original eigenvalues according to:

$$\tilde{\Sigma} = \sqrt{\frac{n}{l}}\Sigma_C$$

Since $\tilde{G} = \tilde{U}\tilde{\Sigma}\tilde{U}^T$, the approximation of the original large matrix, G, can be shown:

$$G \approx \tilde{G} = C\left(\sqrt{\frac{l}{n}}\left(C^TC\right)^{\frac{1}{2}}\right)C^T$$

Thus, the original eigen-decomposition can be replaced with the decomposition of $\sqrt{\frac{1}{k}}\left(C^TC\right)^{\frac{1}{2}}$.

A benefit of this approach is that it has been shown by [30] that the approximation error is bounded and is inversely proportional to the number of sampled columns.

2.2.2 NYSTROM APPROXIMATION APPROACH

Another method that attempts to overcome the problems of a large data set by approximating the spectral decomposition is presented. This method uses a Nystrom approximation of spectral decomposition to approximate a manifold [31, 32, 33, 34].

Consider a $n \times n$ similarity matrix, G, which is symmetric positive and semi-definite. The spectral decomposition of this matrix is $G = U\Sigma U^T$ where Σ is the set of eigenvalues and U are the corresponding eigen-vectors. As is the case for dimensionality reduction, l columns are randomly sampled from G without replacement such that $l << n$. Because G is symmetric positive and semi-definite, it can be rearranged as follows:

$$G = \begin{bmatrix} W & G_{21}^T \\ G_{21} & G_{22} \end{bmatrix}$$

where W is the $l \times l$ matrix consisting of the intersection of the l sampled rows and columns, G_{21} corresponds to an intersection of sampled and unsampled rows, and G_{22} corresponds to the cross-intersection of the unsampled rows. The set of

sampled columns is identified as C:

$$C = \begin{bmatrix} W \\ G_{21} \end{bmatrix}$$

The authors of [31] used the Nystrom method to approximate the spectral decomposition. The Nystrom method is a common technique to speed up kernel machines. According to the Nystrom method, the similarity matrix can be approximated as:

$$G \approx \tilde{G} = CW^{+}C^{T}$$

where W^{+} is the pseudo-inverse of W. By substitution, the Nystrom method approximates G_{22} in $\tilde{G}$ using W and C such that:

$$\tilde{G} = \begin{bmatrix} W & G_{21}^{T} \\ G_{21} & G_{21}W^{+}G_{21}^{T} \end{bmatrix}$$

It was shown that the Nystrom method outperforms column-sampling approximations for large face databases. The approximated eigenvalues and eigenvectors are related to the originals as follows:

$$\tilde{\Sigma} = \left(\frac{n}{l}\right)$$

$$\tilde{U} = \sqrt{\frac{l}{n}}CU_{w}\Sigma_{W}^{+}$$

where $W = U_{W}\Sigma_{W}U_{W}^{T}$.

The column sampling method generally generates more accurate singular values and low-rank matrix projections while the Nystrom method constructs better low-rank approximations [35].

2.2.3 LANDMARK MULTIDIMENSIONAL SCALING

Another approach to implement a scalable dimensionality reduction method is Landmark Multidimensional Scaling (Landmark MDS or LMDS) [36, 37, 38]. Essentially, this algorithm extends the classical MDS by sampling the large data set into a smaller, more manageable one. Classical MDS is performed to extract low dimensional projections and the remaining data points are embedded into the low

dimensional space.

This algorithm is split into a few steps. First, a set of landmarks is sampled from the entire large data set. The particular method is used arbitrary. The authors suggest that either random sampling or min-max sampling may be used.

Next, classical MDS is performed on the landmarks to find a $k \times n$ matrix L.

$$B_n = -\frac{1}{2} H_n \Delta_n H_n$$

where H_n is the mean-centering matrix such that $[H_n]_{ij} = \sigma_{ij} - \frac{1}{n}$

A distance-based triangularization method is then used to embed the out-of-sample points to the lower dimensional representation. The new coordinates of a point a are obtained by an affine linear transformation of the vector $\vec{\sigma}_a$ of its squared distances to the landmarks.

$$\vec{x}_a = -\frac{1}{2} L_k \left(\vec{\sigma}_a - \vec{\sigma}_u \right)$$

where $L_k = \left[\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_k} \right]^T$ is the embedding vector. Here, λ and v are eigenvalues and eigenvectors found from the classical MDS step.

It was shown in [33] that this algorithm falls in the category of an application to the Nystrom approximation. This procedure was influential to the development of the algorithm presented in this dissertation. A difference is that this dissertation's algorithm is applied onto Isomap and incorporates a general sparse neighbor selection method along with an adaptive sampling approach.

2.3 SPARSE LEARNING

In the literature, it has been shown that sparse representation has applications in a wide range of computer vision and pattern recognition including face and object recognition [39], compressive sensing [40, 41], subspace learning [1], medical image analysis [42], and dictionary learning [43]. This dissertation's method to alleviating the problems associated with large data sets revolves around finding a sparse representation of the data set. By reducing the sample size of the data set, a full eigen-decomposition could be performed. In this section, methods that exploit sparsity for manifold learning are explored.

2.3.1 SPARSITY PRESERVING PROJECTIONS

Sparsity preserving projections (SPP) performs dimensionality reduction by focusing on preserving a sparse representation rather than preserving neighborhoods [44].

Consider, a data set X with n samples with m dimensions. The problem is formally expressed as finding a set of sparse weights s_i that can reconstruct a data point with the rest of the data set. This becomes a minimization problem:

$$\min_{s_i} \|s_i\|_1$$

where $\|\cdot\|_1$ represents an ℓ_1-norm. The minimization is constrained with the reconstruction error of each point.

$$\|x_i - Xs_i\| < \epsilon$$
$$1 = 1^T s_i$$

Here, ϵ represents an error tolerance. The second equation is a normalization constraint. In order to project these weights to a lower dimensionality, [44] proposes the following objective function:

$$\min_{w} \sum_{i=1}^{n} \left\| w^T x_i - w_T X \tilde{s}_i \right\|^2$$

where $w \in \mathbb{R}^m$ is a projecting direction vector. This minimization can be represented as a maximization problem:

$$\max_{w} \frac{w^T X S_\beta X^T w}{w^T X X^T w}$$

where $S_\beta = S + S^T - S^T S$. Here, $w^T X X^T w = 1$ to prevent degenerate solutions. As the case with many previous manifold learning approaches discussed, spectral decomposition is used to find the lower dimensional projections. The optimal solution to the objective function correspond to the largest eigenvalues of the following eigen-decomposition problem:

$$X S_\beta X^T w = \lambda X X^T w$$

This method was shown to be an improvement to linear approaches such as local preserving projection and neighborhood preserving embedding for face data sets. A problem that occurs with this method is the calculation of the sparse weights. This is a computationally expensive step since it is performed over all data points. This problem is addressed in the proposed method of this dissertation by limiting sparse weight calculations to a neighborhood.

SPP has been applied to face recognition [44] and eyebrow recognition [45].

2.3.2 SAMPLE-DEPENDENT LOCALITY PRESERVING PROJECTION

Sample-dependent locality preserving projection (SLPP) is a dimensionality reduction approach that employs sparse graph construction [1].

For a data set, X, the adjacency matrix, W_{ij}^S, is a matrix that gives a similarity metric between all points in the data set.

$$W_{ij}^S = \begin{cases} \exp\left\{-\frac{d(x_i, x_j)}{2t^2}\right\}, & \exp\left\{-\frac{d(x_i, x_j)}{2t^2}\right\} > \frac{1}{n}\sum_{k=1}^{n} \exp\left\{-\frac{d(x_i, x_j)}{2t^2}\right\} \\ 0, & \texttt{otherwise} \end{cases}$$

where $d(x_i, x_j) = \|x_i - x_j\|^2 / \sum_{k=1}^{n} \|x_i - x_k\|^2$ can be seen as a distance metric between two points x_i and x_j. A similarity parameter, t, is used to control the selection of neighbors. This particular adjacency matrix is an adaptation of the adjacency matrix of classical locality preserving projection (LPP) [46]. The difference is the inclusion of sparse neighbor selection. Conventional LPP generally connect neighbors using a k-nearest neighbor or ϵ-neighborhood approach. SLPP on the other hand considers x_i and x_j neighbors only if their similarity is greater than the mean of similarities between x_i and all other points.

With the adjacency matrix, W^S, a cost function could be created as below:

$$\sum_{i,j} \|y_i - y_j\|^2 W_{ij}^S = w^T X L^S X^T w$$

where y_i and y_j are the lower dimensional representations of x_i and x_j. Also, $L^S = \sum_i Wij^S + \sum_i Wij^S - W^S$ is the Laplacian matrix. The optimization then becomes a minimization problem:

$$\min_{w} w^T X L^S X^T w$$

$$\text{s.t. } w^T X L^S X^T w = 1$$

Once again, the minimization is an eigen-decomposition of the similarity matrix.

SLPP uses an adaptation of the adjacency matrix from LPP to achieve a adaptive and sparse graph. In this dissertation, this sparse graph idea is extended to be used with existing manifold learning approaches using an ℓ_1-based optimization.

2.3.3 SELECTING LANDMARK POINTS FOR SPARSE MANIFOLD LEARNING

A method that applies sparse learning to landmark selection for dimensionality reduction is described in [47]. The algorithm uses the sparseness property of the least absolute value subset selection operator (LASSO) [48, 49] and least angle regression stagewise (LARS) methods [50].

The sparsity is based on an estimate W that minimizes the cost function, E, below:

$$E = \|Y - KW\|^2 + \gamma \|W\|_1$$

where γ is a tuning parameter that controls the amount of regularization. K is a square symmetric semidefinite positive matrix which is generally a similarity metric and Y is the response. The first term is the reconstruction error seen with many optimization functions. The second term is the regularization that promotes sparsity. The inclusion of this term makes the solution more difficult to find. LASSO and LARS are employed to solve functions of this form. Essentially, landmarks with non-zero values in W are selected as landmarks. A sparse sample set of a large data set could created.

While this algorithm is useful in reducing the size of a data set, it may not be scalable to very large data sets since it requires the LASSO computation to be conducted on the entire data set. In this dissertation's proposed method, large scale LASSO operations are avoided using a bounded neighborhood approach. Another difference is that this algorithm focuses on landmark selection while our approach focuses on neighborhood selection as discussed in Chapter 3.

Chapter 3

PROPOSED METHOD

In this chapter, a method for large scale manifold learning is presented. First, an incremental method is described that employs sampling to reduce the size of a large data set to something that is manageable for traditional manifold learning algorithms calculated on conventional computers. A linear embedding approach is used to re-combine the unsampled points into the manifold. Next, the algorithm is enhanced using the ℓ_1-norm to automatically select sparse neighbors. Since neighborhood selection is often used in manifold learning, especially the graph based methods, this novel neighborhood selection can be applied to many manifold learning methods. One such application is on the Isomap algorithm which is normally very sensitive to neighborhood selection due to possible short circuiting and leaking. The overall goal of this manifold learning algorithm is not only to be scalable to large data sets, but also be adaptable to various neighborhoods.

3.1 INCREMENTAL LARGE-SCALE MANIFOLD LEARNING

3.1.1 SAMPLING

In this step, the number of data points used for processing will be reduced using sampling. Ideally, landmarks sampled should be the smallest subset that can pre-serve the geometry in the original data. The next step performed after sampling is dimensionality reduction. To keep a faithful representation of the original manifold, landmarks used for the manifold skeleton learning should be carefully selected from the original data. Thus, to preserve the overall geometric structure of an underly-ing manifold, efficient sampling is essential in the system. Here, a sampling method is proposed that samples data points based upon the local curvature variation of each data point. Min-max sampling and random sampling are also introduced as alternatives.

Local Curvature Variation Sampling

In local curvature variation (LCV) sampling, landmarks are chosen depending upon the level of curvature of a data point's local neighborhood. It is based on the assumption that areas of high curvature provide a greater insight on manifold geometry than relatively flat areas. Figure 3 shows a toy data set to illustrate the basic idea of LCV. Heuristically, to preserve the data structure after sampling, more data points should be kept near area 'A' rather than area 'B' in Figure 3 because data structures near 'A' change more abruptly. Based on this observation, an importance value was assigned for each of the points by computing the local curvature for it. For each data point in the data set, its k_{curv}-nearest neighbors were found and the local curvature c_i of each point x_i was found [51]:

$$c_i \approx \frac{1}{k_{curv} - 1} \sum_{l=2}^{k_{curv}} \frac{\arccos\left(\sigma_{min}\left(Q_i^T Q_{i_l}\right)\right)}{\left\|\theta_l^{(i)}\right\|_2}$$

where $\sigma_{min}\left(\cdot\right)$ represents the smallest singular value. Q_i is an orthonormal basis of the tangent space of k_{curv} neighbors. Normalization was performed by scaling of the magnitude of the tangent space projection $\theta_l = Q^T \left(x_{i_l} - \bar{x}_i\right)$, where $\bar{x}$ is the mean of the points in the neighborhood. A probability density function, $p\left(x\right)$, was obtained by normalizing the curvature values such that they sum to one.

$$p\left(x_i\right) = \frac{c_i}{\sum_{j=1}^{n} c_j}$$

Samples were then selected from the probability density function using importance sampling. It can be seen from the probability function $p\left(x_i\right)$ that the probability value for each point x_i is directly correlated to its curvature value c_i. Thus, points with higher curvature have higher possibilities of being selected. A small set of n representative points were selected yielding $X_{samp} = \left[x^{(1)}, x^{(2)}, \ldots, x^{(n)}\right]^T$.

Min-max Sampling

Another sampling method is min-max sampling. The general goal of this method is to maximize the distance between sampled points. The result is a set of samples that are evenly distributed in the spatial domain. In this algorithm, a seed is selected to become the first sample or landmark. Next, the point with the largest distance
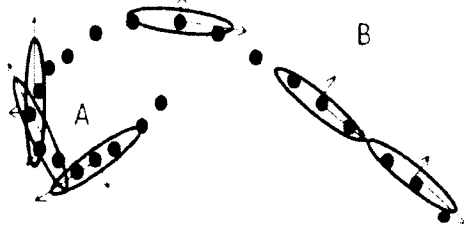
Figure 3: Local curvature variations

away from the sampled point is chosen as the second landmark.

The subsequent points are calculated based upon distances from previously selected landmarks using an iterative two step process. First, the distance, D, between all non-landmarks to every selected landmark is calculated. Thus, D is an $i \times n - i$ matrix where i is the number of currently selected landmarks and n is the total number of sample points. The minimum distance $D^{\min}$ of each row is then selected.

$$D_i^{\min} = \min D_i$$

Essentially, the distance to the closest landmark of each remaining point is found. This is the "min" portion of this algorithm. Finally, the point with the largest distance in $D^{\min}$ is chosen as the next landmark in the "max" phase of the algorithm.

$$\arg \max D^{\min}$$

The process is then iterated until the desired number of landmarks k is reached. This particular algorithm generally results in a relatively uniform distribution of landmarks. It is important to note that after the initial point is selected, the rest of the algorithm is deterministic.

While this method will guarantee that the samples are distributed, it is not without its drawbacks. For example, outliers in a data set will have a large impact since they have a relatively larger distance than data samples that are grouped in clusters. Additionally, densely grouped clusters will be sampled sparsely which in result may not represent the data accurately.

Random Sampling

A very simple sampling method and the de facto standard sampling method is random sampling. Here, samples are chosen using a simple random draw without replacement from the set of all data points. The results from this will provide a baseline for other sampling approaches. In some applications, complex sampling algorithms may not be necessary for successful manifold learning, but rather costs additional computations. One such situation is when a large portion of the data set is to be sampled thus not requiring sampling to be efficient.

3.1.2 MANIFOLD SKELETON LEARNING

The main constraints of manifold learning algorithms on large data sets are the memory and computational costs of having a very large number of samples. With almost all nonlinear manifold learning techniques, there is an eigen-decomposition of an $n \times n$ similarity matrix which becomes a problem for large n. Additionally, the computation of this similarity matrix can also become a bottleneck. For example, Isomap requires calculating geodesic distance between every single point and diffusion maps requires a diffusion weight between every single point. Sampling the large data set alleviates these problems.

A manifold learned from a sampled data set could be seen as an approximation of the manifold learned from the entire data set. In particular, the manifold learned from this approach can be viewed as a Nystrom approximation of the manifold if it were trained upon the original large data set. The small data set size after sampling allow the direct application of a manifold learning algorithm. We refer to the manifold found using the sampled data points as the manifold skeleton.

The sampled data set may be used with a variety of conventional nonlinear dimensionality reduction techniques. In this dissertation, we mainly focus on using the Isomap approach for manifold learning because of its intuitiveness and straight-forward step by step procedure. The Isomap algorithm is explained in detail in Section 2.1.2, but as a reminder, the algorithm is broken down into three steps. First, a neighborhood graph is computed using a neighborhood selection approach. Then, pair-wise geodesic distances are calculated. And then, MDS is performed for dimensionality reduction. A novel neighborhood selection method will be discussed in Section 3.2 that will address the leaking issues with the algorithm.

The result after this step will be a low dimensional representation of the data using the sampled points. The notion behind using the sampled points instead of using the entire data set is that nonlinear dimensionality reduction techniques typically require an eigen-decomposition step of an $n \times n$ similarity matrix. When dealing with large scale data, this step will be prohibitive in terms of both computation time and memory. In the next step, the remaining data will be inserted into the low dimensional space using an embedding algorithm. Thus, the low dimensional representation found in this step will be referred to as the manifold skeleton.

3.1.3 EMBEDDING BY LLE:

An important step in the large scale application of manifold learning is incorporating the non-sampled points with the manifold skeleton. In a very large data set where only a small sample is processed, the majority of data points will fall into the category of an out-of-sample point. Thus, an effective embedding algorithm is essential to successful manifold learning. In this section, local linear embedding (LLE) is introduced as an embedding technique that is often used in the literature [36]. Later in Section 3.2.4, a novel embedding technique using ℓ_1-norm optimization is discussed.

Once the manifold skeleton is learned, the remaining data points are inserted into it using the LLE algorithm [16, 52].

First, each out-of-sample point X_r is connected to its k-nearest landmarks X_k in the original high-dimensional space. A linear model was then computed to reconstruct X_r by minimizing the following function:

$$E(W) = \sum_r \left| X_r - \sum_k W_{rk} X_k \right|^2$$

To embed the data point X_r into the lower-dimensional manifold, we reconstructed it in the low-dimensional space as Z_r using the weights W_{rk} derived above. The manifold space reconstruction can be summarized as follows:

$$Z_r = \sum_k W_{rk} Z_k$$

where Z_k are the low-dimensional representations of the landmarks X_k. An example of the LLE application is shown in Figure 4, where the red dots are landmarks
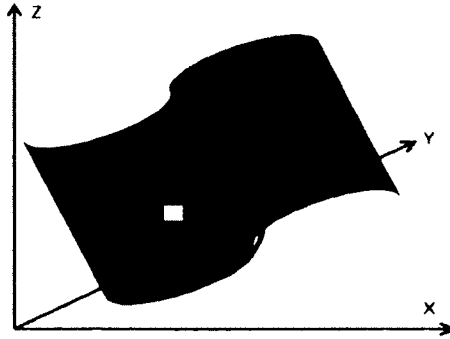
Figure 4: Illustration of local linear embedding

comprising the manifold skeleton, and the yellow square is a data point to be embedded into the skeleton. After this step, all the original data points are converted to the low dimensional nonlinearly reduced manifold space.

Essentially, this algorithm is calculating the reconstruction weights of a data point using the sampled landmarks and then using those landmarks and weights to reconstruct the data point in manifold space.

3.2 SPARSITY ENHANCEMENTS TO LARGE SCALE MANIFOLD LEARNING

In this section, the general method discussed in Section 3.1 is modified to incorporate the ℓ_1-norm. A novel neighborhood selection algorithm is introduced that will adaptively select sparse neighbors in the graph construction and embedding steps. The algorithm takes advantage of the sparsity property of the ℓ_1-norm optimization. In this section, the differences between the ℓ_1- and ℓ_2-norms are discussed. In particular, we focus on the reason that ℓ_1 optimization has a tendency to produce sparse weights. Next, the changes to the graph construction and embedding module are presented. Then, each module is individually tested to see if the ℓ_1-norm makes an improvement over the baseline method. Finally, the new proposed method is tested against benchmark data sets.

3.2.1 DIFFERENCE BETWEEN ℓ_1- AND ℓ_2-NORM

The main purpose of using an ℓ_1 based optimization rather than the more ubiquitously used ℓ_2-norm is that a sparse optimization can be achieved under the ℓ_1-norm.

This is useful in neighborhood selection since it is usually more preferable to connect a point with only a few neighbors. In order to explain why the ℓ_1-norm has the sparsity property, it is useful to discuss the background of these norms.

Both of these methods produce a metric that is used to represent the relative distance or size of a vector. They also have the same properties that all norms possess. From conventional mathematics, these properties include:

$$\|x\|_p > 0 \text{ when } x \neq 0$$

$$\|x\|_p = 0 \text{ iff } x = 0$$

$$\|kx\|_p = |k| \, \|x\| \text{ for any scalar } k$$

$$\|x + y\|_p <= \|x\| + \|y\|$$

Because of these encompassing properties, the various kinds of norms may theoretically be interchanged with one another.

The most obvious difference between the two norms is in the way each is calculated. Consider a simple vector $x \in \mathbb{R}^n$. The ubiquitous ℓ_2-norm is calculated as follows:

$$\|x\|_2 := \sqrt{\sum_{i=0}^{n} x_k^2}$$

The ℓ_1-norm on the other hand is defined as:

$$\|x\|_1 := \sum_{i=0}^{n} |x_k|$$

In other words, the ℓ_2-norm is the square root of the sum of each element squared while the ℓ_1-norm is the sum of the absolute value of each element.

The difference between the calculations are easy to see but the reason that an ℓ_1-norm produces a sparser representation is more difficult to realize. The explanation may best be clarified graphically. In Figure 5, a 2-dimensional example of optimization using the two methods is compared. In both plots, a minimization is desired. The plot on the left demonstrates ℓ_2 minimization while the right plot minimizes the same problem using ℓ_1. The concentric shapes give a sense of distance for both norms. For the ℓ_2 case, distances are calculated using the square root of a sum of squares which geometrically represents a circle centered around the origin. Thus,
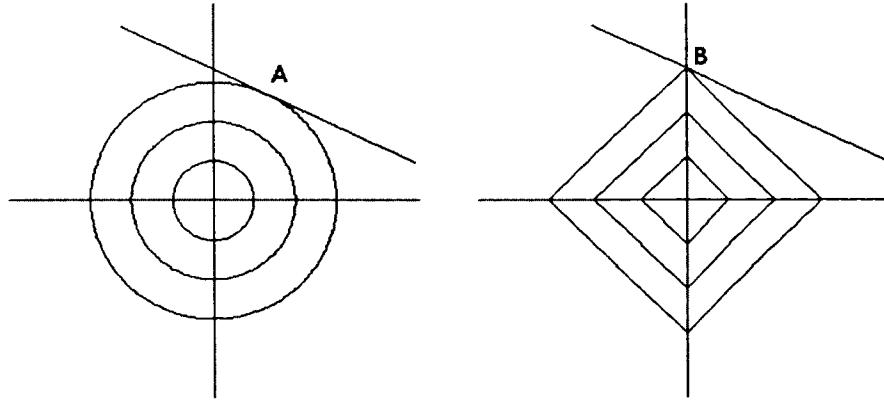
Figure 5: A 2 dimensional example of optimization using ℓ_2 minimization (left) and ℓ_1 minimization (right)

each point on the circles represent the same distance away from the origin. Similarly, the diamond represents the same distance away from the origin using the ℓ_1-norm.

The optimal solution for each case in Figure 5 occurs when the line intersects one of the concentric shapes. The minimum for the ℓ_2 case occurs at A while the minimum for the ℓ_1 case occurs at B. Here, point A will have a horizontal and vertical component resulting in two non-zero values to represent it. On the other hand, the pointed tip of the diamond at point B is the minimum distance for the ℓ_1 case. This point will only have one non-zero value since the horizontal component is zero. Thus, ℓ_1 optimization generally produces a sparser representation compared to ℓ_2 optimization because of the shape of the distance calculations.

3.2.2 SOLVING ℓ_2- AND ℓ_1-OPTIMIZATION

As discussed in the previous section, calculating an ℓ_1-norm and ℓ_2-norm from a vector is relatively simple. Optimization of the ℓ_1-norm, on the other hand, is not trivial. This is because a typical ℓ_2-norm optimization problem is differentiable while the ℓ_1-norm is not. For the ℓ_2 case, consider a common least squares problem that reconstructs a target t with N samples:

$$E(\omega) = \sum_{n=1}^{N} \left\{ t_n - w^T \phi(x_n) \right\}^2$$

where $\phi(x)$ is a basis function and w are weights. Since the ℓ_2-norm is differentiable, the minimum can be found by solving for the roots of the first derivative.

$$0 = \sum_{n=1}^{N} \left\{ t_n - w^T \phi(x_n) \right\} \phi(x_n)^T$$

$$0 = \sum_{n=1}^{N} t_n \phi(x_n)^T - w^T \left(\sum_{n=1}^{N} \phi(x_n)\phi(x_n)^T \right)$$

The optimal solution for the weights W_s can be solved using the Moore-Penrose pseudo-inverse.

$$W_s = \left(\Phi^T \Phi \right)^{-1} \Phi T$$

Note that the minimization can thus be found using simple matrix operations. Since the least squares method using the ℓ_2-norm is so straightforward, it has become the ubiquitous standard for these types of problems.

While ℓ_2 optimization is straightforward, the absolute value operation of ℓ_1-norm optimization is non-differentiable thus making the calculation more difficult. In this implementation, we used a toolbox known as Sparse Learning with Efficient Projections (SLEP) to solve the optimization [53]. SLEP is a gradient based approach that is efficient and thus suitable for large data sets.

In this section, we discuss a brief outline of the SLEP method. A more detailed description can be found in [54]. SLEP is based on an adaptive line search method known as Nesterov's method. A line search method is an iterative approach to finding a local minimum. Essentially, these minimization techniques find a descending direction and compute a step size along the direction of the descent. The process is iterated until a local minimum is found. Common line search techniques include gradient descent and Newton's method.

The first obstacle in line search methods is to calculate a gradient. Let $\mathbf{b} = [b_1, b_2, \ldots, b_m]^T$, $\mathbf{1}$ be a vector of all ones, $\mathbf{c}$ is a vector with all entries being c and $A = [b_1 a_1, b_2 a_2, \ldots, b_m a_m]$. The gradient $f'(w, c)$ can be calculated as below:

$$f'(w, c) = \left[\nabla_w f(w, c)^T, \nabla c f(w, c) \right]^T$$

$$\nabla_c f(w, c) = -\frac{1}{m} \mathbf{b}^T (\mathbf{1} - \mathbf{p})$$

$$\nabla_w f(w, c) = -\frac{1}{m} A^T (1 - \mathbf{p})$$

$$p = 1./(1 + \exp(-Aw - \mathbf{b} \odot \mathbf{c}))$$

where $\odot$ denotes component-wise multiplication.

Now, consider the problem to be a smooth convex minimization problem where the weights are constrained to an ℓ_1-ball:

$$\min_{w,c} g(w, c)$$

$$\text{subject to } \|w\|_1 \leq z$$

Nesterov's method is an optimal first-order line search method for smooth convex optimization. As with other line search methods, there are two sequences: $\{x_k\}$ and $\{s_k\}$. Here, $\{x_k\}$ is the sequence of approximate solutions and $\{s_k\}$ is the searching sequence.

$$s_k = x_k + \beta_k (x_k - x_{k-1})$$

$$x_{k+1} = s_k - \frac{1}{L_k} g'(s_k)$$

where β_k is a tuning parameter. L is called the Lipschitz gradient. Thus $1/L_k$ can be seen as a step size where:

$$L = \max_{x \neq y} \frac{\|g'(x) - g'(y)\|}{\|x - y\|} < +\infty$$

An illustration of Nesterov's method is shown in Figure 6. Here, the search point s_k is the affine combination of x_{k-1} and x_k. Each successive x_k is dependent on the gradient at s_k and the step size. The values are computed recursively until they arrive at the optimal solution x^*.

The global convergence rate of this method is $\mathcal{O}(1/k^2)$ while gradient descent, a popular line search method, is $\mathcal{O}(1/k)$.

3.2.3 ℓ_1 GRAPH CONSTRUCTION

In the previous sections, the background of the ℓ_1-norm is discussed. In particular, the differences between the ℓ_1-norm and the ℓ_2-norm are identified along with the sparsity property of the ℓ_1-norm. Also, a brief overview in solving an ℓ_1 optimization problem was discussed. The next few sections focus on the application of
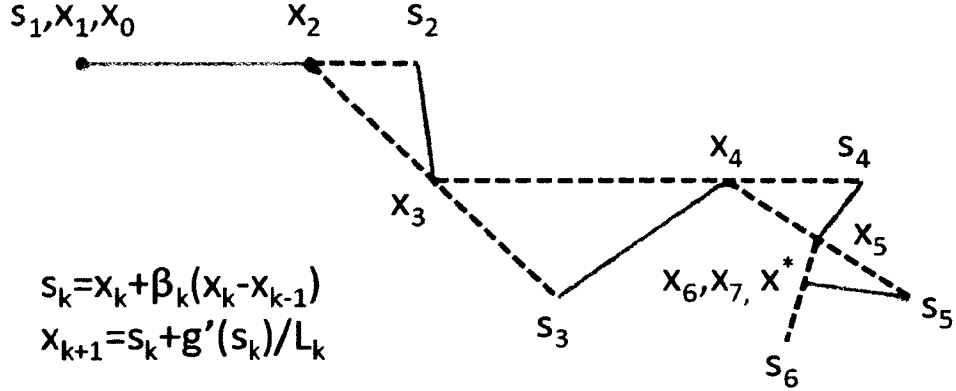
Figure 6: Example of Nesterov's method

ℓ_1 minimization to manifold learning.

In this section, the graph construction of Isomap is modified using a sparse ℓ_1 optimization. Recall from Section 2.1.2, the Isomap algorithm consists of three steps. First, a neighborhood is created to form a neighborhood graph. Next, a geodesic distance matrix is calculated from the neighborhood graph. This matrix will consist of pairwise distances between every sample of the data set where the distances are calculated by paths from the neighborhood graph. Finally, dimensionality reduction is performed on the distance matrix to form the lower dimensionality space while preserving the geodesic distances found in the second step. A successful manifold using Isomap depends largely on the formation of the neighborhood graph. In some cases, it has been found that a good graph construction is more important than selecting a good manifold learning algorithm [55].

One of the major drawbacks of the Isomap algorithm is that it is susceptible to leaking or short circuiting. This can happen if neighbors are selected such that a short cut is created between two areas that do not follow the intrinsic structure of the manifold. As a result, large geodesic distances are misrepresented as short distances because of a poorly formed graph. As an example, refer to Figure 7. The intrinsic distance between the two points along the intrinsic structure is large. However if there were neighbors directly between the two points, the distance calculated will adhere more with the dotted line producing an erroneous geodesic distance calculation. The leaking problem is especially prevalent in noisy data sets. The graph construction is thus a very critical step for the Isomap method to create a successful manifold.
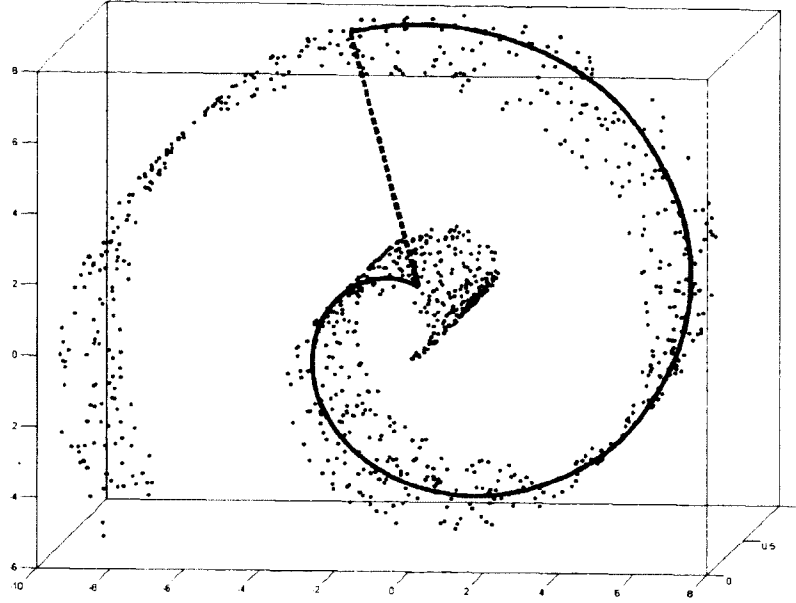
Figure 7: Swiss roll leak example. Blue solid line shows geodesic distance while dotted blue line shows Euclidean distance between the same points.

Because of the short circuiting problem, it is undesirable to have a large neighborhood. In the original graph construction method, the nearest neighbors are chosen from k-nearest neighbors using Euclidean distance. The selection of k is critical in creating a good graph. If k is too large the risk of short circuiting is higher. On the other hand, if k is too small, the graph may not be fully connected. Another problem that occurs is that a good k value for one location may be inappropriate for another location. In this section, these problems are addressed with an adaptive neighbor selection method implemented with the ℓ_1-norm.

Consider an $n \times m$ high dimensional data set X that we want to connect via a graph. Now consider a target sample x_i which will need to be connected to a few neighbors. A reconstruction cost function is used such that:

$$\min_{\omega_i \geq 0} \frac{1}{2} \| X\omega_i - x_i \|_2^2 + \lambda \| \omega_i \|_1$$

with the constraint,

$$\| \omega_i \| > 0$$

The first term is the reconstruction error while the second term enforces the ℓ_1 sparsity. Here, the tuning parameter λ controls the contribution of the sparsity term. The additional constraint is added so that there is at least one non-zero weight. Any sample, x_j, from X that has a non-zero weight, ω_{ij}, will be considered a neighbor.

In this optimization, every point in X is a potential neighbor for x_i. Since the number of samples will generally be high, the computational expense could also be a problem. It is safe to assert that the neighborhood should also be close with respect to the ℓ_2-norm. Therefore, the computational problem can be avoided by limiting the number of samples that could be potential neighbors. The neighborhood is first confined to the k nearest neighbors. This will guarantee that neighbors will not be selected far away from the target point and also speeds up the ℓ_1 optimization. Thus, the final optimization becomes:

$$\min_{\omega_i \geq 0} \frac{1}{2} \left\| X^{\{i\}}\omega_i - x_i \right\|_2^2 + \lambda \left\| \omega_i \right\|_1 \tag{1}$$

where $X^{\{i\}}$ are the k nearest neighbors of x_i. The entire graph is found by finding neighbors for all samples in the data set X. The optimal value for k will vary depending upon the application but a good starting value may be the dimensionality of the original dataset m. This is because the reconstruction term is a linear combination of neighbors. If the number of neighbors selected is above the original dimensionality $k > m$, the solution may be linearly dependent and result in an infinite number of solutions for the reconstruction term. After the graph is constructed using the proposed ℓ_1-norm approach, the remaining steps of the Isomap algorithm follow normally.

3.2.4 ℓ_1 EMBEDDING

The method for embedding as described in Section 3.1.3 was modified. The goal of this effort is to more effectively recombine points that are not used in the dimensionality reduction step into the manifold skeleton. The basic LLE algorithm requires one parameter k for the number of neighbors to use as weights. By using the ℓ_1-norm, the neighbors can be chosen automatically among all n data points. The non-zero weights of W_{rk} correspond to the neighbors selected to make a reconstruction of the point in manifold space. This adaptive approach is expected to be more robust since the value for k is not fixed for each data point. The LLE equation from Section

3.1.3 is thus modified as follows:

$$E\left(W\right) = \sum_r \left(\left| X_r - \sum_{k=1}^{n} W_{rk} X_k \right|^2 + \lambda \sum_{k=1}^{n} |W_{rk}|_1 \right)$$

The ℓ_1-norm in the second term promotes sparsity of the solution. This is because most weights calculated from the ℓ_1-norm will be zero. The few neighbors with a non-zero weight, W_{rk}, will be chosen as the landmarks to reconstruct the data in the low dimensional manifold. The tuning parameter λ controls the trade-off between reconstruction error and sparsity of the solution.

One drawback to this is that the localization of neighbors in Euclidean space is not guaranteed. This is because each point in the data set is considered to be a potential neighbor. Another consequence of the large number of potential neighbors is that the number of non-zero weights could potentially be very high.

In a sense, the ℓ_1-norm has the benefit of having an adaptive number of neighbors while the original method guarantees localization and a maximum number of neighbors. In order to incorporate all of these properties, the ℓ_1 optimization function was then modified to only select neighbors from within the k-nearest neighbors:

$$E\left(W\right) = \sum_r \left(\left| X_r - \sum_{k=1}^{k'} W_{rk} X_k \right|^2 + \lambda \sum_{k=1}^{k'} |W_{rk}|_1 \right) \tag{2}$$

The neighbors in this formula are restricted to the k' nearest neighbors of each point. In the implementation, k' can be a larger number than typically used in the original method since the sparsity term will reduce the number of neighbors. As with the ℓ_1-norm based graph construction approach, a starting value for the number of nearest neighbors k' may be the number of dimensions in the original data set. Using the weights found from this new cost function, the embedding can still be performed following the same procedure as LLE.

3.2.5 SUPERVISED GRAPH METHOD

In this section, a change to the proposed method is introduced to alter it for supervised learning. The graphing technique presented in Section 3.2.3 is an unsupervised approach. In other words, the method will try to find the manifold only using unlabeled data. In many data sets, the class label of the training information

is prior information. To fully utilize the information available, an adaptation is incorporated into the proposed general procedure to account for labeled information. Intuitively, samples should be connected to other samples that are the similar to itself. The ℓ_1-norm in Section 3.2.3 selects only a few neighbors that can reconstruct the target point. The goal is to have neighbors that are similar to the target point or are of the same class as the target point. In the supervised method where training labels are known, the set of potential neighbors can be selected such that only same class labels are selected as neighbors.

In this case, the set of potential neighbors only contains the k nearest neighbors belonging to the same class as x_i.

A problem arises when this approach is used. If samples are only connected to its same class neighbors, then the graph will not be fully connected. There will be no path through the graph to connect any inter-class points. While this separation may seem beneficial, it poses a problem when making comparisons between multiple classes. For example, it is not possible to conclude whether an unlabeled data point is closer to one class or another.

Thus, a modification to the supervised approach is necessary. In particular, it is necessary to create an additional connection, edge weight, or branch in the graph to join disjoint clusters. But the core of the problem is is how to do so. Here, we propose two different methods to connect interclass clusters.

The first method is to create a branch in the graph connecting the closest two neighbors of two class clusters. Consider two clusters from classes A and B whose samples are $x^{(A)}$ and $x^{(B)}$ respectively. The selection of the two closest points can be written as below:

$$\arg\min_{i,j} \left\| x_i^{(A)} - x_j^{(B)} \right\|$$

Thus, a connection is added in the graph between points $x_i^{(A)}$ and $x_j^{(B)}$. This example showed a two class data set but the process is easily repeated for multiple classes.

The second method we propose is to connect the centroids, or geometric center, of class clusters. This is inspired by the k-means clustering method. Consider a data set X with m classes such that the clusters for each class is represented as $X^{(1)}, X^{(1)}, \ldots, X^{(m)}$. The centroid, C, is calculated as follows:

$$C^{(i)} = \frac{1}{N^{(i)}} \sum_{j=1}^{N^{(i)}} x^{(i)}(j)$$

where $N^{(i)}$ is the number of samples in class i. Since the actual centroid generally does not correspond to a point that exists in the sample set. The closest sample point is instead selected

$$\arg\min_{x_j} \left\| x_j^{(i)} - C^{(i)} \right\|$$

The centroids for each class cluster is found and are connected to one another in the manifold graph.

During our experiments with these two schemes, it was found that the second approach was more robust and stable. This may be due to the first approach being susceptible errors caused by dense data sets and outliers. Class clusters are often overlapping. So by connecting the two closest points of two class clusters, there would not be much separation between the classes. Even if the clusters are generally well separated, one outlier falling near the center of an interclass cluster will result in manifold distances that are skewed to be shorter than expected.

Also, the second approach was found to be less computationally intensive than the first. This can be explained in the two class problem. Consider two class clusters with N_1 and N_2 samples. Since the first problem requires a distance calculation between every sample of both class, the computations required is $O(N_1 * N_2)$. On the other hand, the second method's centroid calculation requires $O(N_1 + N_2)$ calculations. This trend compounds further when more classes are considered. Because of these reasons, the second inter-class connection approach is used in the experiments presented in Chapter 4.

Chapter 4

VALIDATION ON BENCHMARK DATA SETS

In this chapter, experiments were conducted using the methodology described in Chapter 3. The goal is to first validate the sparsity enhancements of the graph construction and embedding steps. A variety of benchmark data sets are used to test the two modules. Next, experiments were conducted on the full proposed approach with the combination of the graph construction and embedding alterations. In this case, popular image data sets were used. In further chapters, the proposed system is applied to a MRI brain tumor classification and progression prediction data set.

4.1 UCI DATA SETS

The first set of benchmark data used are from University of California Irvine's (UCI) machine repository [56]. Since multiple data sets are used from this repository, it is useful to describe each data set all at once. The UCI repository consists of many data sets suitable mainly for classification. Of the 188 classification data sets offered in the repository, five were chosen due to their use of quantitative data versus qualitative values and their prolific use in the literature.

wine: The wine data set has 13 variables. Essentially, this is a classification problem to identify three kinds of wine cultivars using the provided data. These attributes include alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and proline. This data set contains 178 samples. Chemical analysis were performed on the wines grown in the same region in Italy but derived from three different cultivars. The classification is to identify the cultivar from the wine it produced.

wdbc: The Wisconsin diagnostic breast cancer (wdbc) data set is a classification data set with two classes, malignant and nonmalignant. The data set includes 569 cell samples with each sample represented by 30 variables. Of which, ten variables are real-valued features computed from each cell nucleus. The class distribution of

this data set is 357 benign and 212 malignant samples. The data set was created by University of Wisconsin's Dr. William H. Wolberg and include patients seen by Dr. Wolberg since 1984. The data was taken from patients exhibiting invasive breast cancer but with no evidence of distant metastases at the time of diagnosis.

ionosphere: The ionosphere data set was collected by a radar system in Goose Bay, Labrador of Canada. Recall that the ionosphere is the term for a region of the upper atmosphere from 53-370 miles in altitude. The goal of this data set is to determine if the radar signals return a signal of structures in the ionosphere. A "Good" labeled instance represents a radar signal with evidence of some type of structure while "Bad" labeled instances do not and have a signal that pass through the ionosphere. This data set contains 351 samples with 34 variables. The radar system consists of 16 high-frequency antennas that uses a total of 6.4 kilowatts of power.

musk: The musk data set is another classification data set. This time, the classification is to separate molecules as musks and non-musks. The data set started with 92 molecules that were judged by human olfactory experts to belong to each class. Of the original 92 molecules, 47 were determined to be musks while the remaining 45 were non-musks. To represent the molecule, 166 variables that describe the molecules shape, conformation, and size are selected thus resulting in 166 dimensions. Since molecular bonds can rotate, a single molecule can be represented in many different shapes. By considering the rotations, 476 molecule combinations are generated. So in summary, this binary classification data set contains 476 samples each with 166 features.

sonar: The sonar data set is interested in classifying sonar signals bounced off a metal cylinder versus those bounced off a roughly cylindrical rock. The data set contains 208 samples, of which 111 are of a metal cylinder at various angles and under various conditions while 97 are from rocks under similar variations. A frequency-modulated chirp was transmitted to the object while incrementally raising the chirp frequency. In total, each sample is represented by 60 features ranging from 0 to 1.0. The feature represents the energy of a particular frequency band over a period of time.

4.1.1 ℓ_1 GRAPH CONSTRUCTION EXPERIMENT

First, the graph construction module of the proposed ℓ_1 method is tested using data sets from the UCI repository. In each experiment, $\frac{2}{3}$ of the samples are used for training. The remaining $\frac{1}{3}$ of the samples are used for testing. The goal of this experiment is to compare the graph construction of Isomap using the standard approach versus the proposed adaptive neighborhood selection. As this experiment focuses only on the graph construction and not other modules such as sampling or embedding, the separate modules are avoided by applying manifold learning on the entire data set. In this way, no sampling occurs and since there are no out-of-sample points, there is no embedding that needs to be done.

Classification was performed on the learned manifold using k-nearest neighbor classification with $k = 5$. The number of nearest neighbors for graph construction is varied from 5 to 15 with each data set. In the case of the proposed ℓ_1 neighbor selection approach, the number of nearest neighbors represents the size of the maximum size of the neighborhood selected. Due to the sparsity property of the proposed graph construction, the actual number of nearest neighbors may be smaller. The parameter controlling the sparsity, λ of the ℓ_1 term from Equation 1 was set to $\lambda = 0.1$. Each experiment was conducted 10 times and the average classification accuracy was recorded.

Figures 8-12 show the results from data sets in the UCI repository. In each of the figures, the vertical axis represents the average classification accuracy over 10 trials. The horizontal axis is the number of nearest neighbors used for the graph construction step. The blue solid line represents the standard Isomap approach with nearest neighbor graph construction. The red line with "o" markers represents the proposed ℓ_1 enhanced graph construction method.

The results proved to be promising. The proposed graph construction method showed improvement compared with standard Isomap. In Figures 9, 10, and 12 were consistently improved. Figures 8 and 11 showed a comparable result between the two methods. Thus, the proposed method either performs better or comparable to the original. Also from the figures, it can be seen that the proposed method is more robust to the number of nearest neighbors selected. For example in Figure 10, the classification accuracy for classical Isomap degrades after 8 nearest neighbors while the accuracy of the proposed remains relatively stable. The performance drop of the standard Isomap approach may be due to the manifold leakage. This suggests that
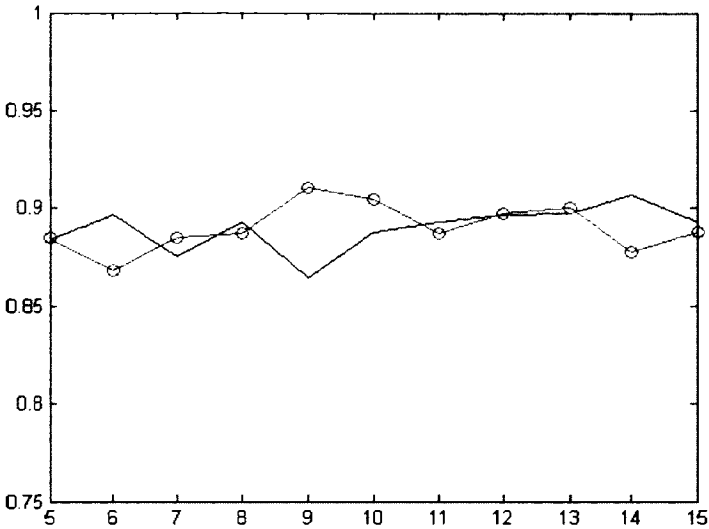
Figure 8: Classification accuracy for ionosphere data set. The red line with "o" line markers denotes proposed method. Blue line represents Isomap.
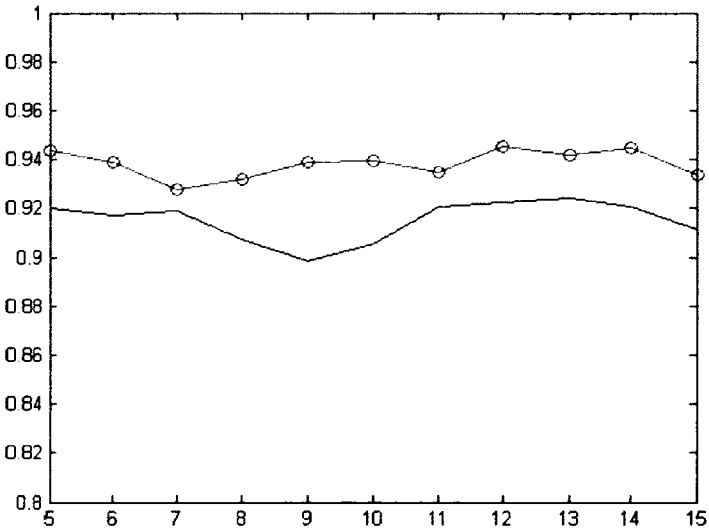
Figure 9: Classification accuracy for wdbc data set. The red line with "o" line markers denotes proposed method. Blue line represents Isomap.
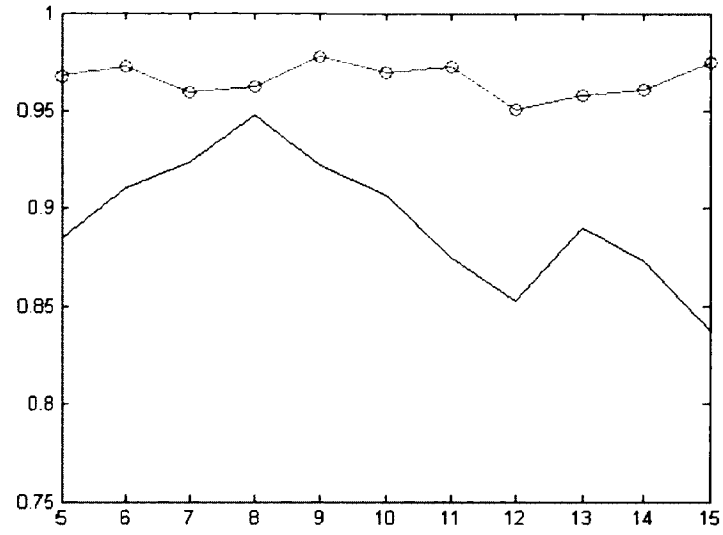
Figure 10: Classification accuracy for wine data set. The red line with "o" line markers denotes proposed method. Blue line represents Isomap.
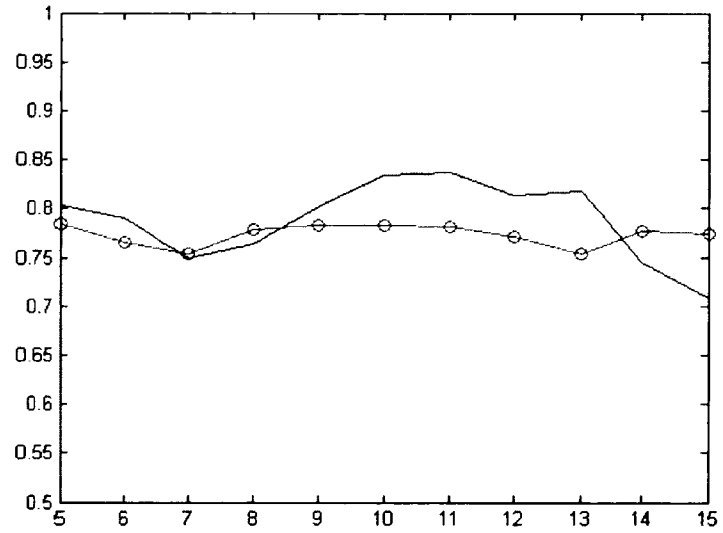
Figure 11: Classification accuracy for musk data set. The red line with "o" line markers denotes proposed method. Blue line represents Isomap.
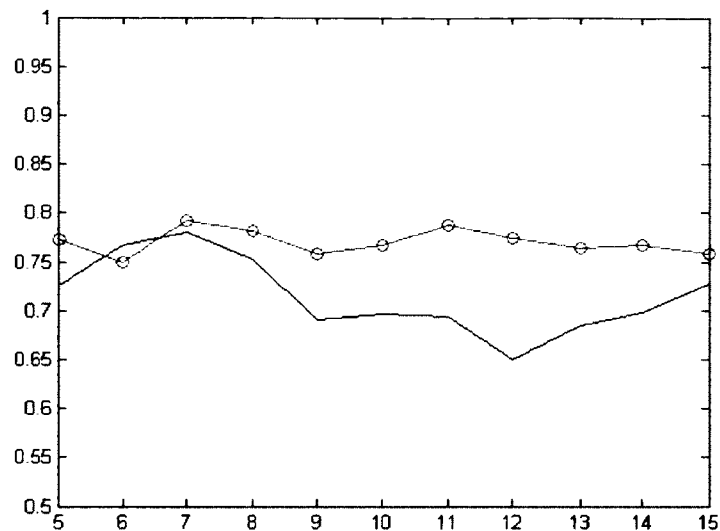
Figure 12: Classification accuracy for sonar data set. The red line with "o" line markers denotes proposed method. Blue line represents Isomap.

the ℓ_1-based approach is effective in overcoming the leaking problem.

4.1.2 ℓ_1 GRAPH CONSTRUCTION RESULTS VERSUS SLPP

Another experiment was conducted testing the efficacy of the proposed graph construction versus another sparse projection approach. Recall, sample-dependent locality preserving projection (SLPP) described in Section 2.3.2. Like the proposed method, this dimensionality reduction technique is also focused on preserving sparse neighborhood relationships.

This experiment was conducted matching the same specifications as presented in the SLPP literature [1]. The wine data set was reduced to a two dimensional projection using various dimensionality reduction techniques. As the methods being compared were all unsupervised, the unsupervised version of the proposed method is used in this experiment. Figure 13 shows the results. Going from left to right and top to bottom, the methods used in this test were the proposed approach, SLPP, locality preserving projection (LPP) [46], unsupervised discriminant projection (UDP) [57], neighborhood preserving embedding (NPE) [58], and PCA. The plots for all methods except the proposed approach were generated by [1]. A reuse license for this figure

Table 1: Wine results

Method	Accuracy (Std Dev)
Raw	82.6 (.23)
Isomap	91.6 (.21)
ℓ_1-Isomap	97.8 (.41)
Supervised Isomap	96.6 (.14)
Supervised ℓ_1-graph	97.8 (.25)

is given in the Appendix. The paper left no quantitative results but rather emphasized the separation between classes. Judging from this metric, it is apparent that the proposed method provides a better separation between the three classes when compared to the other five methods.

Classification was applied on the data set to produce numeric results. The graph construction method was tested against Isomap and variations of the proposed method. k-nn classification with $k = 5$ was implemented as the classification algorithm. The sparsity parameter λ was set to 0.1.

Table 1 shows the average results over the 100 trials between the tested methods. Here, Raw represents the results with no manifold learning. The classification was applied directly on the 13-dimensional original data set. This gives a baseline for comparison. Basic Isomap showed an improvement over the baseline suggesting that the data set is indeed nonlinear in nature. The proposed ℓ_1-enhanced Isomap shows an improvement on the regular Isomap approach with 0.987 versus 0.916 accuracy. Next, the supervised graph construction method described in Section 3.2.5 was employed. Recall this method only connects same-class neighbors during the graph construction step. This improves the classification accuracy of Isomap to 0.966 while the supervised ℓ_1-graph construction showed no difference. This is likely because the proposed unsupervised ℓ_1-graph construction already tends to select same class neighbors. The separation of classes seen in Figure 13 also support this claim. Figure 14 shows the first two principle dimensions using the proposed unsupervised ℓ_1 method, the supervised version of Isomap, and the supervised version of the ℓ_1 method. The separation of classes in each case is evident. Again, it is clear that the unsupervised ℓ_1 graph construction creates a similar manifold as the supervised approaches.
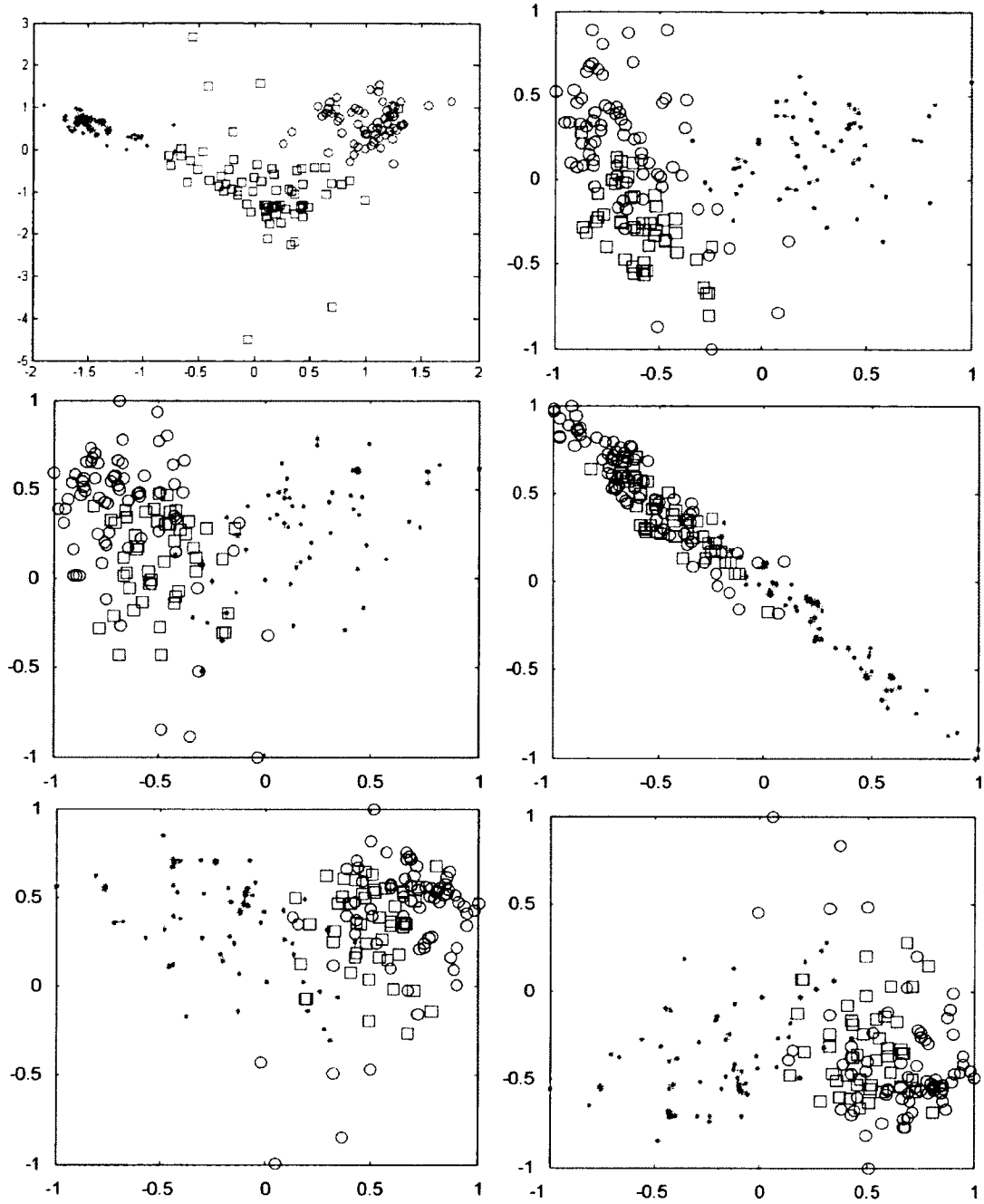
Figure 13: First 2 principle dimensions of wine data set. Left to right, top to bottom: Proposed method, SLPP, LPP, UDP, NPE, PCA. See Appendix for figure reuse license [1].
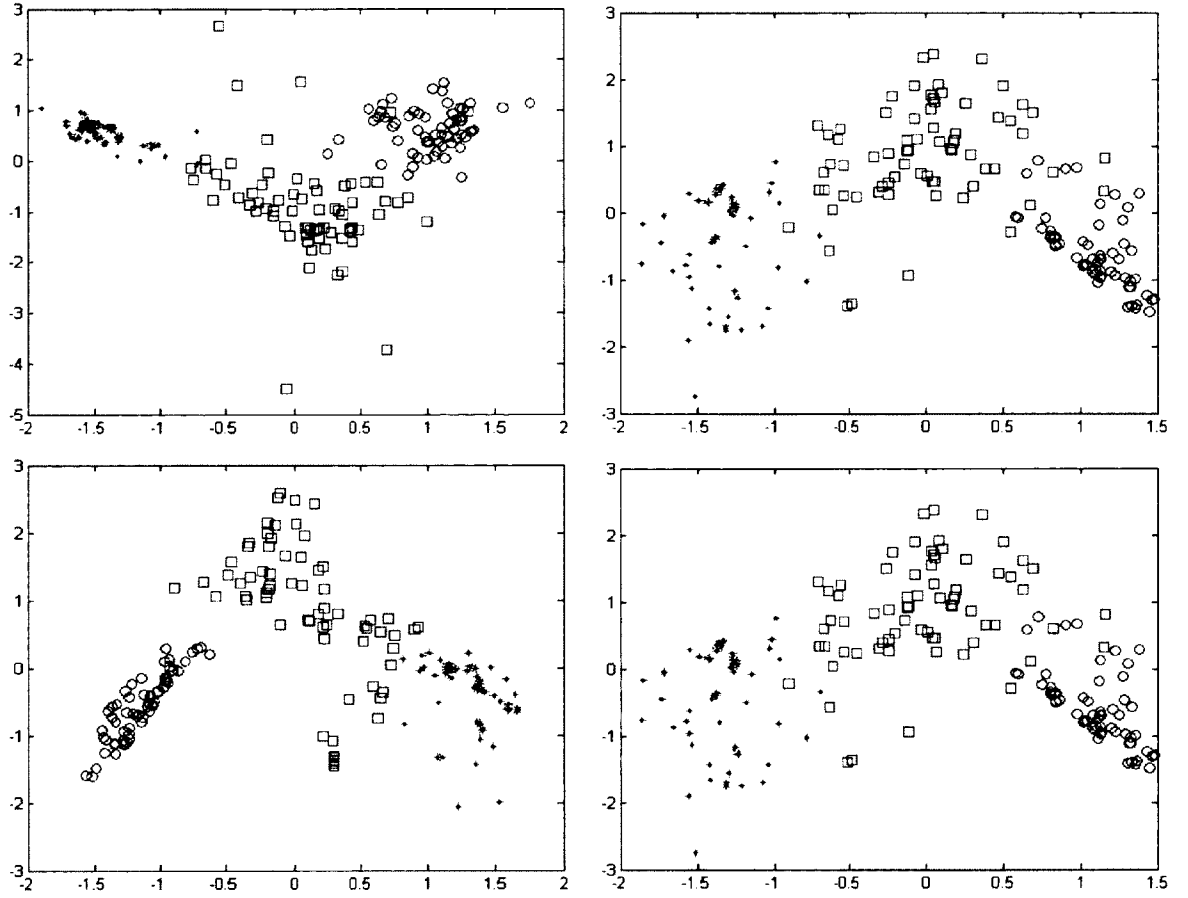
Figure 14: First 2 principle dimensions of wine data set. Top left, ℓ_1 manifold space. Top right, Isomap manifold space. Bottom left, supervised ℓ_1 manifold space. Bottom right, supervised Isomap manifold space

4.1.3 ℓ_1 EMBEDDING EXPERIMENTS

Next, experiments on the ℓ_1 embedding portion of the proposed method are tested using the UCI data sets described in Section 4.1. The goal of this experiment is to compare embedding using LLE versus the proposed ℓ_1-based sparse approach. In contrast with the graph construction test, a subset of the samples are randomly selected for manifold learning. In each experiment, $\frac{1}{3}$ of the data set is used by standard Isomap to compute a manifold. The remaining $\frac{2}{3}$ of the samples are embedded into the manifold using the tested method.

Classification was performed on the learned manifold using k-nearest neighbor classification with $k = 5$. The number of nearest neighbors for embedding is varied from 5 to 15 with each data set. In the case of the proposed ℓ_1 neighbor selection approach, the number of nearest neighbors represents the size of the maximum size of the neighborhood selected for reconstruction into manifold space. Due to the sparsity property of the proposed embedding approach, the actual number of nearest neighbors may be smaller. The sparsity parameter from Equation 2 for these experiments were set to $\lambda = 0.01$. Each experiment was conducted 10 times and the average classification accuracy was recorded.

Figure 15-19 show the results from data sets in the UCI repository. In each of the figures, the vertical axis represents the average classification accuracy over 10 trials. The horizontal axis is the number of nearest neighbors used for the embedding step. The blue solid line represents the standard LLE approach. The red line with "o" markers represents the proposed ℓ_1 enhanced embedding.

Figures 15, 16, and 17 show that the proposed embedding performed better than the standard embedding. But in Figures 18 and 19, the two approaches have comparable results. Note that the classification accuracy for these two data sets are lower than the others. This may be due to a problem in creating the manifold. Since these data sets are small, the 1/3 of samples used to create the manifold may not be enough to achieve a successful manifold skeleton. A conclusion that can be pulled from these experiments is that the proposed ℓ_1 based neighborhood selection for embedding generally performed better than standard LLE for embedding after Isomap.
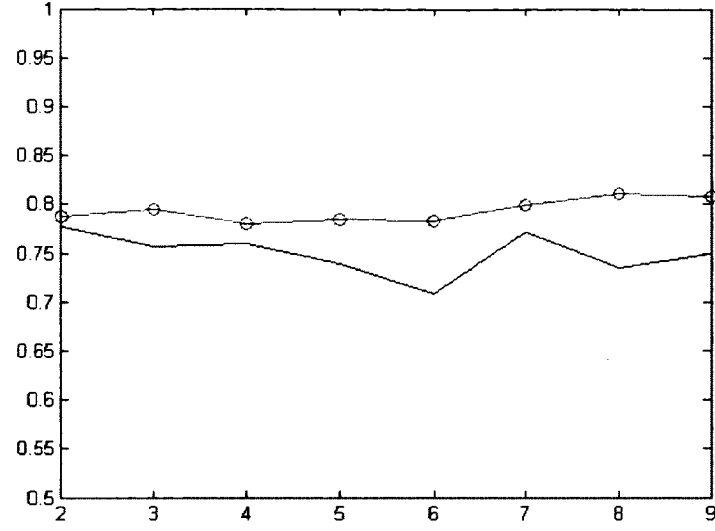
Figure 15: Classification accuracy for ionosphere data set versus the number of near-est neighbors for embedding. The red line with "o" line markers denotes embedding with ℓ_1-embedding. Blue line represents embedding with LLE.
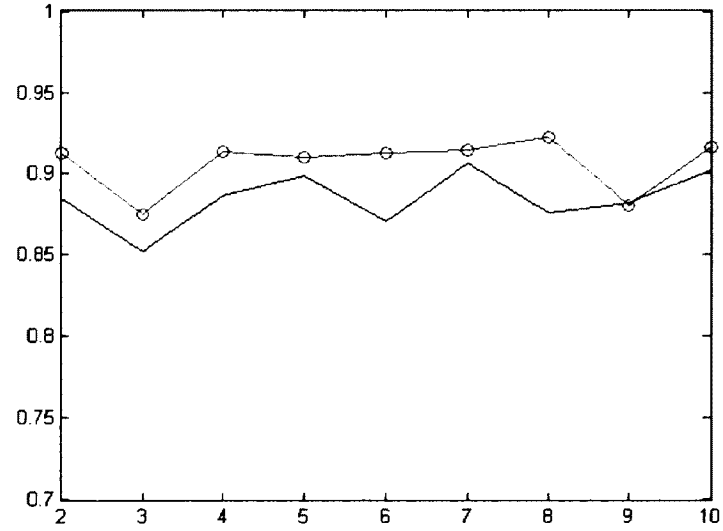
Figure 16: Classification accuracy for wdbc data set versus the number of nearest neighbors for embedding. The red line with "o" line markers denotes embedding with ℓ_1-embedding. Blue line represents embedding with LLE.
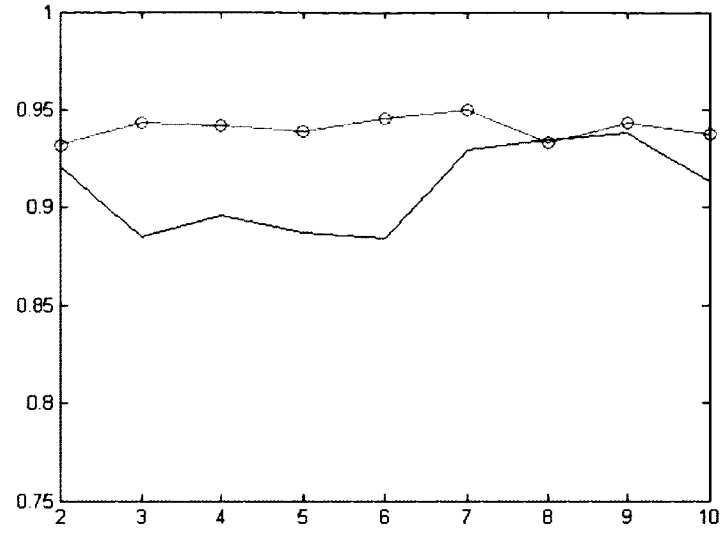
Figure 17: Classification accuracy for wine data set versus the number of nearest neighbors for embedding. The red line with "o" line markers denotes embedding with ℓ_1-embedding. Blue line represents embedding with LLE.
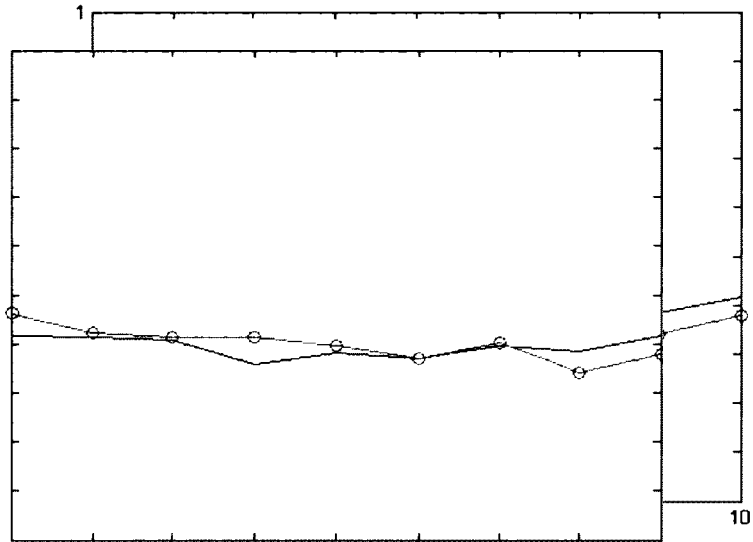
Figure 18: Classification accuracy for musk data set versus the number of nearest neighbors for embedding. The red line with "o" line markers denotes embedding with ℓ_1-embedding. Blue line represents embedding with LLE.
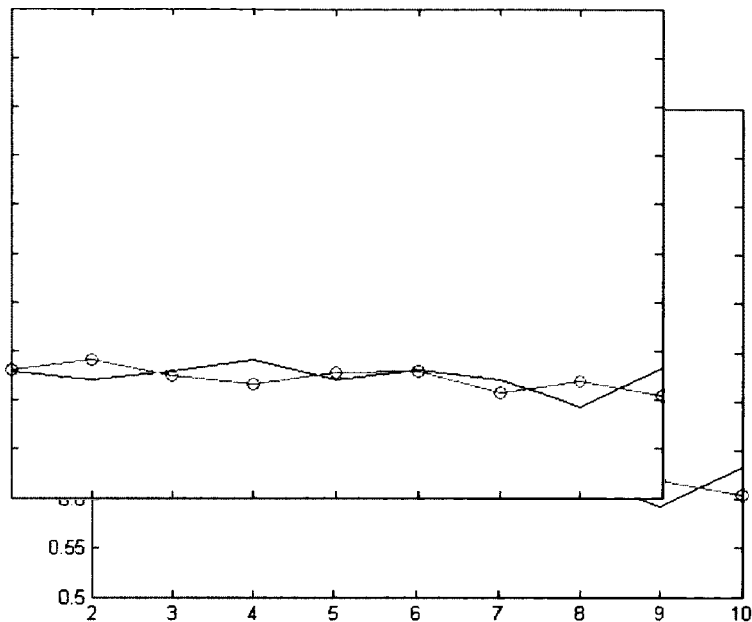
Figure 19: Classification accuracy for sonar data set versus the number of nearest neighbors for embedding. The red line with "o" line markers denotes embedding with ℓ_1-embedding. Blue line represents embedding with LLE.

4.1.4 ℓ_1 EMBEDDING RESULTS WITH HELIX

Along with the UCI experiments on the embedding module, another data set was tested to provide additional results. A 5000-sample synthetic helix data set was generated. The helix is a one dimensional line embedded into a spiral shape in three dimensional space. Figure 20 shows an example of a helix used in this experiment. The data set is comprised of two classes, shown as blue and red in the figure.

In this experiment, 1500 of the 5000 samples are used to create the manifold using LTSA. Recall LTSA is described in Section 2.1.2. This method was used over Isomap because it offered a faster computational time. The remaining 3500 are embedded into the manifold skeleton. k-nearest neighbor classification is used to determine the class of the embedded samples where $k = 1$. The experiment was repeated over 30 trials. Table 2 shows the results of this experiment in terms of average classification accuracy. The number in parenthesis is the standard deviation. Figure 21 shows an example unfolded manifold for LLE and ℓ_1-enhanced embedding.

Figure 21 shows an example result from each of the two methods. On the left of the figure shows the manifold found with ℓ_1-embedding that is correctly unfolded.
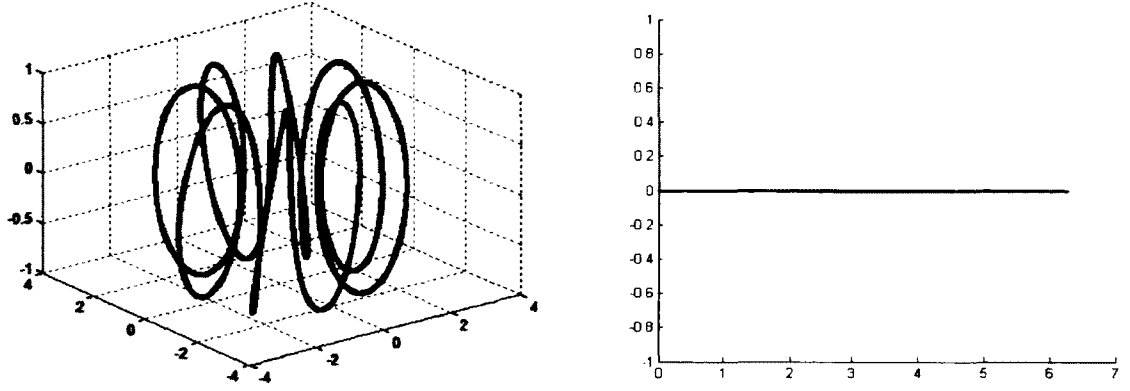
Figure 20: An example generated for the helix data set. The three dimensional helix (left) and the intrinsic 1-dimensional representation (right)
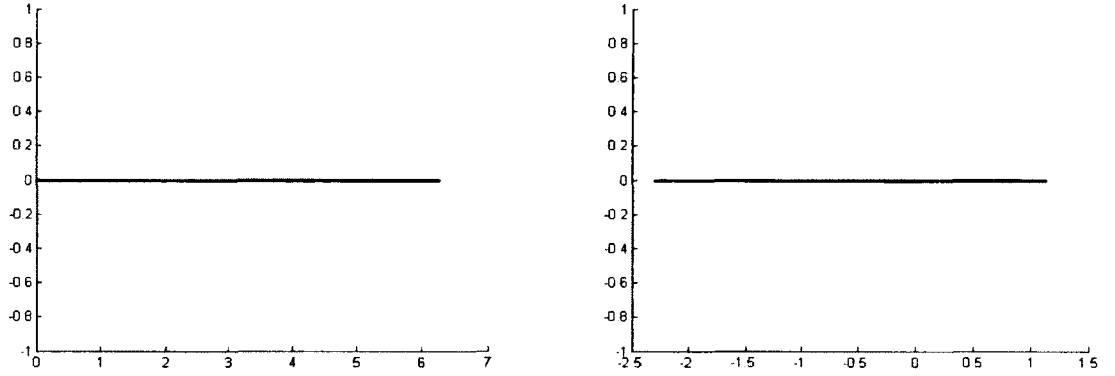
Figure 21: Left, helix manifold found with ℓ_1 embedding. Right, helix manifold found with LLE embedding

On the right is an incorrectly unfolded manifold using the original method. The results show that ℓ_1 embedding performed marginally better than LLE.

4.2 RESULTS OF NEW PROPOSED METHOD ON IMAGE DATA SETS

In the previous section, numerical data from the UCI machine learning repository are used to individually test the sparse ℓ_1-enhanced graph construction and embedding steps. Next, the full proposed manifold learning system is applied to three image data sets; the University of Manchester Institute of Science and Technology (UMIST)

Table 2: Helix data set classification accuracy for embedding experiment. Standard deviation is shown in parenthesis.

Experiment	Accuracy
LLE	.878 (.082)
ℓ_1 embedding, $\lambda = 0.01$	.907 (.084)
ℓ_1 embedding, $\lambda = 0.03$	.930 (.060)
ℓ_1 embedding, $\lambda = 0.05$	.898 (.073)
ℓ_1 embedding, $\lambda = 0.07$	.915 (.068)

face database, the Columbia University Image Library object database (COIL-20), and the Yale University face database (Yale). These image data sets were selected because of their non-linearity and prevalence in manifold learning literature.

4.2.1 UMIST

The UMIST face data set [59] is a common benchmark data set used in machine vision testing. The data set contains 564 face images of 20 subjects. The subjects vary in race, gender, and appearance. The subjects are photographed while varying the pose angle. Figure 22 shows sample images from one individual of the data set. The files are originally approximately 220×220 pixels with 256-bit grey-scale. To make the images more uniform, each image was down-sampled to 40×40 pixels. This was then converted into a vector by stacking the columns. Thus, each sample contains 1600 dimensions.

This experiment was modeled after an experiment conducted in the literary work presenting DONPP [27]. For this data set, a small number of samples are randomly selected from each subject. Separate experiments are taken with 5, 7, and 9 training samples for each subject. Since there are 20 subjects/classes in this data set, the number of training samples are 100, 140, and 180 respectively. Isomap is used to find the manifold from the training samples using the ℓ_1 graph construction method described in the proposed method. The remaining data samples are embedded into the manifold skeleton using the ℓ_1 embedding technique. Classification is performed using k-nn nearest neighbor classification with $k = 1$. Each experiment is repeated 10 times. The average classification accuracy and standard deviation is recorded.

Table 3 shows the results. We calculated the results for PCA, LTSA, standard

Figure 22: Sample images of one subject in the UMIST face database
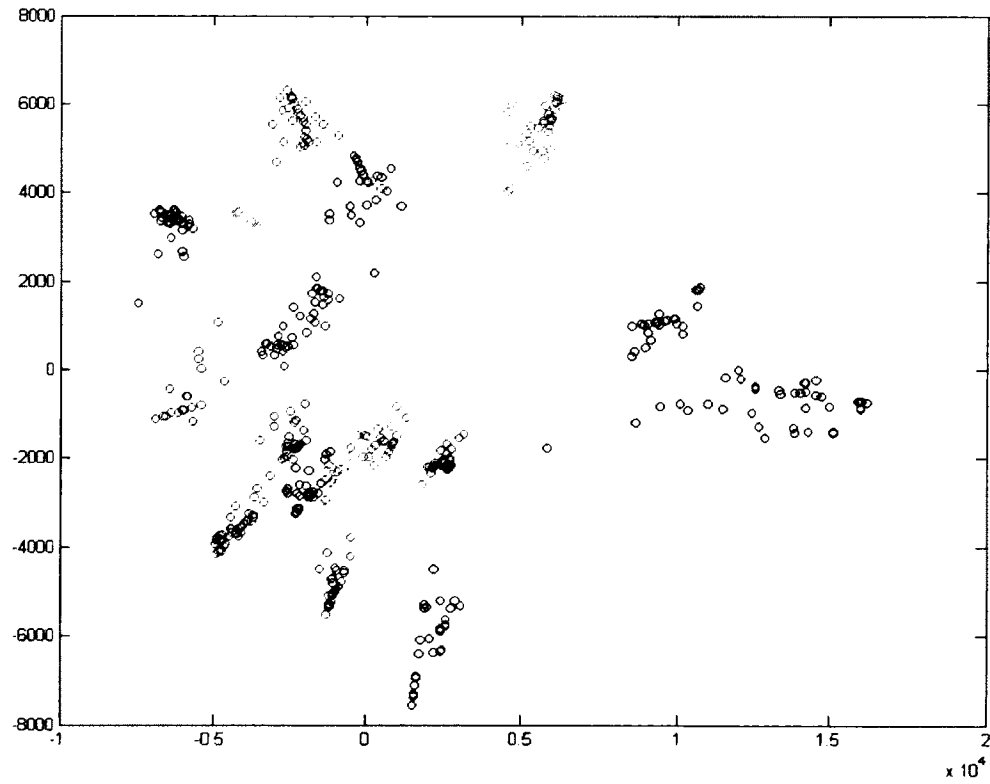
Figure 23: First 2 dimensions of manifold space for UMIST

Table 3: UMIST results

Method	5 Training Samples	7 Training Samples	9 Training Samples
PCA	83.68 (1.34)	89.89 (2.42)	93.19 (1.95)
LTSA	81.66 (2.79)	88.90 (2.28)	91.72 (2.51)
Isomap	79.69 (2.87)	86.67 (3.17)	88.80 (1.52)
LLE	78.75 (7.33)	85.66 (2.17)	95.95 (3.83)
LDA*	88.51	93.31	95.14
LPP*	86.06	91.36	93.44
NPE*	86.53	91.52	94.30
MFA*	92.61	94.28	96.20
ONPP*	92.34	95.89	97.52
DONPP*	**93.27**	**96.85**	**98.17**
Proposed ℓ_1	92.11 (1.81)	96.41 (1.25)	98.00 (0.95)

Isomap, LLE, and the Proposed approach. The methods labeled with an asterisk were performed by [27] and are presented as reference. These values are presented without a standard deviation. Figure 23 shows the first two dimensions for the manifold of the proposed approach. Each color represents a different class. The separation of clusters suggest that manifold is unfolded correctly.

From Table 3, the proposed approach showed a marked improvement over classical Isomap, LTSA, along with the linear methods. It showed a comparable result to DONPP.

4.2.2 COIL-20 DATABASE

The COIL-20 [60] image database consists of objects placed on a motorized turntable against a black background. Images were taken while the object was rotated 360 degrees. Thus, the object's pose is altered. A total of 72 images were taken of each object with an rotation interval of 5 degrees. Figure 24 shows sample images of the 20 objects in the COIL-20 database. To normalize the images in this data set, each image is down-sampled to 40×40 pixels.

This experiment was modeled after an experiment conducted in the literary work presenting DONPP [27]. For this data set, a small number of samples are selected for each subject. Separate experiments are taken with 5, 10, and 15 training samples for each subject. Since there are 20 objects/classes in this data set, the number

Figure 24: Sample images of 20 objects in COIL-20

of training samples are 100, 200, and 300 respectively. Isomap is used to find the manifold from the training samples using the ℓ_1 graph construction method described in the proposed method. The remaining data samples are embedded into the manifold skeleton using the ℓ_1 embedding technique. Classification is performed using k-nn nearest neighbor classification with $k = 1$. Each experiment is repeated 10 times. The average classification accuracy and standard deviation is recorded.

Table 4 shows the results. We calculated the results for PCA, LTSA, standard Isomap, LLE, and the Proposed approach. The methods labeled with an asterisk were performed by [27] and are presented as reference. These values are presented without a standard deviation. Figure 25 shows the first two dimensions for the manifold of the proposed approach. Each color represents a different class. The separation of clusters suggest that manifold is unfolded correctly.

From Table 4, the proposed approach again showed a marked improvement over classical Isomap. The results show that the proposed approach is better than all methods tested except for DONPP, which produced a comparable result. Compared to other methods in each test, the proposed approach's classification accuracy is at least one standard deviation higher. For 5 training samples per class, DONPP is marginally better while the proposed approach is marginally better for 10 training samples per class. The classification accuracy of these two methods are the same for 15 training samples.
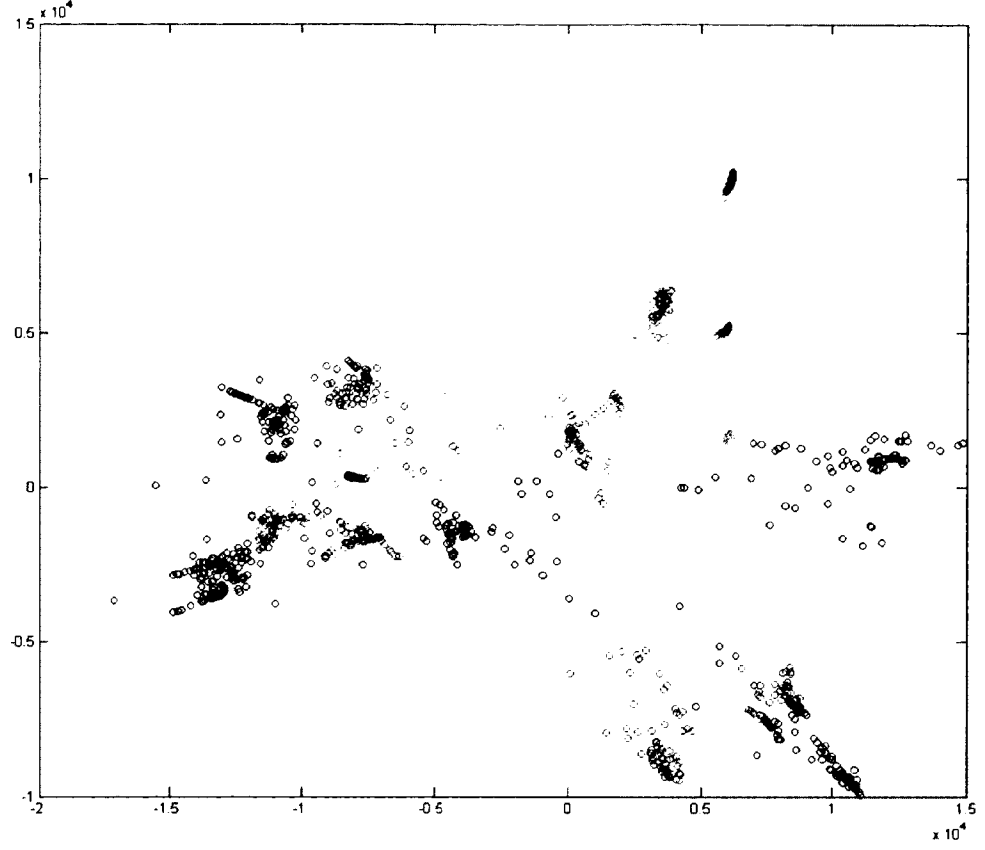
Figure 25: First 2 dimensions of manifold space for COIL-20

Table 4: COIL-20 results

Method	5 Training Samples	10 Training Samples	15 Training Samples
PCA	81.88 (2.39)	89.14 (1.00)	93.43 (0.73)
LTSA	53.99 (3.09)	55.86 (2.68)	55.45 (3.79)
Isomap	75.11 (3.19)	83.29 (2.76)	88.87 (1.35)
LLE	72.06 (4.07)	84.40 (7.04)	91.73 (0.98)
LDA*	80.82	87.24	91.76
LPP*	80.08	86.82	91.32
NPE*	80.43	87.15	91.45
MFA*	81.94	88.25	90.70
ONPP*	85.82	91.83	94.81
DONPP*	**87.36**	94.11	**97.04**
Proposed ℓ_1	87.17 (1.30)	**94.37 (1.32)**	**97.04 (0.57)**

Figure 26: Sample images of one subject in the Yale face database

4.2.3 YALE DATABASE

The Yale database [61] is comprised of 15 subjects. The samples within each subject vary with facial expression and lighting. There are 11 images for each subject for a total of 166 sample images. For each subject, there is one image per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. A sample set of sample images for one subject of the Yale database is displayed in Figure 26. As with the UMIST and COIL-20, the Yale database is also down-sampled to 40×40 pixels.

This experiment was modeled after an experiment conducted in the literary work presenting DONPP [27]. For this data set, a small number of samples are selected for each subject. Separate experiments are taken with 3, 5, and 7 training samples for each subject. Since there are 15 objects/classes in this data set, the number of training samples are 45, 75, and 105 respectively. Isomap is used to find the manifold from the training samples using the ℓ_1 graph construction method described in the Proposed method. The remaining data samples are embedded into the manifold skeleton using the ℓ_1 embedding technique. Classification is performed using k-nn nearest neighbor classification with $k = 1$. Each experiment is repeated 10 times. The average classification accuracy and standard deviation is recorded.

Table 5 shows the results. We calculated the results for PCA, LTSA, standard Isomap, LLE, and the Proposed approach. The methods labeled with an asterisk were performed by [27] and are presented as reference. These values are presented without a standard deviation. Figure 27 shows the first two dimensions for the manifold of the proposed approach. Each color represents a different class. The separation of clusters suggest that manifold is unfolded correctly.

Table 5: Yale face database results

Method	3 Training Samples	5 Training Samples	7 Training Samples
PCA	74.74 (2.65)	79.01 (3.49)	82.46 (3.37)
LTSA	69.67 (5.40)	72.2 (5.60)	80.98 (2.81)
Isomap	68.43 (3.44)	73.63 (3.70)	76.07 (4.18)
LLE	69.32 (7.06)	71.98 (4.49)	76.90 (6.81)
LDA*	64.08	72.78	80.83
LPP*	67.00	73.44	82.33
NPE*	68.40	74.33	81.17
MFA*	64.33	73.44	82.67
ONPP*	67.90	77.00	82.83
DONPP*	68.75	77.44	84.33
Proposed ℓ_1	**81.07 (2.11)**	**85.05 (2.50)**	**88.03 (2.45)**
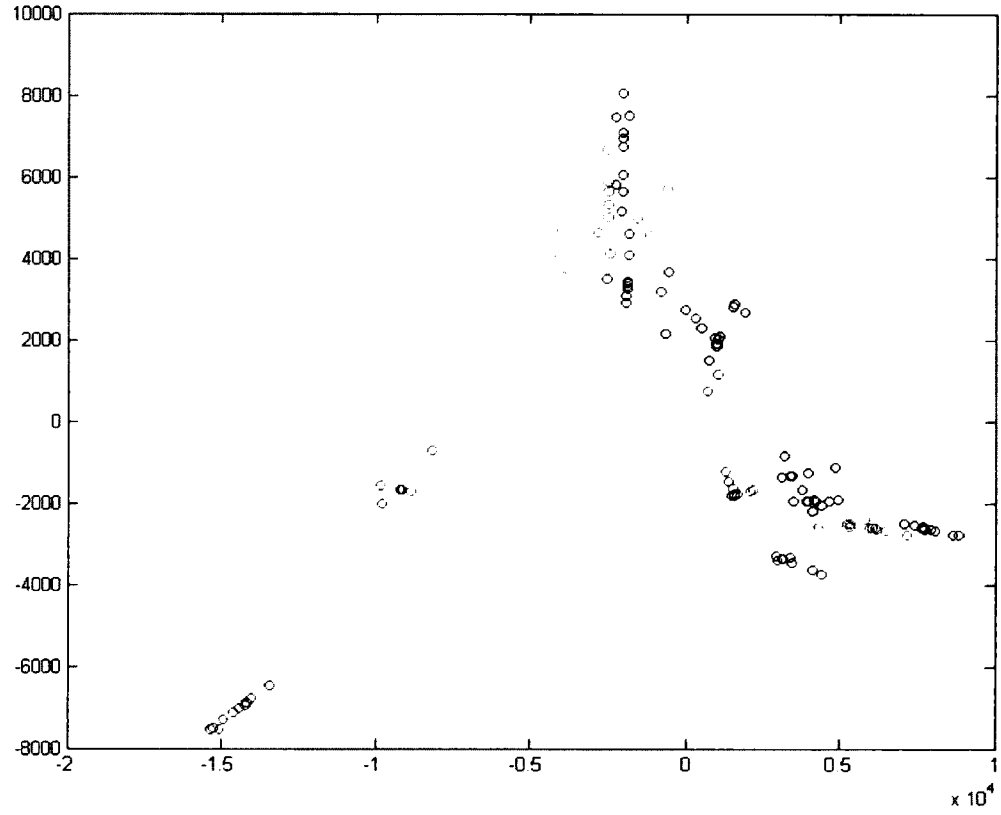
Figure 27: First 2 dimensions of manifold space for yale

For this final image data set, the proposed approach produces a higher classification accuracy than the other methods in each of the tests.

4.3 BENCHMARK RESULTS SUMMARY

In summary, the proposed ℓ_1-enhanced manifold learning shows a considerable improvement to the standard Isomap approach in the benchmarks. The proposed system also has comparable results to DONPP for two image data sets while performing considerable better in one of the tested image data sets.

Chapter 5

APPLICATION TO MRI DATA SET

In this chapter, the process of reducing a large-scale high-dimensional data set into a low-dimensional manifold is discussed with a focus on MRI data. The incremental approach from Section 3.1 is used. The adaptive ℓ_1 based enhancements are applied to this data set in Chapter 6. This application incorporates multiple signal processing methods including standardization, sampling, dimensionality reduction, feature selection and classification [62].

5.1 MRI EXPERIMENT INTRODUCTION

With the rapid advancement of diagnostic imaging technology, multi-dimensional, large-scale, and heterogeneous medical data sets are generated routinely in clinical imaging exams. For instance, magnetic resonance (MR) diffusion tensor imaging (DTI) has become a routine component of the brain MR imaging exams in many institutions. Together with the traditional T1, T2 or fluid attenuated inversion recovery (FLAIR) weighted MRI scans, this new imaging modality has provided additional information and has shown potential for better brain tumor diagnosis. However, interpreting these large-scale, high-dimensional data sets simultaneously is challenging [63, 64]. For example, quantitative maps of apparent diffusion coefficients (ADC) derived from DTI imaging have been reported as useful indicators in distinguishing tumor tissue from surrounding edema by Sinha et al [65]. But other researchers found that the differentiation has not always been successful [66, 67, 68]. Lam et. al reported that ADC values were not useful in identifying tumor types [69]. These claims have been challenged by a number of researchers whose results have shown that high-grade gliomas have lower ADC values while low-grade ones are opposite [67, 70, 68, 71]. Analysis performed on fractional anisotropy (FA) values also showed contradictory results in FA's ability for differentiation of enhancing tumor from edematous brain cancer [65]. Similar situations can also be found in meningiomas diagnosis. A significant difference in peritumoral ADC and FA values between low-grade meningiomas and high-grade gliomas has been reported by Bastin et al. [72]. This

possibly reflects the presence of tumor-infiltrated edema in gliomas. On the other hand, Lu et al. showed that there is no statistically significant difference of ADC and FA values between intra- and extra-axle lesions [73]. No significant difference was found for ADC values between peritumoral hyperintense regions or peritumoral normal-appearing white matter and high-grade gliomas [74].

The above contradictory results may be due to methodological variations which include, for example, mismatch of ADC values with the biopsy examination results [63], the use of diffusion gradients applied in a single direction instead of along three orthogonal directions, heterogeneous study group, differently categorized tumor types, patients having previous surgeries, adjuvant therapy, or use of steroids [64]. A single MRI scan may not provide enough discriminating power to reliably differentiate various tissues. Experiments using large-scale high-dimensional data sets from our recent studies [75, 76] have shown that one can efficiently differentiate brain tumor tissue from tissues in progressed and normal regions. In the previous study, we integrate all available MRI scans into a high-dimensional data set and perform a classification task in the high-dimensional space. Our results [75, 76, 77] indicated that an individual MRI scan is probably insufficient to reliably distinguish different tissue types. But taking advantage of multiple MRI images may lead to a successful classification because those different tissue types are located in different regions with possible overlaps in the constructed high-dimensional space.

Due to the fact that significant correlations exist among these multi-dimensional images, a hypothesis is made that low-dimensional geometry data structures (manifolds) are embedded in the high-dimensional space. Those manifolds might be hidden from human viewers because it is challenging for human viewers to interpret high-dimensional data. When correctly extracted from the high-dimensional space, the hidden manifolds may provide particularly useful information for brain cancer studies. For example, one may investigate the residence of cancer and normal tissues on the manifolds to derive rules to accurately classify cancer regions. Moreover, the bridge manifolds connecting the cancer and normal tissue manifolds may provide hints for identifying cancer progression trajectory. This knowledge can be used for predicting future tumor growth and allow for better patient management. For example, tumor treatment can be adapted depending on the aggressiveness of the tumor growth.

Many manifold learning algorithms essentially perform an eigenvector analysis

on a data similarity matrix whose size is $N \times N$, where N is the number of data samples. The memory complexity of the analysis is at least $O(N^2)$. This is not feasible for very large data sets in terms of both computational and storage requirements for a regular computer. To solve this problem, statistical sampling methods are typically used to sample a subset of data points as landmarks. A skeleton of the manifold is then identified based on the landmarks. The remaining data points can be inserted into the skeleton by a number of methods such as Nystrom approximation, column-sampling, and Local Linear Embedding (LLE) [28, 78, 52]. To keep a faithful representation of the original manifold, effective sampling should be considered. Under-sampling will distort true embedded geometry structures and thus lead to subsequent manifold learning failure. Oversampling may introduce unnecessary noise. For example, landmark multidimensional scaling (MDS) performs poorly for randomly chosen landmarks if the data is noisy (contains outliers) [79]. Also, data may sometimes collapse to a central point in the low-dimensional space if certain "important" samples are missing [52, 80].

In this chapter, we present a large-scale manifold learning schema for brain tumor study. First, a set of landmarks from a large data set was selected based on an importance function learned from the data. Next, a manifold skeleton was learned using the Isomap algorithm [19]. The remaining data points were then inserted into the skeleton using the LLE method [16]. There were several parameters to be optimized including the number of landmarks needed and the number of neighbors in the Isomap algorithm. Two cost functions were designed for optimizing the parameters. The method was applied to MRI data sets from several brain tumor patients aiming to identify normal, tumor and progressed tissues on the manifold.

The contribution of this chapter consists of two parts: 1) the integration of existing methods toward a novel application and 2) a new sampling approach for large-scale manifold learning.

5.2 PROPOSED SYSTEM

Many current nonlinear dimensionality reduction techniques in the literature require an eigen-decomposition of an $N \times N$ matrix. There are roughly 65k (each slice contains 256×256 pixels) high-dimensional data points in one MRI slice. Even segmenting just the brain region can usually yield over 30k points. Thus, applying a nonlinear dimensionality reduction technique directly on the data is computationally
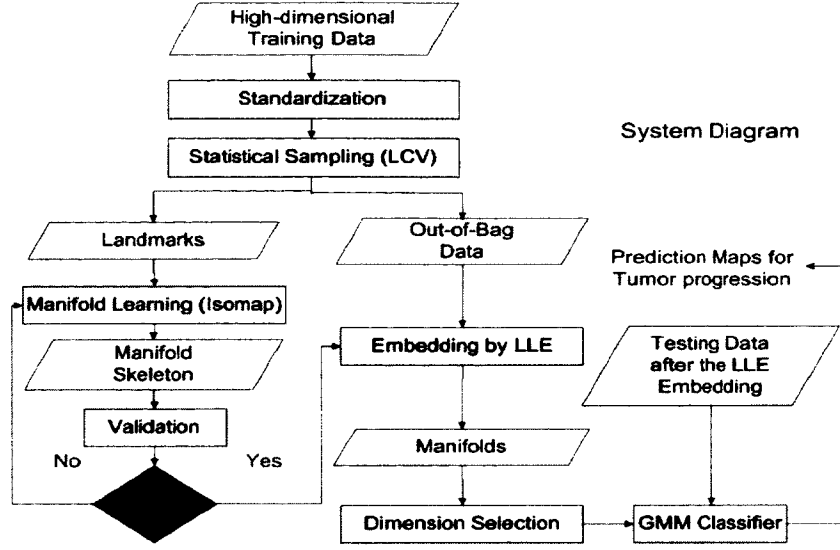
Figure 28: Proposed System Diagram

prohibitive. Therefore, an advanced sampling technique was developed to select a set of landmarks based on local curvature variations. A manifold was learned from the set of landmarks to produce a manifold skeleton. The remaining data points were then inserted into the skeleton using the Local Linear Embedding (LLE) algorithm [16]. After that, a Gaussian Mixture Model (GMM) classifier was trained using sampled points from the tumor and normal regions defined at visit 1. Using the model, a posterior probability map was found for the entire brain region. By adjusting the threshold of this probability map, a tumor region and a progressed tumor region can be predicted. Other classification methods are also reviewed and are proposed for this research effort's remaining work. The system diagram of the proposed framework is shown in Figure 28. In this section, each of these steps are described.

5.3 DATA PREPARATION

The MRI data of brain tumor patients were collected using various MRI scans including FLAIR, T1-weighted, post-contrast T1-weighted, T2-weighted, and DTI.
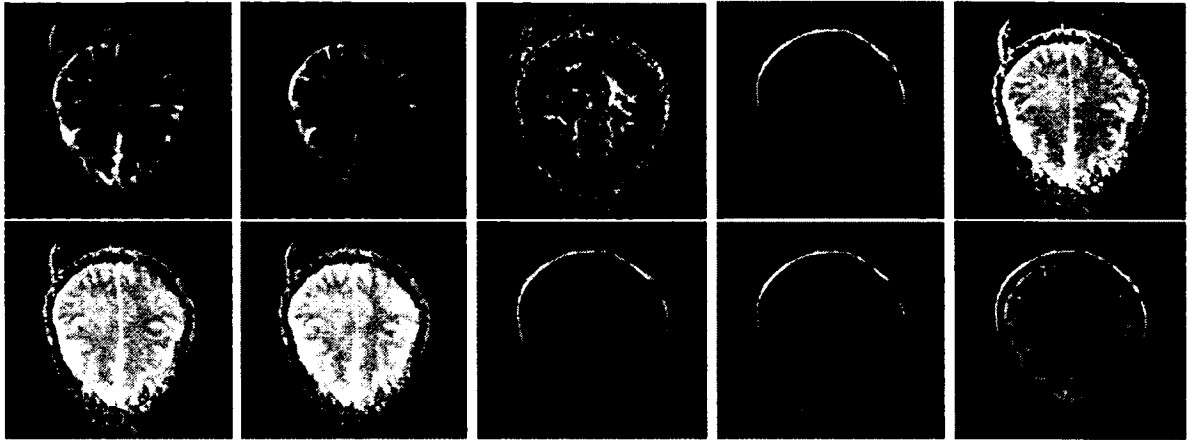
Figure 29: Examples of registered images from the ten MRI series: right to left) ADC, DTI, FA, FLAIR, Max-, Middle-, Min eigenvalues, Post-contrast T1-weighted, T1-weighted and T2-weighted.

Five scalar volumes were also computed from the DTI volume including apparent diffusion coefficient (ADC), fractional anisotropy (FA), max-, min-, and middle- eigenvalues, yielding a total of ten image volumes for each visit of every patient. Each patient went through a series of scans with an interval of one or two months, and a rigid registration was utilized to align all volumes to the DTI volume of the first visit using the vtkCISG toolkit [81]. After registration, each pixel location can be represented by a ten-dimensional feature vector corresponding to the ten MRI volumes. Figure 29 shows images from the ten registered MRI volumes. Two visits were selected in this study and denoted as "visit 1" and "visit 2" where visit 2 showed an expanded tumor region. Hyper-intensity regions were defined on the FLAIR scans as tumors. A similarly sized region far away from the abnormal regions was also defined as a highly confident normal region for training purposes. Figure 30 shows example MRI slices overlaid on the defined tumor and normal regions.

5.3.1 INTENSITY STANDARDIZATION

Different MRI scans do not follow a standard scale. Intensity values in MRI images can vary greatly between separate acquisitions [82]. This is the case even for imagery taken from the same patient, using the same machine, and with the same technician. Intensity scaling was performed as a preprocessing step for MRI images to ensure that the intensity ranges for similar structures are consistent between MRI
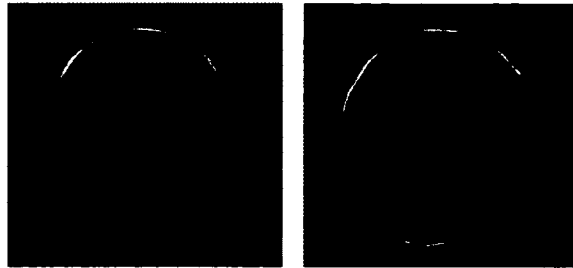
Figure 30: Tumor and normal regions defined for Subject 1 where the red polygon defines abnormal regions and the yellow dotted polygon denotes normal regions. (left) FLAIR image at visit 1. (right) FLAIR image at visit 2 showing a larger tumor region.

acquisitions [83]. This step is essential if direct data comparisons are to be made between data acquisitions. In this study, the g-scale standardization was employed for its robustness to abnormal structures [83]. The presence of abnormal structures such as lesions can create problems for other techniques that scale images using the global histogram [84]. The hyper-intensity region of the abnormal structure can cause an unpredicted increase in the overall intensity of the image histogram and skew standardization values. In contrast, the g-scale standardization finds the largest g-scale region which avoids including abnormal structures. A g-scale region is essentially a large fuzzy connected region. Intensity markers are found within the largest g-scale region to perform a linear transformation to a standard scale. These markers are features derived from the intensity histogram such as the mean, highest intensity, and lowest intensity. The largest g-scale regions of MRI data tend to exclude abnormal structures thus reducing its impact on the standardization. The transformation performed is a piece-wise linear histogram transformation based upon markers found in the largest g-scale region. Figure 32 shows the g-scale transfer function.

The main challenge in the g-scale standardization is to find a consistent structure in MRI images that can give reliable intensity markers in the presence of abnormal structures. The largest g-scale region of a typical MRI scan is a large white matter region [83]. To segment this region, [83] uses a fuzzy selector approach that separates the image into a multitude of fuzzy connected regions. Each of these

regions are referred to as g-scale regions. More specifically, a g-scale region is essentially a fuzzy connected region where adjacent points satisfy a homogeneity condition $|f(c) - f(d)| < \theta_g$. Here, c and d are arbitrary neighboring points, θ_g is the homogeneity threshold, $f(\cdot)$ is the feature vector at a point, and $|\cdot|$ represents a dissimilarity metric. As shown in Figure 31, the largest g-scale region selected a large white matter region while avoiding abnormal structures since the transition to these areas do not satisfy the homogeneity condition.

Once the largest g-scale region was found, a transformation to a standard scale was made by assigning three marker values to the g-scale region, μ_i, p_1, and p_2, where μ_i denotes the mean intensity value, and p_1 and p_2 represent the upper and lower bound pixel values in the g-scale region. In order to exclude outliers in the g-scale region, a percentage p of the data points with pixel values smaller than the lower bound or larger than the upper bound were trimmed. Assuming that values for a standard scale were obtained, each intensity value x can be transformed to the standard scale through the following equations:

$$
x' = \begin{cases}
s_2 + (x - s_2)\frac{s_2 - s_1}{p_2 - p_1}, & x > s_2 \\
\mu_s + (x - \mu_i)\frac{s_2 - \mu_s}{p_2 - \mu_i}, & \mu_i < x < s_2 \\
\mu_s + (x - \mu_i)\frac{s_1 - \mu_s}{p_1 - \mu_i}, & s_1 < x < \mu_i \\
s_1 + (x - s_1)\frac{s_2 - s_1}{p_2 - p_1}, & x < s_1
\end{cases}
\tag{3}
$$

where μ_s is the standard g-scale mean, s_1 is the standard g-scale lower bound, and s_2 is the standard g-scale upper bound. Those standard values were found by training as described below.

The standard g-scale intensity markers were found in the training phase by averaging marker values over a set of MRI images. In this study, 60 training samples having abnormal regions were selected. It is important to note that the derived sequences have absolute values and thus, the standardization algorithm was only applied to the acquired scans, FLAIR, T1-weighted, post-contrast T1-weighted, T2-weighted, and DTI. In our study, the largest g-scale region was found from the FLAIR sequence and used as a reference for all sequences. This is because the largest g-scale region in the FLAIR modality is more consistent in selecting similar structures than the other sequences. For each training sample, the mean intensity value μ_i^t, lower bound p_1^t,
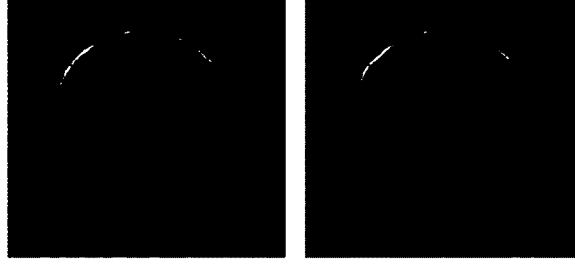
Figure 31: Standardization example. (left) Original image. (right) Contour shows g-scale region. A large white matter region is selected while avoiding the tumor.
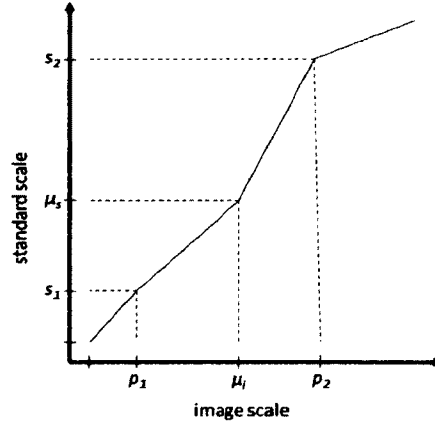
Figure 32: Standardization piece-wise transformation.

and upper bound p_2^t were computed. The average of the mean values μ_i^t were calculated for the entire training set to produce $\mu_s = \frac{1}{T} \sum_{t=1}^{T} \mu_i^t$. Similarly, s_1 and s_2 can be derived. This was repeated for the other acquired sequences to produce standard values for each sequence using the g-scale region found in the FLAIR sequence. The transformation function in Equation (3) was then used to transform each sequence to standard values. Ideally, all images will have the same scale after standardization.

5.3.2 APPLY INCREMENTAL APPROACH

The incremental approach described in Chapter 3 is applied to this data set. For sampling, the LCV method as described in 3.1.1 was used. The set of data that are sampled are considered training data while the remaining data points are considered the testing data. Isomap is applied on the sampled data to produce a manifold. Since

Figure 33: Example set of sampled points. The sampled points are collected using LCV within the selected normal and abnormal regions. The dotted yellow contour denotes the marked normal region while the solid red contour denotes the marked abnormal region. (left) Original Image. (right) Sampled Points using LCV.

the manifold learned is only a subset of the entire data set, the manifold is referred to as the manifold skeleton.

The number of sampled points are varied between 1000, 1500, 2000, and 2500. Also, the number of nearest neighbors used for Isomap's graph construction is varied between 9, 11, and 13. This results in 12 different combinations of parameters. To select the best combination, a validation method is applied to each manifold to produce a metric for comparison as described in the next section. The parameters that produce the best value for the validation metric will be used on the testing data. Thus, only one manifold will be selected while the others are discarded. Note that the validation is performed only on the sampled data points before embedding. The samples that were not selected to become landmarks during the sampling step are embedded into the manifold skeleton. Embedding into the manifold skeleton was implemented using local linear embedding.

5.3.3 VALIDATION OF THE LEARNED MANIFOLD SKELETON:

Successful manifold learning depends on many factors such as parameter choices for the learning algorithm and numerical stability of the algorithm. In this study, the need arises to determine the number of landmarks to be selected and the number of neighbors to be used by the dimensionality reduction algorithm. Two cost functions below are proposed,

$$SF = \frac{\sum\limits_{i,j,i \neq j} \left(d_G\left(i,j\right) - d_Y\left(i,j\right)\right)^2}{\sum\limits_{i,j,i \neq j} d_G^2\left(i,j\right)}$$

and

$$Acc = \frac{\text{No. of correctly classified training data}}{\text{Total no. of training data}}$$

where SF is the stress function and Acc represents the training accuracy after manifold learning. Intuitively, if the nonlinear manifold in high-dimensional space is successfully unfolded, the Euclidean distance between points i and j in the low-dimensional space will be the same as that of the geodesic distance in the high-dimensional space. If a set of labeled training data is provided, which is the case in our study, the Acc value is a good criterion to verify the manifold learning. Otherwise, the stress function can be used in an unsupervised manner requiring no label information.

5.3.4 DIMENSION SELECTION:

After the manifold was learned, the intrinsic dimensionality of the data was found to be around four or five. However, the learned manifold dimensions are usually not equally important for the subsequent classification. In this step, a feature selection method was performed to identify important dimensions. The post-manifold dimensions were ranked by the Fisher score [85], which is a widely used supervised feature selection method and is defined for a two-class problem as,

$$F\left(j\right) \equiv \frac{\left(\bar{z}_{j1} - \bar{z}_{j2}\right)^2}{s_{j1}^2 + s_{j2}^2}$$

where s_{j1} and s_{j2} are the standard deviations of class 1 and class 2 for the jth feature and $\bar{z}_{j1}$ and $\bar{z}_{j2}$ are the means of class 1 and class 2 for the jth feature respectively.

In the experiments, the Fisher scores for features beyond the third highest were found to be two orders of magnitude smaller than the highest score. The lower scores were discarded in the subsequent classification step. After this step, the data set was reduced to three dimensions which can be easily visualized.

5.3.5 CLASSIFIER TRAINING

The last step is to classify the training data. In this section, various classification algorithms are discussed. For the completed work, a GMM was used to classify pixels in the MRI image to normal and abnormal regions. The GMM was chosen for this step since it results in a posterior probability map instead of a binary classification. Other algorithms presented here will be used in this framework's future work.

GMM Classifier

The data set was applied to a Gaussian mixture model (GMM) for classification. A benefit of using GMM is that a posterior probability mapping can be generated. Here, the landmarks chosen in the landmark selection step were used for training the parameters of the GMM using the Expectation Maximization (EM) algorithm [4]:

Step 1: Initialize the means μ_k, covariance σ_k, and mixing coefficients π_k for the k-th component in the GMM.

Step 2: Expectation step. Evaluate the responsibilities using the current parameters

$$\gamma\left(u_{nk}\right) = \frac{\pi_k \mathcal{N}\left(z_n | \mu_k, \sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(z_n | \mu_j, \sigma_j\right)}$$

where K is the total number of components in the GMM. The responsibilities, $\gamma(u_{nk})$, at each point can be viewed as the posterior probability of each Gaussian function.

Step 3: Maximization step. Compute new parameter values using the current responsibilities

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma\left(u_{nk}\right) z_n$$

$$\sigma_k^{new} = \mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma\left(u_{nk}\right) z_n$$

$$\pi_k^{new} = \frac{N_k}{N}$$

where

$$N_k = \sum_{n=1}^{N} \gamma\left(\mu_{nk}\right)$$

and N is the total number of data samples for training. Here, the means, covariance, and mixing coefficients are re-estimated to maximize the responsibilities or posterior probabilities in Step 2.

Step 4: Evaluate the log likelihood

$$\ln p\left(Z|\mu,\sigma,\pi\right) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k\mathcal{N}\left(z_n|\mu_k,\sigma_k\right)\right\} \tag{4}$$

If the values do not converge, then the process is reiterated at step 2. With each iteration of the Expectation and Maximization step, the log likelihood function in Equation (4) will always increase and is guaranteed to converge to a local optimum.

After training, each point in the MRI data set can be classified into the class that has the highest probability $p\left(Z_c|\mu,\Sigma,\pi\right)$. Each pixel will then have a probability that the pixel belongs to the abnormal class and a probability map can be produced for the MRI data set. The final classification can be computed by thresholding the GMM classification probability.

SVM classifier

Support vector machines (SVM) can also be used as a classifier. SVM separates class clusters with high dimensional hyperplanes such that the distance between data points and the hyperplane is maximized [86]. This can be conceptualized using the concept of a margin, which is defined as the smallest distance between any sample and the decision boundary. The SVM will thus find a decision boundary such that the margin from any sample to the decision boundary is maximized [4]. Considering the case of a two-class linearly separable data set, a decision boundary can be made such that all data points will satisfy the constraint

$$t_n\left(w^\top\phi\left(x_n\right)+b\right)\geq 1, \qquad n=1,\ldots,N \tag{5}$$

where N is the number of input vectors, $t_n \in \{-1,1\}$ signifies the target classification, w is a normal vector to the hyperplane, $\phi\left(x\right)$ denotes a fixed feature-space transformation, and b is the bias parameter. In high dimensional space, this is the representation for the decision hyperplane. Maximizing the margin to the hyperplane can be represented by maximizing $\|w\|^{-1}$ or equivalently minimizing $\|w\|^2$. Thus the

optimization problem becomes

$$\arg\min_{w,b} \frac{1}{2} \|w\|^2.$$

In order to expand SVM to classes with overlapping class distributions, which are much more prevalent in real data sets than linearly separable data, a relaxation term must be included in the constraint in Equation 5. A slack variable $\xi_n \geq 0$ is introduced for each data point $x_1 \ldots x_n$ and is defined as $\xi = |t_n - y(x_n)|$ where $y(x)$ is the predicted classification from the SVM. Conceptually, the slack variable will be zero if the point is inside the correct margin boundary and positive otherwise. The constraint is therefore modified to:

$$t_n y(x_n) \geq 1 - \xi_n, \qquad n = 1, \ldots, N.$$

The slack variables now allow the SVM to softly penalize points that are on the wrong side of the decision boundary. The minimization of this model then becomes

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|w\|^2$$

where $C > 0$ is an adjustable parameter controlling the trade-off between the slack variable and the margin.

While SVM is popular for many machine learning applications, a drawback of SVM is that it is a non-probabilistic binary linear classifier. Thus, a posterior probability is not found like the GMM classifier.

5.3.6 REGION SELECTION

After classification, a pruning step is used to reduce noise. Small regions of false positives may occur far away from the known tumor region. Since these isolated regions are unlikely to be tumor regions, small clusters marked positive were considered as noise and discarded. This step was performed on the binary classification of the GMM classifier. A morphological close operation was applied to smooth the boundary of the predicted regions. The number of pixels contained in each disjoint cluster was found and only the largest region was defined as the final predicted tumor region.
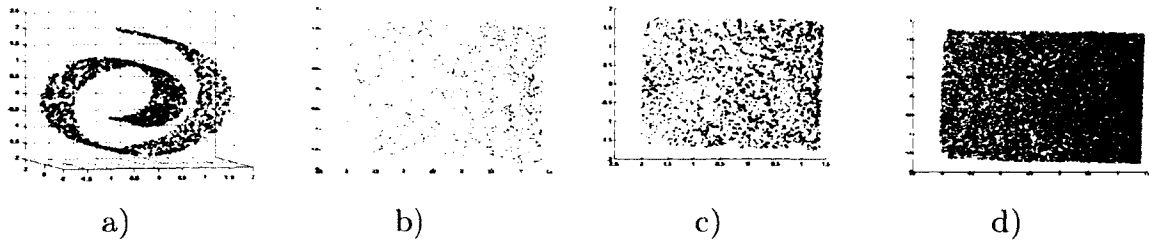
Figure 34: Manifold learning results for swiss roll data sets. A) Original data set. B) Manifold from random sampling. C) Manifold from LCV sampling. D) Manifold result for a very large Swiss Roll data set having 20k data points using LCV sampling.

5.4 PRELIMINARY EXPERIMENTS AND RESULTS

5.4.1 RESULTS FOR A SIMULATED DATA SET - THE SWISS ROLL

Figure 34A-D) show results for the "Swiss roll" data set. Figure 34A) is the original data set having 2000 data points. Figure 34B) shows the learned results based on 900 randomly selected landmarks (SF = 0.4527). Figure 34C) is the result based on 900 landmarks selected based on the LCV concept (SF = 0.4293). Figure 34D) illustrates the result for a very large Swiss Roll data set having 20k data points. A direct manifold learning for this data set is computationally expensive, so we utilized the manifold skeleton learned in Figure 34C) and inserted all the 20k data points into the skeleton based on the LLE algorithm. Results in B) is slightly worse than that in C) as expected.

5.4.2 TESTING METHODS

Along with the proposed method, five other testing methods were implemented for comparison in the previously completed work [87].

Method 1: Proposed The system diagram of the proposed method is shown in Figure 28. First, MRI data sets were standardized using the g-scale standardization method. Second, the LCV sampling strategy was employed to randomly select a set of landmarks inside the defined tumor and normal regions for manifold skeleton learning. Third, the remaining data points were then inserted into the skeleton using the LLE algorithm. Fourth, three dimensions were selected by the Fisher

score based on the selected landmarks and a GMM model was trained using the selected dimensions. Note that only those selected landmarks from LCV sampling were utilized for GMM training. And lastly, the trained GMM model was applied to the whole MRI scan to produce a probability map for abnormal tissue. A crisp classification was also generated based on the probability map by thresholding.

Method 2: PCA The second method is a variation of the proposed method, where the principle component analysis (PCA) method was used instead of Isomap for dimensionality reduction. This will allow for a comparison between a nonlinear and a conventional linear dimensionality reduction method. The sampling and embedding steps were performed the same way as we did in Method 1. While PCA is less computationally intensive and does not require these steps, the sampling and embedding processes will allow us to directly compare the proposed method with PCA.

Method 3: RAW In the RAW method, everything was the same as the proposed method except that we did not perform dimensionality reduction. LCV was used to select landmarks for training the GMM classifier using all ten features of the data set. This will give a comparison between methods employing dimensionality reduction versus raw data.

Method 4: RAW with Fisher score This method is a variation of Method 3. We directly utilized the Fisher score to reduce the dimensionality from ten to three without prior dimensionality reduction methods.

Method 5: PCA without Fisher score Method 5 is the same as Method 2 except that we did not use the Fisher score to select dimensions. Instead, we just kept the first three principle components corresponding to the first three largest eigenvalues. Thus, the effect of the Fisher score on the classification accuracy can be evaluated.

Method 6: Proposed without Fisher score Method 6 is the same as the proposed method except that we just kept the first three coordinates resulting from the Isomap algorithm. Again, this method will provide some insight on how important the Fisher score step is.
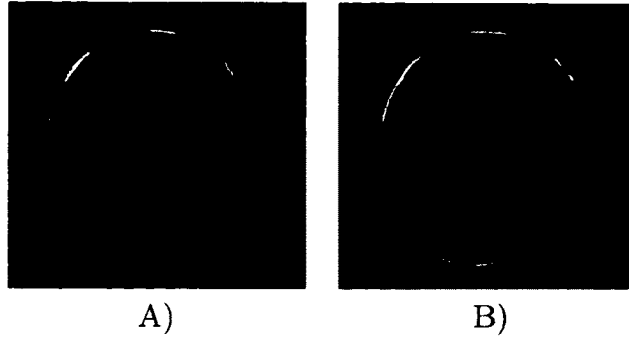
Figure 35: Original FLAIR images for Subject 1 with marked abnormal and normal regions. Red polygon defines abnormal regions and the yellow dotted polygon denotes normal regions. A) Visit 1. B) Visit 2 with the progressed tumor.

5.4.3 PRELIMINARY RESULTS

We applied the six methods described above to four patients' MRI scans. Figures 35, 38, 41 and 44 show each patients' marked abnormal region with a red polygon and also the marked normal region with a yellow dotted polygon overlaid in the FLAIR series. The results are shown in Figures 36, 37, 39, 40, 42, 43, 45, and 46. Column A) in Figures 36, 39, 42, and 45 presents the GMM probability map for abnormal tissue. Column B) shows corresponding binary versions by thresholding the probability maps from column A) with a threshold value of 0.5. The images in column B) also show the marked abnormal regions at visit 1 with a red solid-line polygon along with the marked normal regions with a yellow dotted-line polygon. In column C), the same binary classification images are overlaid with the tumor contours defined at visit 2. The scatter plots in columns D) and E) in Figures 37, 40, 43, and 46 illustrate a three-dimensional representation of the data. In these figures, the red dots denote the abnormal tissue samples, the blue dots are the normal tissue samples and the green dots represent progressed tissue samples, i.e., those tissue samples outside the tumor contours defined at visit 1. These green dots can be interpreted as tumor growth. More specifically, the green dots of column D) show points that are predicted as abnormal but lie outside of the contour of visit 1. Alternatively, the green dots of column E) show points that are marked as abnormal at visit 2 but were not marked as such at visit 1. The difference between column D) and column E) lies in that column D) shows the predicted results while column E) demonstrates the ground truth.

Figure 36: Subject 1 results. Row 1) Proposed. Row 2) PCA. Row 3) RAW. Row 4) Fisher score only. Row 5) PCA w/o Fisher. Row 6) Proposed w/o Fisher.
Column A) GMM probability. Column B) Visit 1 classification. Column C) Visit 2 classification.

D)   E)

Figure 37: Subject 1 results showing 3D feature space of sampled points and progressed tumor points.

Row 1) Proposed. Row 2) PCA. Row 3) RAW. Row 4) Fisher score only. Row 5) PCA w/o Fisher. Row 6) Proposed w/o Fisher.

Col D) Red points show abnormal sampled points. Blue points show normal sampled points. Green points show predicted progressed tumor points. Points predicted as abnormal outside of Visit 1's abnormal contour. Col E) Green points show actual progressed tumor points. Points in set of abnormal points in Visit 2 while not in set of points of Visit 1.

Table 6: Method comparison: Subject 1

	Sensitivity Visit 1	Specificity Visit 1	Sensitivity Visit 2	Precision Visit 2	Average
Method 1: Proposed	0.987	1	0.860	0.566	0.853
Method 2: PCA w/ Fisher	0.978	1	0.769	0.647	0.849
Method 3: Raw data	0.752	1	0.562	0.801	0.779
Method 4: Fisher score only	0.965	1	0.691	0.856	0.878
Method 5: PCA w/o Fisher	0.994	1	0.784	0.722	0.888
Method 6: Proposed w/o Fisher	0.972	1	0.702	0.812	0.871

A)   B)

Figure 38: Original FLAIR images for Subject 2 with marked abnormal and normal regions. Red polygon defines abnormal regions and the yellow dotted polygon denotes normal regions. A) Visit 1. B) Visit 2 with the progressed tumor.
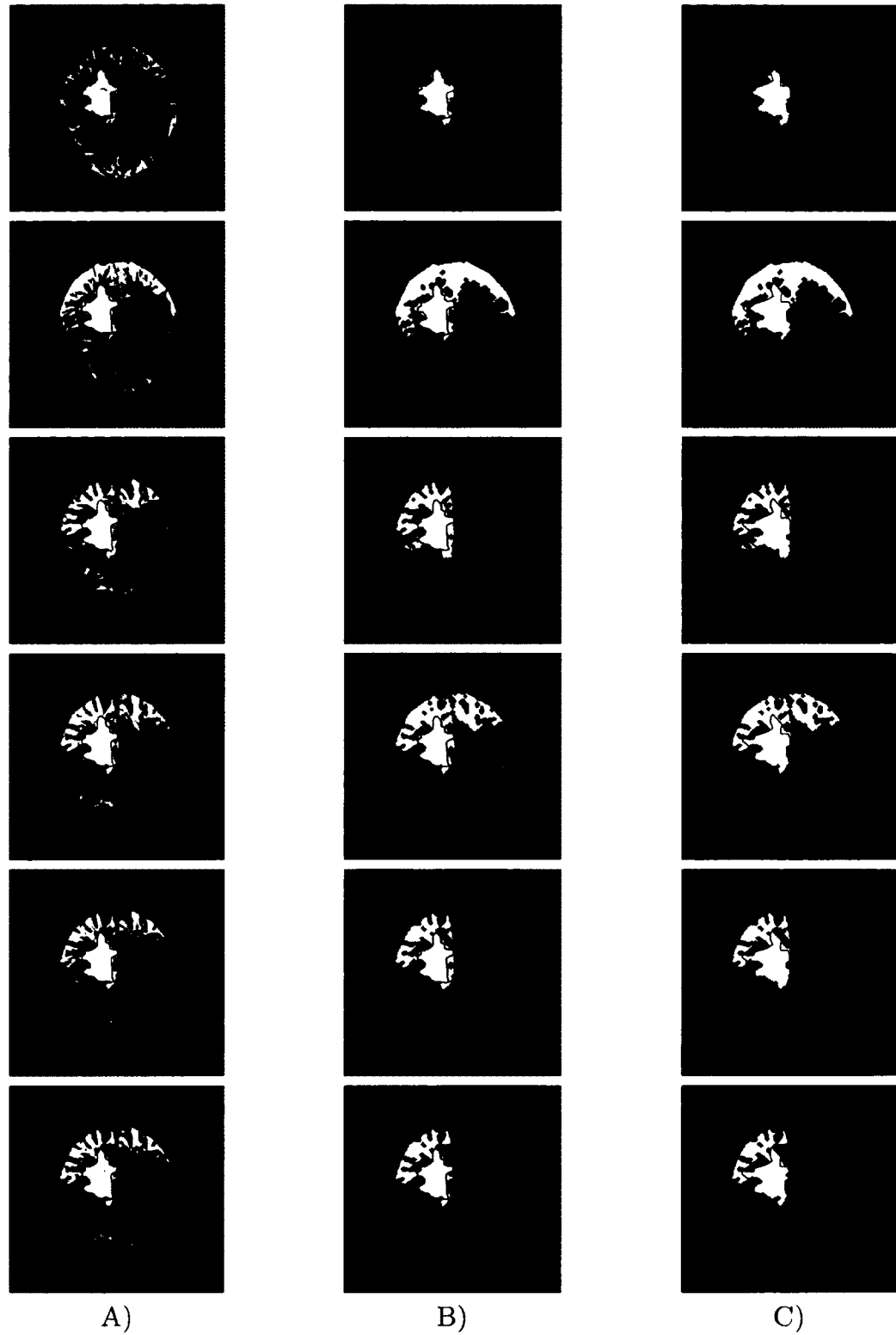
Figure 39: Subject 2 results. Row 1) Proposed. Row 2) PCA. Row 3) RAW. Row 4) Fisher score only. Row 5) PCA w/o Fisher. Row 6) Proposed w/o Fisher. Column A) GMM probability. Column B) Visit 1 classification. Column C) Visit 2 classification.

D)   E)
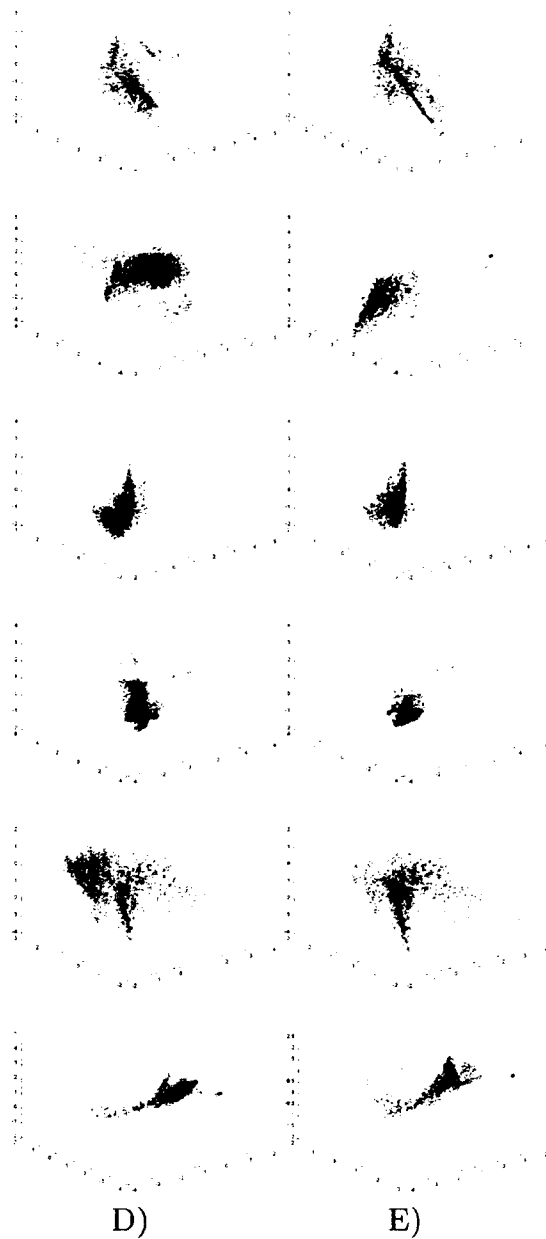
Figure 40: Subject 2 results showing 3D feature space of sampled points and progressed tumor points.

Row 1) Proposed. Row 2) PCA. Row 3) RAW. Row 4) Fisher score only. Row 5) PCA w/o Fisher. Row 6) Proposed w/o Fisher.

Column D) Red points show abnormal sampled points. Blue points show normal sampled points. Green points show predicted progressed tumor points. Points predicted as abnormal outside of Visit 1's abnormal contour. Column E) Green points show actual progressed tumor points. Points in set of abnormal points in Visit 2 while not in set of points of Visit 1.

Table 7: Method comparison: Subject 2

|	Sensitivity Visit 1	Specificity Visit 1	Sensitivity Visit 2	Precision Visit 2	Average
Method 1: Proposed	0.943	1	0.626	0.822	0.839
Method 2: PCA	0.970	1	0.627	0.297	0.723
Method 3: Raw data	0.991	1	0.718	0.564	0.818
Method 4: Fisher score only	0.991	1	0.668	0.376	0.759
Method 5: PCA w/o Fisher	0.989	1	0.676	0.561	0.807
Method 6: Proposed w/o Fisher	0.968	1	0.631	0.556	0.789
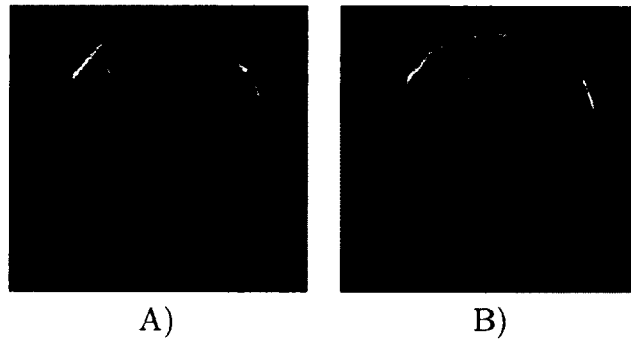
A)   B)

Figure 41: Original FLAIR images for Subject 3 with marked abnormal and normal regions. Red polygon defines abnormal regions and the yellow dotted polygon denotes normal regions. A) Visit 1. B) Visit 2 with the progressed tumor.
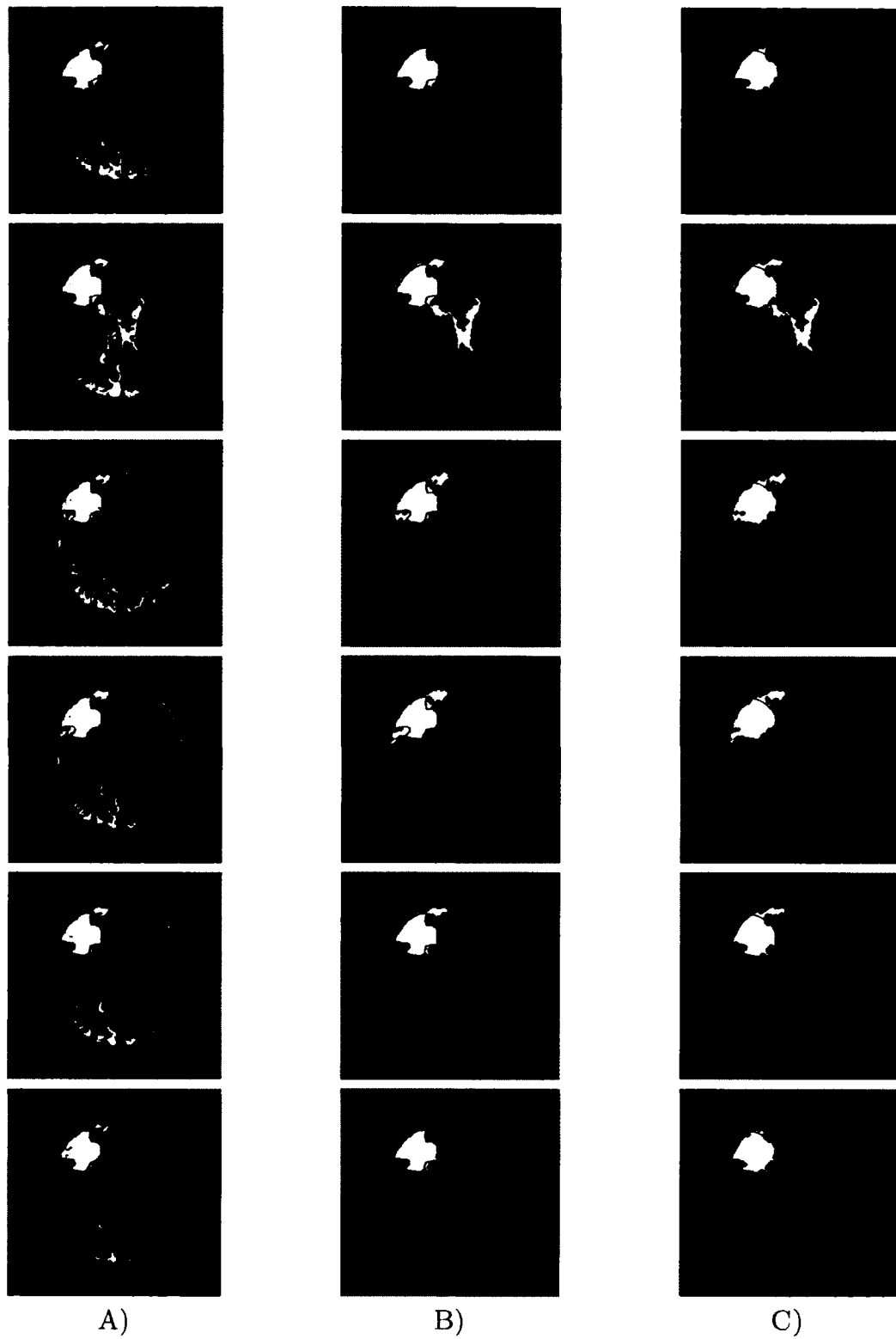
Figure 42: Subject 3 results. Row 1) Proposed. Row 2) PCA. Row 3) RAW. Row 4) Fisher score only. Row 5) PCA w/o Fisher. Row 6) Proposed w/o Fisher. Column A) GMM probability. Column B) Visit 1 classification. Column C) Visit 2 classification.

D)   E)
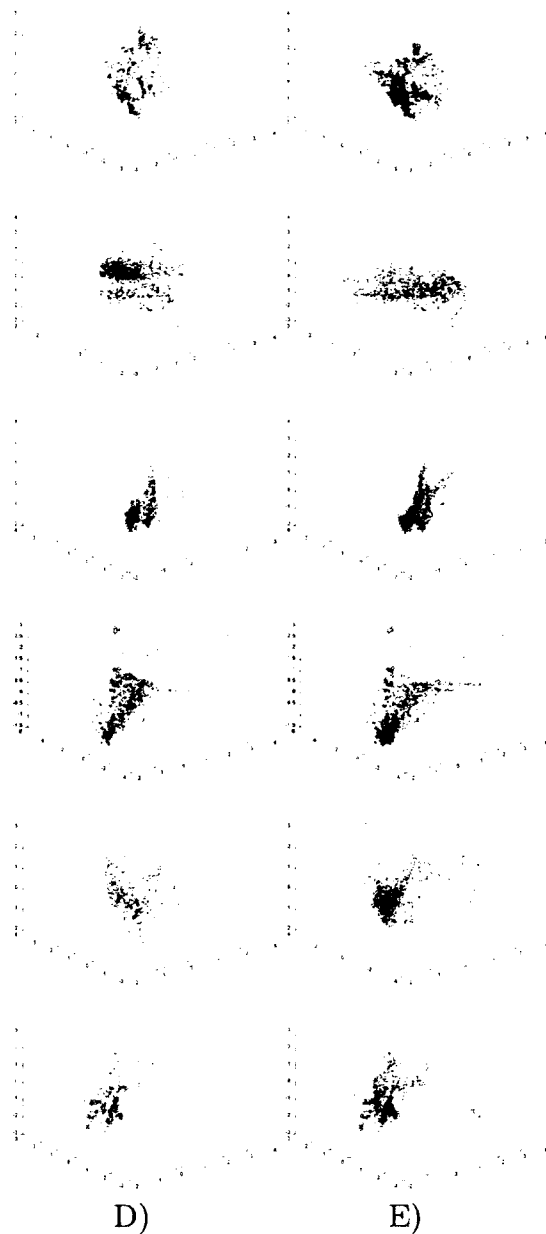
Figure 43: Subject 3 results showing 3D feature space of sampled points and progressed tumor points.

Row 1) Proposed. Row 2) PCA. Row 3) RAW. Row 4) Fisher score only. Row 5) PCA w/o Fisher. Row 6) Proposed w/o Fisher.

Column D) Red points show abnormal sampled points. Blue points show normal sampled points. Green points show predicted progressed tumor points. Points predicted as abnormal outside of Visit 1's abnormal contour. Column E) Green points show actual progressed tumor points. Points in set of abnormal points in Visit 2 while not in set of points of Visit 1.

Table 8: Method comparison: Subject 3

	Sensitivity Visit 1	Specificity Visit 1	Sensitivity Visit 2	Precision Visit 2	Average
Method 1: Proposed	0.921	1	0.591	0.971	0.871
Method 2: PCA	0.950	0.995	0.625	0.590	0.790
Method 3: Raw data	0.944	1	0.656	0.863	0.866
Method 4: Fisher score only	0.921	1	0.637	0.845	0.851
Method 5: PCA w/o Fisher	0.950	1	0.606	0.872	0.857
Method 6: Proposed w/o Fisher	0.919	1	0.589	0.969	0.869

A)　　　　　　B)
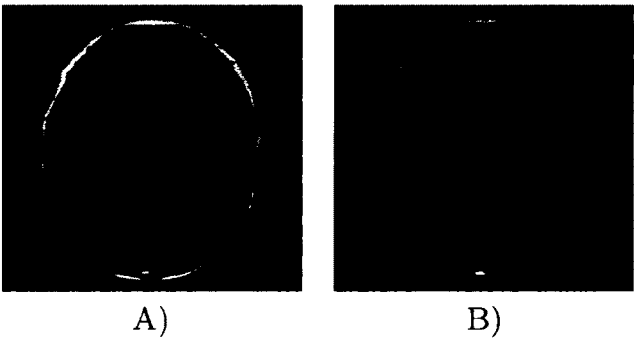
Figure 44: Original FLAIR images for Subject 4 with marked abnormal and normal regions. Red polygon defines abnormal regions and the yellow dotted polygon denotes normal regions. A) Visit 1. B) Visit 2 with the progressed tumor.
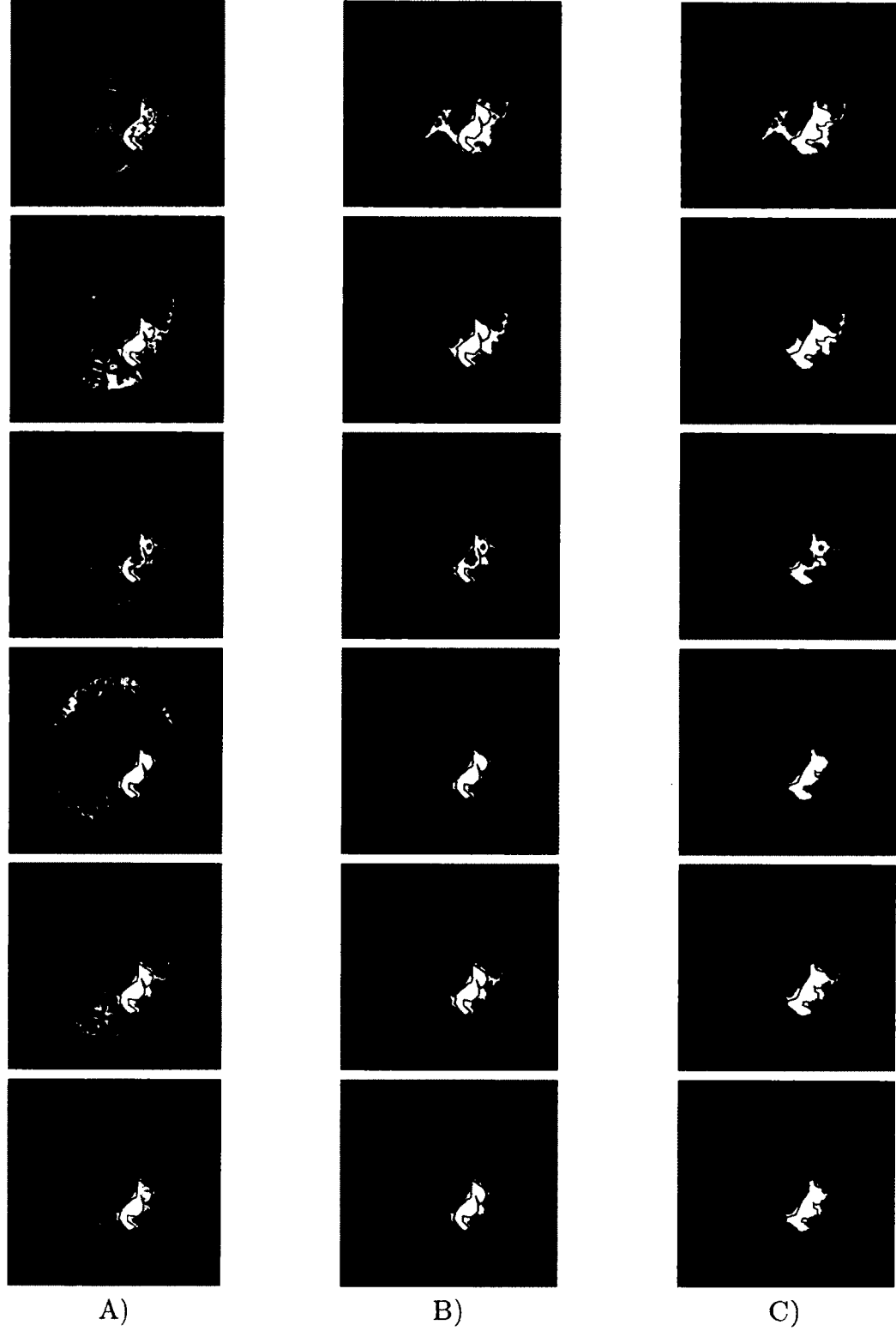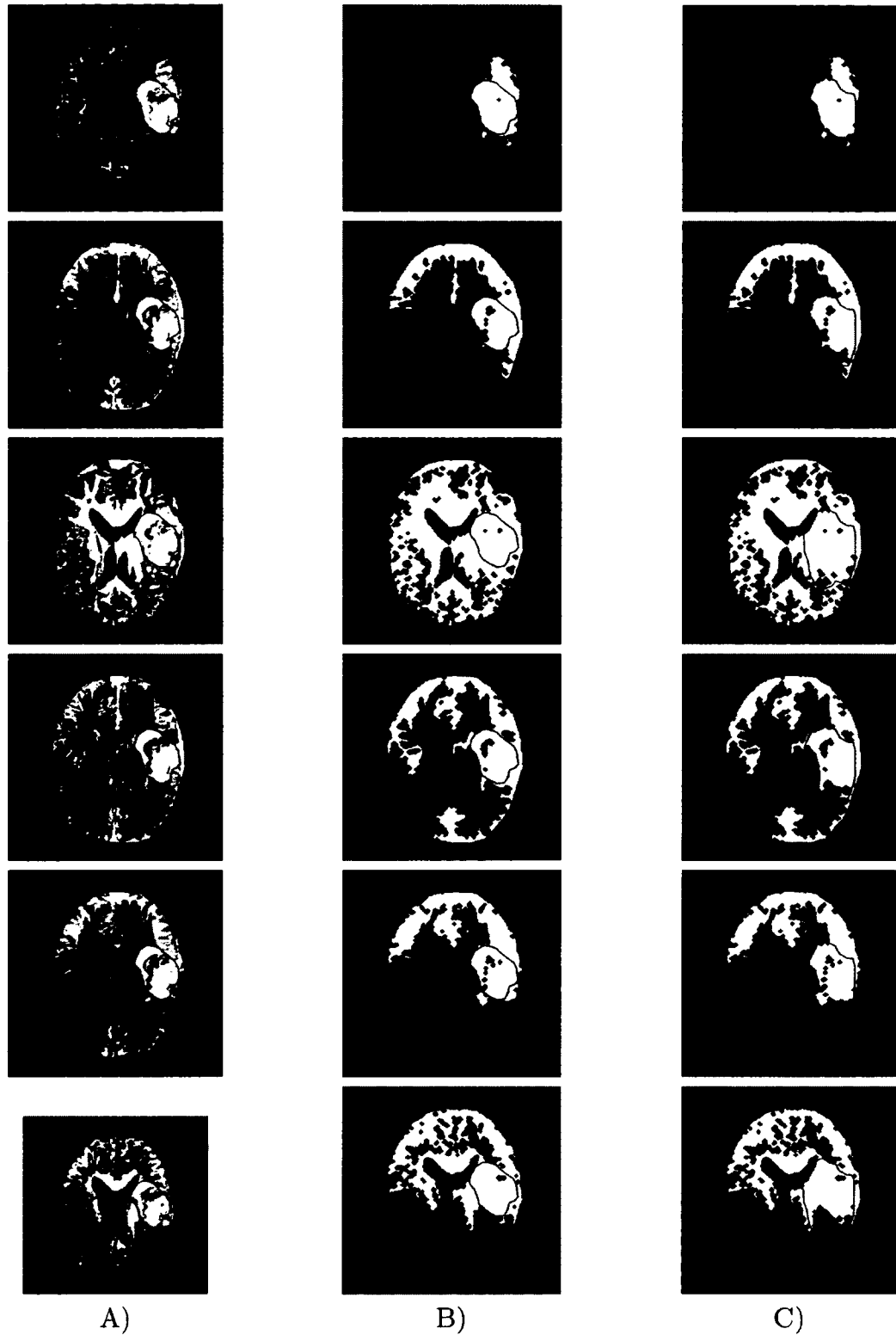
Figure 45: Subject 4 results. Row 1) Proposed. Row 2) PCA. Row 3) RAW. Row 4) Fisher score only. Row 5) PCA w/o Fisher. Row 6) Proposed w/o Fisher.
Column A) GMM probability. Column B) Visit 1 classification. Column C) Visit 2 classification.
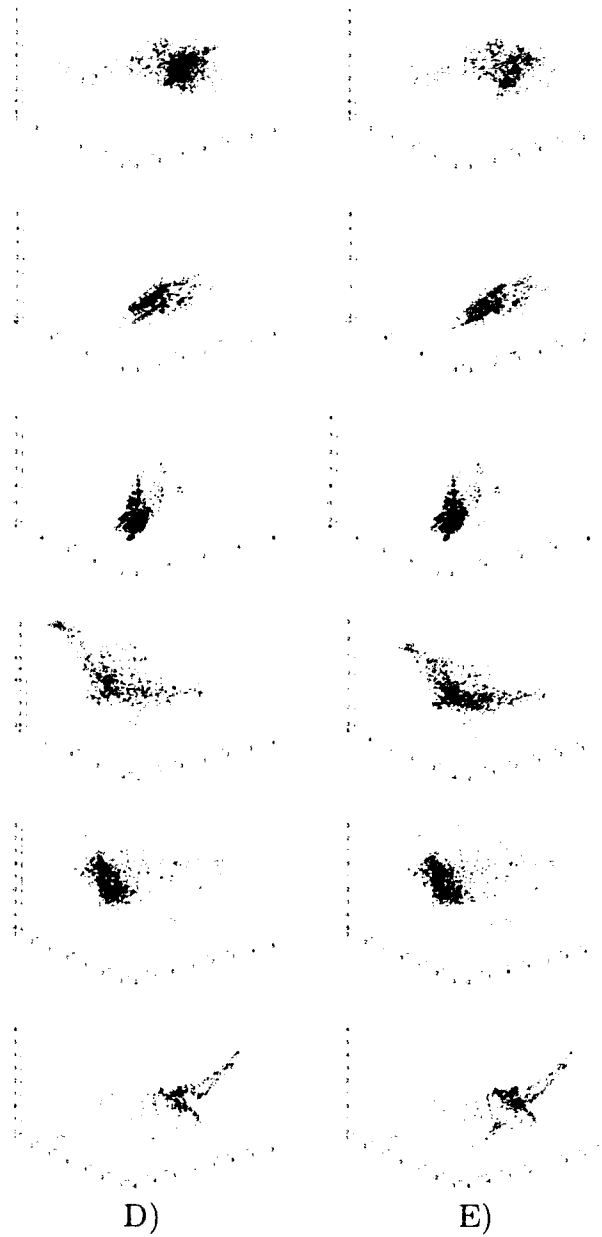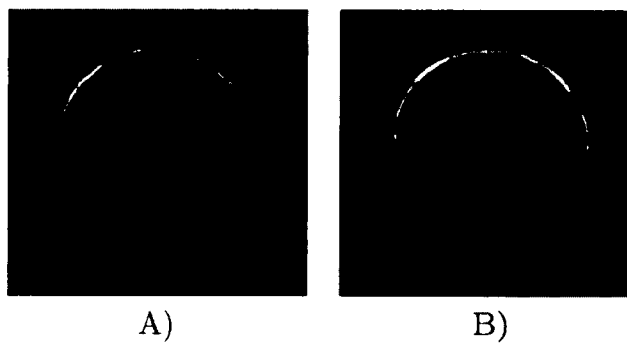
D)   E)
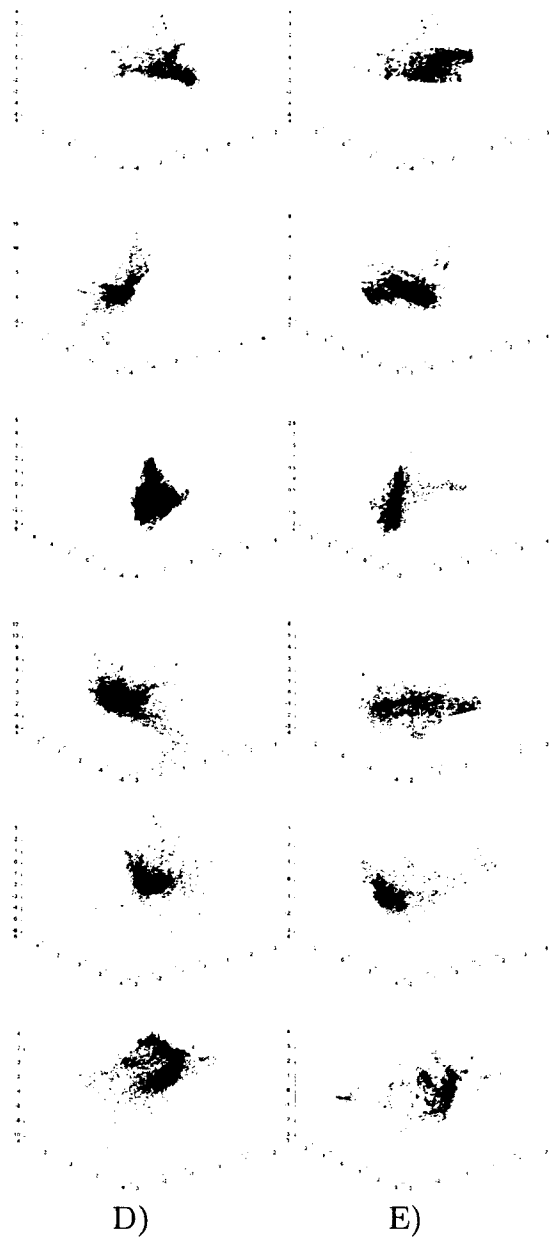
Figure 46: Subject 4 results showing 3D feature space of sampled points and progressed tumor points.

Row 1) Proposed. Row 2) PCA. Row 3) RAW. Row 4) Fisher score only. Row 5) PCA w/o Fisher. Row 6) Proposed w/o Fisher.

Column D) Red points show abnormal sampled points. Blue points show normal sampled points. Green points show predicted progressed tumor points. Points predicted as abnormal outside of Visit 1's abnormal contour. Column E) Green points show actual progressed tumor points. Points in set of abnormal points in Visit 2 while not in set of points of Visit 1.

Table 9: Method comparison: Subject 4

	Sensitivity Visit 1	Specificity Visit 1	Sensitivity Visit 2	Precision Visit 2	Average
Method 1: Proposed	0.952	1	0.609	0.766	0.832
Method 2: PCA	0.883	1	0.575	0.357	0.704
Method 3: Raw data	0.982	0.487	0.885	0.239	0.648
Method 4: Fisher score only	0.899	0.943	0.632	0.312	0.696
Method 5: PCA w/o Fisher	0.887	1	0.584	0.343	0.704
Method 6: Proposed w/o Fisher	0.962	1	0.712	0.310	0.746

Table 10: Method comparison: Average over 4 data sets

	Sensitivity Visit 1	Specificity Visit 1	Sensitivity Visit 2	Precision Visit 2	Average
Method 1: Proposed	0.951	1	0.663	0.781	0.849
Method 2: PCA	0.945	0.999	0.649	0.473	0.766
Method 3: Raw data	0.917	0.872	0.705	0.617	0.778
Method 4: Fisher score only	0.944	0.986	0.657	0.597	0.796
Method 5: PCA w/o Fisher	0.955	1	0.663	0.697	0.814
Method 6: Proposed w/o Fisher	0.955	1	0.659	0.662	0.819

Tables 6, 7, 8, and 9 show quantitative performance metrics calculated for each subject. The sensitivity measures the number of pixels correctly predicted as abnormal divided by the total number of marked abnormal pixels. This measure was calculated for both visit 1 and visit 2. Specificity is the ratio of the correctly predicted normal tissue samples inside the normal contours. The precision is the number of correctly predicted abnormal pixels divided by the total number of predicted abnormal points. The precision will be 1 if every pixel predicted as abnormal is within the marked abnormal region and conversely, the metric will be low for methods that have an over-estimated tumor region. The precision was calculated only at visit 2 because the abnormal region was expected to expand between visit 1 and visit 2. The average metrics across those four subjects are summarized in Table 10.

5.5 PRELIMINARY DISCUSSION

It is noted that the proposed method achieved the best overall performance (0.849) among the six methods compared as shown in Table 10 followed by the second best method, Proposed w/o Fisher score (0.819). Method 5 (PCA w/o Fisher) obtained a similar result (0.814) as that by Method 6 (Proposed w/o Fisher). The other three methods performed much worse. In terms of sensitivity at visit 1, all methods performed similarly except Method 3 (Raw data). This is also the case for the specificity at visit 1 where Method 3 obtained a significantly lower accuracy of 0.872. However, Method 3 achieved the best sensitivity at visit 2 compared to other methods. Finally, the proposed method performed the best in terms of precision at visit 2.

If the system is aimed for tumor growth prediction, the sensitivity at visit 2 seems to be the most important performance metric and Method 3 (Raw data) is the best among the compared methods. However, as shown in Figures 39 and 45, the tumor regions predicted by Method 3, and others, are largely over-estimated, thus making the tumor prediction not useful. On the other hand, the proposed method has well confined prediction regions with a much higher precision, leading to the best overall performance.

The predicted tumor regions are usually not well matched to the tumor contours defined at either visit 1 or visit 2. This is because those tumor contours were defined by referring to the FLAIR scans only. The proposed method took into account information from all available series and analyzed the data in the high-dimensional

space that is beyond the capability of human viewers. The posterior probability map shown in column A) of Figures 36, 39, 42, and 45 might provide extra information to radiologists for brain tumor diagnosis. By incorporating radiologists' domain knowledge, the proposed method might help improve their ability to predict tumor growth.

In addition, the predicted tumor regions usually go beyond tumor contours defined at visit 1, implying that the proposed method does have prediction capability. However, for some cases, i.e. in Figures 39, 42 and 45, some portions of the tumors at visit 2 were missed. Possible reasons for this failure include registration errors between visit 1 and visit 2, efficiency of the manifold learning, human error, etc. In this study, we should pay more attention to regions at visit 1, especially of those outside the tumor contours defined at visit 1. We are currently investigating if significant differences exist in the MRI series between the two regions inside the predicted tumor contours: those inside and those outside the tumor contours defined at visit 1. Results from this work might provide evidence if the predicted tumor contours are better tissue characterizations than those defined on FLAIR only.

It is not clear if the Fisher score component can improve the classification. In the proposed method, the Fisher score improved the overall performance from 0.819 to 0.849 (Method 6 vs. Method 1). Similar results can be observed for Method 3 and Method 4. However, adding the Fisher score component to Method 5 (effectively Method 2) degraded the overall performance from 0.819 to 0.766.

In columns D) and E) of Figures 37, 40, 43, and 46 for the proposed method, we note that normal and tumor manifolds are well separated in the low-dimensional space and the progressed manifold is found to lie roughly between them but closer to the tumor manifold. It is also noted that without the manifold learning method, those dimensions are largely overlapped, i.e., the third row in columns D) and E). Other methods also achieved better low-dimensional representations than Method 3, implying that a manifold learning method is helpful for interpreting the high-dimensional MRI data sets.

5.6 CLINICAL IMPACT

Monitoring and predicting the progression of a tumor is clinically important. One of the most challenging issues in cross-patient tumor prediction is its heterogeneity among patients and with each tumor. Predicting long-term progression sites is and

will remain difficult. In this study, a near-term individual model is constructed using data from one time point while validating using a second time point. Although a universal long-term predictive model that can be applied to all patients would be ideal, a patient specific model based on each patient's prior imaging studies and can predict individual patient's near future progression is still very valuable today. Knowing where a tumor would most likely progress is very useful and can potentially lead to local treatment (e.g. radiation) to prevent (or delay) such from happening. Also, producing a prediction model with just one time point will allow for tumor predictions to be made with limited data such as on the first clinical visit. Therefore, even a near-term prediction can provide some clinical benefits to patients.

5.7 CONCLUSION OF INCREMENTAL APPROACH ON MRI DATA

A large-scale nonlinear manifold learning method was developed for analyzing high-dimensional MRI data sets. Using landmark sampling, the sample size of the MRI data set was reduced so that conventional nonlinear dimensionality reduction techniques can be performed. It was shown that there is a distinct separation between normal and tumor tissues in the low-dimensional space. The points belonging to the tumor progression tend to accumulate between abnormal and tumor tissues, showing a nonlinear transition between the two types of tissues. The proposed algorithm is shown to have more well confined prediction regions than other methods that tended to over-estimate the prediction region. This research may be able to help neurologists in making decisions on treatment of brain tumor patients depending upon the area and rate of tumor progression.

Chapter 6

APPLICATION OF ℓ_1 DIMENSIONALITY REDUCTION

ON MRI DATA SET

6.1 MRI EXPERIMENT WITH ℓ_1 ENHANCEMENTS INTRODUCTION

The second experiment on the MRI data set uses the ℓ_1-based manifold learning algorithm described in Section 3.2 [88]. Recent advances in technology make data acquisition a much cheaper process and data sizes are increasing exponentially. Nowadays, data sets usually contain many samples in a very high-dimensional space, making data "big" in both sample size and data dimension. Big data is difficult to store, transmit, visualize and analyze. Dimensionality reduction thus becomes key in reducing a high-dimensional data sets to a low-dimensional space while preserving the inherent structure of the data set. Manifold learning performs dimensionality reduction by identifying low-dimensional structures (manifolds) embedded in a high-dimensional space. Many algorithms involve an eigenvector or singular value decomposition (SVD) procedure on a similarity matrix of size $n \times n$, where n denotes the number of data samples, making them not scalable to big data. As an alternative, low-rank matrix approximation based incremental manifold learning strategies prove to be effective for obtaining near-optimal solutions to the problems [89, 90, 91]. In those algorithms, sampling methods are typically used to select a subset of data points as landmarks. A manifold skeleton is then learned using the landmarks. Finally, out-of-bag (remaining) points are inserted into the skeleton by various methods such as the Nystrom method, the column-sampling technique or the local embedding scheme [28, 31, 87]. However, high-dimensional data is known to be sparse and highly structured. Current available algorithms do not consider the structured sparse property of data, which may significantly influence the performance of the low-rank methods. We present a novel adaptive neighbor selection approach using the sparseness property of ℓ_1 optimization and aim to create a brain tumor progression model from MRI data.

6.2 METHOD

6.2.1 PROPOSED SYSTEM

In this study, we focus on multi-dimensional MRI scans of brain tumor patients with a progressed tumor over two time points. In our data set, ten MRI image volumes are obtained for each patient. Thus, each pixel location can be represented as a 10-dimensional feature vector. By considering each pixel of the 256×256 MRI scans as a data point, the total number of data points is roughly 65k. The problem with having a high number of data points is that conventional manifold learning approaches require an $n \times n$ eigen-decomposition of a distance matrix. For large values of n, this will become highly computation and memory intensive. An approach to alleviate this issue is to sample points from the large set of features and create a manifold using a smaller subset of features thus resulting in a manifold skeleton. Then the unsampled data points are embedding into the low dimensional space by embedding with local linear embedding (LLE) [16]. Neighborhood selection occurs many times in this approach. Both LLE and many manifold learning approaches require a neighborhood selection step. We introduce an adaptively sparse method of neighbor selection that can be applied directly to current large-scale manifold learning approaches. The system diagram for this method is shown in Figure 47. In the proposed method, a link between abnormal points and the tumor progression region is found from manifold learning of the multi-dimensional MRI scans. A prediction on tumor growth is then made by selecting regions close to abnormal points in manifold space.

6.2.2 SAMPLING

To keep a faithful representation of the original manifold, landmarks should be carefully selected from the original data. Ideally, landmarks should be the smallest subset that can preserve the geometry in the original data. Local curvature variation (LCV) is a sampling method that selects points depending on the curvature level at each point. Intuitively, a manifold's data structure can be preserved effectively by sampling more points from high curvature regions. We assigned an importance value for each of the points by computing the local tangent space variation for it. For each data point in the data set, we found its k-nearest neighbors and performed a local
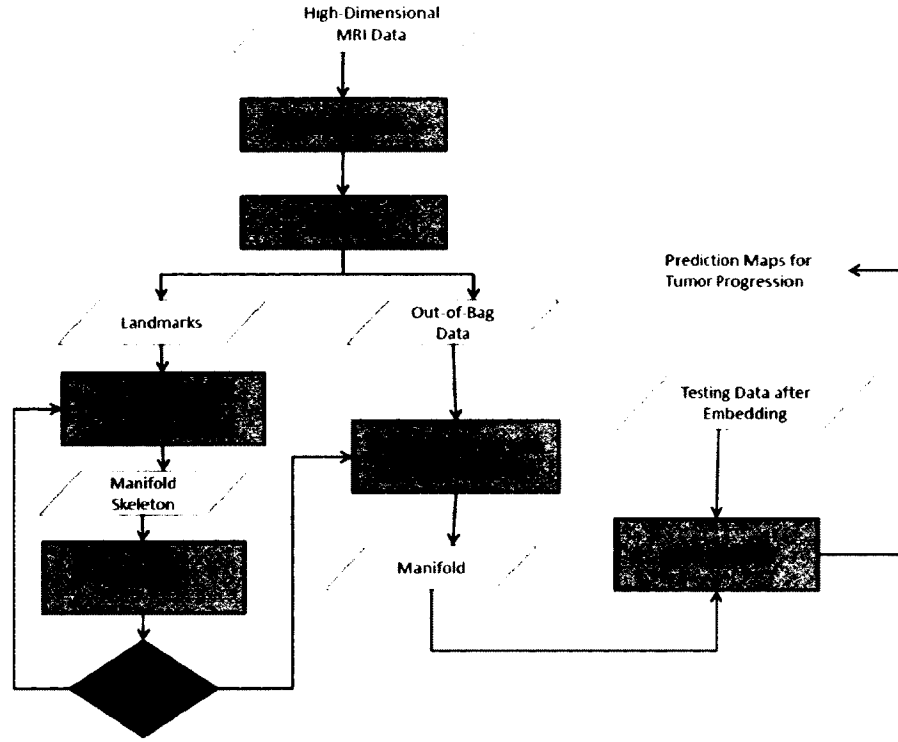
Figure 47: System diagram

principle component analysis on the k-nearest neighbors including itself. We then identified the eigenvector (spans the tangent space) corresponding to the largest eigenvalue. For each data point, it has k such eigenvectors and we computed the mean value of angles between its eigenvector and the eigenvectors of all of its k-nearest neighbors. We then normalized the importance values across all data points such that they sum to one. We then sampled the data set to obtain a set of landmarks based on the importance values.

6.2.3 DIMENSIONALITY REDUCTION

The Isomap method for manifold learning consists of three steps [12]. First, a neighborhood graph is constructed. Second, pairwise distances are calculated through shortest paths from the neighborhood graph. And lastly, Multidimensional Scaling (MDS) converts the pairwise adjacency matrix to a lower dimensional space preserving pairwise geodesic distance. In conventional Isomap, the neighborhood selection is performed with either a ϵ-neighborhood or a k-nearest neighbors approach.

The former method will select all points around an ϵ sized neighborhood to be neighbors. The drawback of this approach is that size of ϵ is difficult to determine. Also, some points may not have neighbors within the specified neighborhood and will thus be unconnected in a neighborhood graph. The k-nearest neighbors will choose the closest k points as neighbors of a data point. An adequate value for k varies within the data set depending on the geometry of the manifold. A point with a k value that is too high may result in "leaking". Likewise, a value that is too low will fail to encompass the underlying manifold.

6.2.4 NEIGHBOR SELECTION WITH ℓ_1-NORM

We propose an adaptive neighbor selection approach using the ℓ_1-norm. A reconstructive cost function is used such that:

$$\min_{\omega \geq 0} \frac{1}{2} \left\| X_{n_i}\omega - x_i \right\|_2^2 + \lambda \left\| \omega \right\|_1$$

where X_{n_i} is the k-nearest neighborhood around x_i and k is the dimension of x_i. ω represents the reconstruction weights. Here, λ controls the sparsity of the solution. The neighbors are selected such that they correspond to the non-zero weights from the equation above. The remaining steps of the Isomap algorithm follow normally.

6.2.5 EMBEDDING WITH ℓ_1-NORM

The goal of this effort is to more effectively insert points that are not used in the dimensionality reduction step into the manifold skeleton. The basic LLE algorithm requires one parameter k for the number of neighbors to use as weights [16]. By using the ℓ_1-norm, the neighbors can be chosen automatically among all n data points. Reconstruction weights are found by solving the following optimization:

$$E\left(W\right) = \sum_i \left(\left| X_i - \sum_{k=1}^{n} W_{ik}X_k \right|^2 + \lambda \sum_{k=1}^{n} |W_{ik}|_1 \right)$$

where X_i is the high-dimensional data point. The non-zero weights of W_{ik} correspond to the neighbors selected to make a reconstruction of the point in manifold space. This adaptive approach is expected to be more robust since the number of neighbors is not fixed for each data point. The ℓ_1-norm in the second term promotes sparsity of the solution. This is because most weights calculated from the ℓ_1-norm

will be zero. The few neighbors with a non-zero weight, W_{ik}, will be chosen as the landmarks to reconstruct the data in the low dimensional manifold. The λ parameter controls the trade-off between reconstruction error and sparsity of the solution.

One drawback to this is that the localization of neighbors in Euclidean space is not guaranteed. This is because each point in the data set is considered to be a potential neighbor. Another consequence in having a large number of potential neighbors is that the number of non-zero weights could potentially be very high.

In a sense, the ℓ_1-norm has the benefit of having an adaptive number of neighbors while the original method guarantees localization and a maximum number of neighbors. In order to incorporate all of these properties, the ℓ_1 optimization function was then modified to the following:

$$E\left(W\right) = \sum_i \left(\left| X_i - \sum_{k=1}^{k'} W_{ik} X_k \right|^2 + \lambda \sum_{k=1}^{k'} |W_{ik}|_1 \right)$$

The neighbors in this formula are restricted to the k' nearest neighbors of each point. In the implementation, a good value for k' is a few values larger than the expected dimensionality of the manifold. To embed the data point X_i into the lower-dimensional manifold, we reconstructed it in the low-dimensional space as Z_i using the weights W_{rk} derived above

$$Z_i = \sum_k W_{ik} Z_k$$

6.2.6 CLASSIFICATION

Next, we classify abnormal regions and normal regions using Gaussian Mixture Models (GMM) with Expectation Maximization optimization. Here, we select the landmarks found in the manifold learning step that fall within the known abnormal and normal regions as the training data. The testing data included all other points within those regions. The advantage of using GMM as the classifier is that a probability map can be created where the probability is the likelihood of a sample to be abnormal. The classification can thus be adjusted by a simple thresholding of the probability map to form a growth prediction. Morphological filtering is applied to the classification to extract the largest contiguous block as the final classification region.
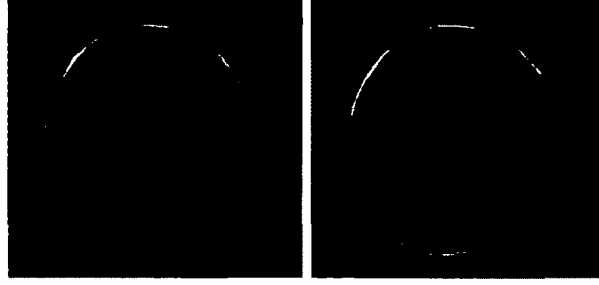
Figure 48: Original ground truths for Subject 1 where the red region is the labeled abnormal region and the yellow is the normal region. (Left) Visit 1. (Right) Visit 2 showing a progressed abnormal region

6.3 EXPERIMENTAL SETUP

The MRI data of brain tumor patients were collected using various MRI scans including FLAIR, T1-weighted, post-contrast T1-weighted, T2-weighted, and DTI. Five scalar volumes were also computed from the DTI volume including apparent diffusion coefficient (ADC), fractional anisotropy (FA), max-, min-, and middle- eigenvalues, yielding a total of ten image volumes for each visit of every patient. Each patient went through a series of scans with an interval of one or two months, and a rigid registration was utilized to align all volumes to the DTI volume of the first visit using the vtkCISG toolkit [81]. After registration, each pixel location can be represented by a ten-dimensional data point corresponding to the ten MRI volumes. Two visits were selected in this study and denoted as "Visit 1" and "Visit 2" where Visit 2 showed an expanded tumor region. Hyper-intensity regions were defined on the FLAIR scans as abnormal regions. A similarly sized region far away from the abnormal regions was also defined as a highly confident normal region for training purposes. Figure 48 shows example MRI slices overlaid on the defined tumor and normal regions. In [87], Tran et al. showed progressed tumor regions lie close to abnormal regions in manifold space. We aim to predict the progression of the tumor in Visit 2 from the manifold learned from Visit 1.

6.4 RESULTS

Figure 49 shows the results of four subjects. Column A denotes the output of the GMM classifier. Column B shows the final classification region after thresholding and filtering the GMM probability map. Here, the solid red polygon are the

Table 11: Results for each subject

	Sensitivity Visit 1	Specificity Visit 1	Sensitivity Visit 2	Precision Visit 2	Average
Subject 1	0.985	1	0.704	0.832	0.880
Subject 2	0.972	1	0.587	0.931	0.872
Subject 3	0.981	1	0.725	0.864	0.893
Subject 4	0.946	1	0.623	0.900	0.867

marked abnormal regions while the dotted yellow polygon denotes the marked normal regions from Visit 1. Column C show the classification region on the progressed hyperintensive region of Visit 2.

Table 11 show quantitative performance metrics calculated for each subject. The sensitivity measures the ratio between the number of pixels correctly predicted as abnormal versus the total number of marked abnormal pixels. This measure was calculated for both Visit 1 and Visit 2. Specificity is the ratio of the correctly predicted normal tissue samples inside the normal contours. The precision is the number of correctly predicted abnormal pixels divided by the total number of predicted abnormal points. The precision will be 1 if every pixel predicted as abnormal is within the marked abnormal region and conversely, the metric will be low for methods that have an over-estimated tumor region. The precision was calculated only at Visit 2 because the abnormal region was expected to expand between Visit 1 and Visit 2. The average metrics across those four subjects are summarized in Table 12 and compared to three other methods. The results for Raw are found by directly applying the GMM classifier in the high dimensional space. For PCA, the dimensionality reduction is performed using principal component analysis. Lastly, [87] follows the same procedure as the proposed method while using ℓ_2-norm optimization.

From Table 12, the proposed method outperforms the other methods in terms of average sensitivity, specificity, and precision. This suggests that the ℓ_1 neighborhood selection creates a more robust manifold. While the proposed approach does not have the best Visit 2 sensitivity, this may be attributed to the other methods over-predicting abnormal regions. This results in the other methods having a low precision.

A)   B)   C)
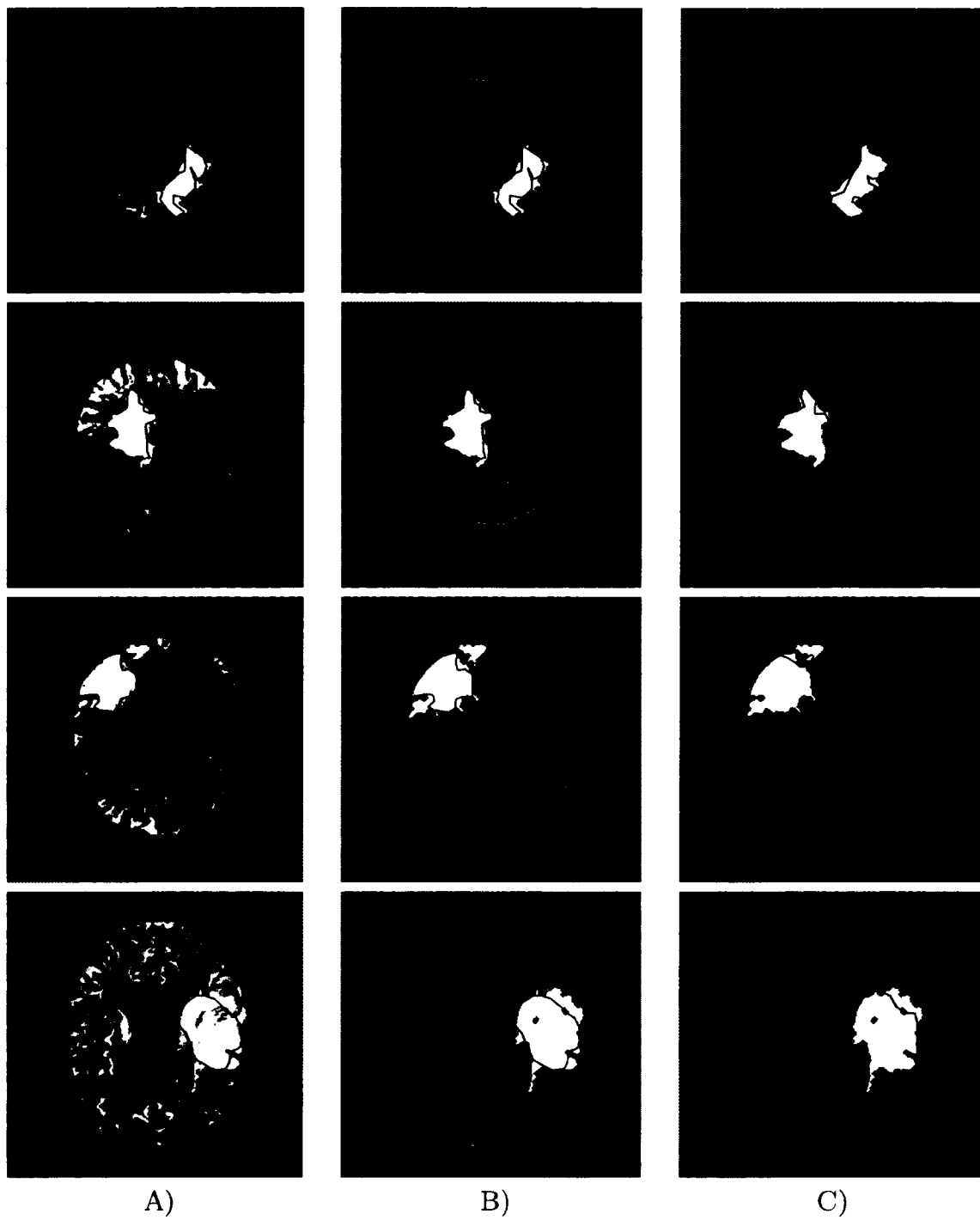
Figure 49: Results for four subjects. Column A shows the output from the GMM. Column B shows the final classification after thresholding and filtering. Column C shows the classification on the second time point

Table 12: Average results over all subjects compared to other methods

	Sensitivity Visit 1	Specificity Visit 1	Sensitivity Visit 2	Precision Visit 2	Average
Proposed	**0.971**	1	0.659	**0.882**	**0.878**
[87]	0.951	1	0.663	0.781	0.849
PCA	0.945	0.999	0.649	0.473	0.766
Raw data	0.917	0.872	**0.705**	0.617	0.778

6.5 CONCLUSION OF ℓ_1-ENHANCED APPROACH ON MRI DATA

We show that a more robust manifold can be achieved using an adaptive neighborhood selection algorithm for large scale manifold learning. The proposed method improves the average classification accuracy of four MRI brain tumor data sets. While we have applied the approach to a specific data set, the general procedure may be easily applied to other nonlinear manifold learning data sets.

6.6 FUTURE WORK

For the remaining work for the MRI data set, LCV and min-max sampling will be applied to the data set. The work completed has been performed using just the random sampling approach. It is expected that LCV sampling will be very beneficial to the nonlinear dimensionality reduction methods especially LTSA since the sampled points using local curvature variation will allow for accurate computation of the local tangent space in LTSA. Also, the LLE embedding algorithm using the ℓ_1 norm will be implemented to adaptively chose the neighbors used in the embedding step. This is the first research effort using dimensionality reduction of a high dimensional MRI data set for brain tumor progression prediction. Also, the future work includes testing the proposed methods on more patient data sets.

Chapter 7

COMPUTATIONAL COMPLEXITY OF THE PROPOSED METHOD VERSUS RANDOMIZED SVD

One of main focuses of this work has been to avoid an eigen-decomposition of a large similarity matrix. In the proposed algorithm, sampling is used to reduce the size of a large data set and thus allowing for an exact decomposition of the smaller data set. This can be seen as an approximation of the spectral decomposition of the original large data set. Another approach is to approximate the decomposition of the large similarity matrix directly using randomized singular value decomposition (SVD). In this chapter, an experiment is conducted to compare the two approaches.

7.1 EXACT EIGEN-DECOMPOSITION

Eigen-decomposition or spectral decomposition refers to the process of breaking down a matrix into a standard form represented by vectors and weights, called respectively, eigenvectors and eigenvalues. Any $m \times n$ matrix A can be expressed as:

$$A = \sum_{t=1}^{r} \sigma_t u^{(t)} v^{(t)T}$$

such that r is the rank of A, $u^{(t)} \in \mathbb{R}^m$, $v^{(t)} \in \mathbb{R}^n$. Here, $u^{(t)}$ and $v^{(t)}$ are the left and right eigenvectors of A respectively and σ_t are the corresponding eigenvalues. Generally, the complexity of directly computing the spectral decomposition of an $m \times n$ matrix is $O\left(mn^2 + m^2n\right)$; or $O\left(n^3\right)$ for a $n \times n$ square matrix.

7.2 RANDOMIZED SVD

While the eigen-decomposition gets exponentially harder to calculate as n or m increases, it is not impossible to approximate. Singular value decomposition (SVD) can be approximated using row sampling approaches such as randomized SVD.

While the exact calculation of the SVD cannot be calculated easily for a large data set, an approximation can be calculated using the method proposed by [92].

Instead of performing an SVD on the entire matrix, a randomized subset of rows or columns are picked to form a smaller matrix. Hence, the approach is referred to as randomized SVD or fast SVD. Theoretically, the SVD of the smaller matrix will give a good approximation to the SVD of the initial large matrix using only a fraction of the computations.

7.2.1 METHOD: RANDOMIZED SVD

The randomized SVD algorithm approximates the top k right eigenvalues and eigenvectors given an $m \times n$ matrix A. The algorithm samples s rows from A to form an $s \times n$ matrix. The probability to choose any of the m rows is $p_1, \ldots, p_m$ such that $\sum_{i=1}^{m} p_i = 1$.

Randomized SVD Algorithm

Input: $m \times n$ matrix A, integers $s \leq m$, $k \leq s$.

Output: $m \times k$ matrix H, $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$.

1. **Row sampling:**

   for $t = 1$ to s

   - Pick an integer from $\{1 \ldots m\}$, where Prob (pick i) $= p_i$

   - Include $A(i)/\sqrt{sp_i}$ as a row of S.

2. **Decomposition:**

   Compute $S \cdot S^T$ and its singular value decomposition such that

   $$SS^T = \sum_{t=1}^{s} \lambda_t^2 w^{(t)} w^{(t)T}$$

3. **Approximate eigenvectors of A:**

   Compute $h^{(t)} = S^T w^{(t)} / |S^T w^{(t)}|$, $t = 1 \ldots k$. Return H, a matrix whose columns are the $h^{(t)}$, and the estimated eigenvalues $\lambda_1, \ldots, \lambda_k$

In step 2, SS^T is symmetric and thus the left and right eigenvectors are each other's transpose, $w^{(t)}$ and $w^{(t)T}$. In practice, the sampled matrix S is a much smaller matrix than the original matrix A. The original matrix A is no longer used after step 1 and can be discarded from memory during implementation. The amount

of ram required to store this matrix $S \cdot S^T$ is only $O\left(sn\right)$ rather than $O\left(mn\right)$ for A. Step 2 can be performed in $O\left(n\right)$ time as proved in [30].

The major advantage of this algorithm is its improvement in speed. The computations required for randomized SVD include the sampling step which can be performed in $O\left(s\log m\right)$ computations and the normalization of the rows which requires $O\left(sn\right)$ operations in step 1. For step 2, the computation of $S \cdot S^T$ is a straightforward problem of $O\left(s^2 n\right)$ and the computation of the full SVD of this sampled matrix is $O\left(s^3\right)$. Including the matrix operations to compute the approximated decomposition H in step 3, the overall computational time of randomized SVD is $O\left(s\log m + s^2 n + s^3 + nsk\right)$.

Since s is a constant that is generally the size of k and in many cases $s \ll n$, the computations required are theoretically reduced from $O\left(n^3\right)$ for a symmetric matrix size $n \times n$ to only $O\left(n\right)$. Experimental results have shown that a good approximation can be achieved with less samples than theoretically needed [30].

The drawback of this method is that it is not an exact calculation but is rather an approximation.

7.3 RANDOMIZED SVD VERSUS PROPOSED SYSTEM

An experiment was conducted to test the effectiveness of using the randomized SVD method and the proposed method for nonlinear dimensionality reduction. In this experiment, a 5000-point swiss roll is created with two classes in a checkerboard pattern with 5% of Gaussian noise. Figure 50 shows both the 3 dimensional swiss roll along with the 2 dimensional manifold ground truths. The classification accuracy was calculated using a simple k-nearest neighbor classifier with $k = 1$. The randomized SVD experiment follows the procedure of Isomap except for the eigen-decomposition, which is performed using randomized SVD. The number of sampled rows is varied in this approach. The proposed method follows the procedure highlighted in Chapter 3 with a varying number of sampled points. As a baseline, an exact singular value decomposition is performed using all sample points. It is expected that the exact calculation will serve as a maximum for the classification accuracy since both tested methods are approximations. The computer used in this experiment has an Intel Xeon X5460 CPU at 3.16 GHz with 8GB of RAM.

7.4 RESULTS VERSUS RANDOMIZED SVD

Accuracy and time plots of the randomized SVD approach and the proposed method are shown in Figure 51. The top plot is the 1-nearest neighbor classification accuracy. The blue line represents the baseline method using an exact SVD calculation. The bottom plot shows the value of the Matlab routine cputime. This can be interpreted as the number of seconds required to perform each approach. As expected, the baseline accuracy is higher than both methods. For the proposed approach, the classification accuracy rises sharply to around 0.90 before leveling off after 1500 landmarks. The randomized SVD approach has a high accuracy even for a low number of sampled rows. An above 90 percent accuracy is achieved with 10 rows. This may be due to the swiss roll having a low intrinsic dimensionality of 2. However, the classification accuracy is erratic and lower accuracies were seen when a larger number of rows were sampled. On the other hand, the proposed approach is more stable with the classification accuracy rising as more samples are introduced.

In terms of computational time, the randomized SVD approach surprisingly has a slightly longer computational time than the exact approach. This can be attributed to the way the Isomap method is calculated. From Section 2.1.2, Isomap calculates a pairwise distance matrix that is used to calculate the low dimensional space. The randomized SVD approach is applied directly to the pairwise distance matrix. The majority of time in the overall algorithm is spent calculating the pairwise distance matrix. For future work, an extension to the direct application of randomized SVD may be to only calculate rows in the pairwise distance matrix that correspond to sampled rows from randomized SVD.

For the proposed approach, the computational time grows exponentially. A classification accuracy of over 90% is achieved for 1500 samples. The computational time at this number of landmarks is one order of magnitude lower than for exact SVD and randomized SVD. It is important to note that the number of total points in this experiment was set to 5000 such that an exact SVD calculation and comparison could be made. The proposed approach could be scaled to a larger data set where the other methods would be impractical due to the computational time. In summary, while randomized SVD may provide higher accuracies at a lower number of samples, the proposed approach is more stable and has a lower computational time than using randomized SVD for dimensionality reduction.
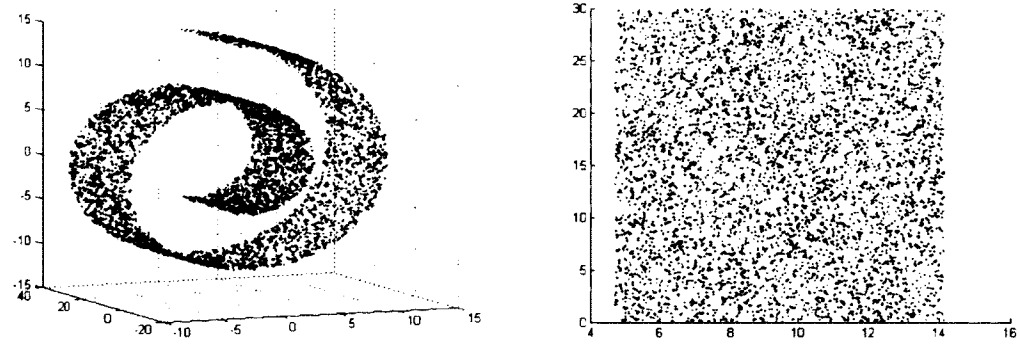
Figure 50: Left, 5000 sample swiss roll data set. Right, 2D unfolded manifold
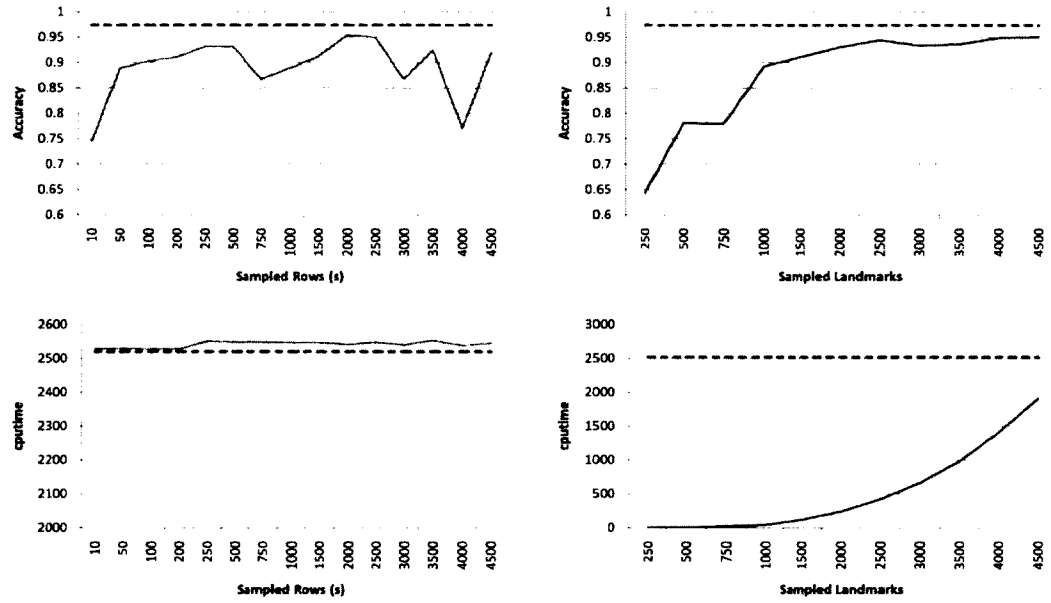
Figure 51: Randomized SVD vs Proposed method for swiss roll data set. Blue dotted line represents Exact SVD over all points. Left, Randomized SVD. Right, Proposed method.

Chapter 8

CONCLUSIONS

In this dissertation, we developed a new nonlinear dimensionality reduction algorithm to find manifolds embedded in large-scale high-dimensional data sets. A framework for large-scale dimensionality reduction was introduced as the Incremental Landmark approach. Under this framework, the manifold for a large data set that cannot be processed using a conventional computer may be approximated. The steps of the framework are to first sample the large data set to a size that is manageable for a conventional computer. Next, the manifold for the sampled set is computed to form the manifold skeleton. Finally, the unsampled points are embedded into the manifold skeleton using LLE. This framework is applied to an MRI tumor identification and progression prediction data set in Chapter 5. A novel sampling method based upon local curvature variation was also introduced and applied to the MRI data set. The results from this experiment showed that the Incremental Landmark approach preformed better than linear methods and also better than using the raw data set for classification.

The main contribution of this dissertation was the introduction of a novel ℓ_1-based neighbor selection algorithm. Two modules form the Incremental Landmark approach are modified to incorporate the adaptively neighbor selection algorithm, the graph construction of Isomap and the LLE embedding step. Each of these methods were tested against the standard methods from the Incremental Landmark approach using benchmark data sets from UCI's data set repository. From these experiments, the adaptive ℓ_1 neighborhood selection showed improvement over the standard methods. For the case of the graph construction module, the adaptive neighbor selection may alleviate issues of leaking from standard Isomap. The two modules were combined to form the final large scale manifold learning algorithm. This final proposed approach was tested versus manifold learning algorithms from the literature using three image data sets; Yale face data set, UMIST face data set, and COIL-20 object recognition data set. The proposed approach showed improvement over the linear methods and sparse projection approaches. Compared with DONPP, the proposed

approach achieved comparable results with the UMIST and COIL-20 data set while achieving a better average classification accuracy with the Yale data set.

Finally, the proposed ℓ_1 based proposed method is applied on the large-scale MRI brain tumor data set. The proposed approach showed an improvement over the Incremental Landmark approach.

References

[1] B. Yang and S. Chen, "Sample-dependent graph construction with application to dimensionality reduction," *Neurocomputing*, vol. 74, no. 1-3, pp. 301 – 314, 2010.

[2] Committee on the Analysis of Massive Data, Committee on Applied and Theoretical Statistics, Board on Mathematical Sciences and Their Applications, Division on Engineering and Physical Sciences, National Research Council, *Frontiers in Massive Data Analysis*. The National Academies Press, 2013.

[3] L. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," 2008.

[4] C. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer, 2006.

[5] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.

[6] B. Scholkopf, A. Smola, E. Smola, and K.-R. Mller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[7] J. A. K. Suykens, "Data visualization and dimensionality reduction using kernel maps with a reference point," *Neural Networks, IEEE Transactions on*, vol. 19, no. 9, pp. 1501–1517, Sept 2008.

[8] C. Chateld and A. Collins, *Introduction to Multivariate Analysis*. Chapman and Hall, 1980.

[9] B. Moghaddam, "Principal manifolds and probabilistic subspaces for visual recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 6, pp. 780–788, Jun 2002.

[10] A. Kocsor and L. Toth, "Kernel-based feature extraction with a speech technology application," *Signal Processing, IEEE Transactions on*, vol. 52, no. 8, pp. 2250–2263, Aug 2004.

[11] L. Zhang, Z. Weida, Y. Lin, and L. Jiao, "Support vector novelty detection with dot product kernels for non-spherical data," in *Information and Automation, 2008. ICIA 2008. International Conference on*, June 2008, pp. 41–46.

[12] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[13] M. Niskanen and O. Silven, "Comparison of dimensionality reduction methods for wood surface inspection," in *Proceedings of the 6th International Conference on Quality Control by Artificial Vision*, Gatlinburg, TN, USA, 2003, pp. 178–188.

[14] I. S. Lim, P. de Heras Ciechomski, S. Sarni, and D. Thalmann, "Planar arrangement of high-dimensional biomedical data sets by isomap coordinates," in *Computer-Based Medical Systems, 2003. Proceedings. 16th IEEE Symposium*, June 2003, pp. 50 – 55.

[15] B. Raytchev, I. Yoda, and K. Sakaue, "Head pose estimation by nonlinear manifold learning," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4, Aug. 2004, pp. 462 – 466 Vol.4.

[16] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[17] T.-M. Chan, J. Zhang, J. Pu, and H. Huang, "Neighbor embedding based super-resolution algorithm through edge detection and feature selection," *Pattern Recognition Letters*, vol. 30, no. 5, pp. 494 – 502, 2009.

[18] R. Duraiswami and V. Raykar, "The manifolds of spatial hearing," in *Acoustics Speech and Signal Processing (ICASSP), 2005 IEEE International Conference on*, vol. 3, 2005.

[19] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, Jan. 2005.

[20] L. Teng, H. Li, X. Fu, W. Chen, and I. fan Shen, "Dimension reduction of microarray data based on local tangent space alignment," in *Cognitive Informatics, 2005. (ICCI 2005). Fourth IEEE Conference on*, Aug 2005, pp. 154–159.

[21] L. Ma, M. Crawford, and J. Tian, "Anomaly detection for hyperspectral images using local tangent space alignment," in *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, July 2010, pp. 824–827.

[22] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5 – 30, 2006, special Issue: Diffusion Maps and Wavelets.

[23] M. Szummer and T. Jaakkola, "Partially labeled classification with markov random walks," in *Advances in Neural Information Processing Systems*. MIT Press, 2002, pp. 945–952.

[24] S. Gepshtein and Y. Keller, "Image completion by diffusion maps and spectral relaxation," *Image Processing, IEEE Transactions on*, vol. 22, no. 8, pp. 2983–2994, Aug 2013.

[25] G. Mishne and I. Cohen, "Multiscale anomaly detection using diffusion maps," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, no. 1, pp. 111–123, Feb 2013.

[26] E. Kokiopoulou and Y. Saad, "Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2143–2156, 2007.

[27] T. Zhang, K. Huang, X. Li, J. Yang, and D. Tao, "Discriminative orthogonal neighborhood-preserving projections for classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 1, pp. 253–263, 2010.

[28] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang, "Matrix approximation and projective clustering via volume sampling," in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, ser. SODA '06. New York, NY, USA: ACM, 2006, pp. 1117–1126.

[29] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix," *SIAM J. Comput.*, vol. 36, no. 1, pp. 158–183, Jul. 2006.

[30] P. Drineas, E. Drinea, and P. S. Huggins, "An experimental evaluation of a monte-carlo algorithm for singular value decomposition," in *Proceedings of the 8th Panhellenic conference on Informatics*, ser. PCI'01. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 279–296.

[31] A. Talwalkar, S. Kumar, and H. Rowley, "Large-scale manifold learning," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.

[32] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 2, pp. 214–225, Feb 2004.

[33] J. C. Platt, "Fastmap, metricmap, and landmark mds are all nystrom algorithms," in *In Proceedings of 10th International Workshop on Artificial Intelligence and Statistics*, 2005, pp. 261–268.

[34] C. Williams and M. Seeger, "Using the nystrm method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 682–688.

[35] A. Talwalkar, S. Kumar, M. Mohri, and H. Rowley, "Large-scale svd and manifold learning," *Journal of Machine Learning Research*, vol. 14, pp. 3129–3152, 2013.

[36] V. de Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," Stanford University, Tech. Rep., 2004.

[37] J. T. Wang, X. Wang, D. Shasha, and K. Zhang, "Metricmap: An embedding technique for processing distance-based queries in metric spaces," *Trans. Sys. Man Cyber. Part B*, vol. 35, no. 5, pp. 973–987, Oct. 2005.

[38] C. Faloutsos and K.-I. Lin, "Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," *SIGMOD Rec.*, vol. 24, no. 2, pp. 163–174, May 1995.

[39] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, Feb 2009.

[40] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, Feb 2006.

[41] R. G. Baraniuk, "More is less: Signal processing and the data deluge," *Science*, vol. 331, no. 6018, pp. 717–719, 2011.

[42] L. Yuan, Y. Wang, P. M. Thompson, V. A. Narayan, and J. Ye, "Multi-source learning for joint analysis of incomplete multi-modality neuroimaging data," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 1149–1157.

[43] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[44] L. Qiao, S. Chen, and X. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognition*, vol. 43, no. 1, pp. 331 – 341, 2010.

[45] X. Yang, X. Xu, and C. Liu, "Eyebrow recognition based on sparsity preserving projections," in *Conference Anthology, IEEE*, Jan 2013, pp. 1–4.

[46] X. He and P. Niyogi, "Locality preserving projections," 2002.

[47] J. Silva, J. Marques, and J. Lemos, "Selecting landmark points for sparse manifold learning," in *Advances in Neural Information Processing Systems 18*, 2005, pp. 1241–1248.

[48] T. Tong Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression," *Annals of Applied Statistics*, vol. 2, no. 1, pp. 224–244, 2008.

[49] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 58, pp. 267–288, 1996.

[50] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist*, 2004.

[51] J. Wang, Z. Zhang, and H. Zha, "Adaptive manifold learning," in *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press, 2005, pp. 1473–1480.

[52] S. Wang, J. Yao, and R. M. Summers, "Improved classifier for computer-aided polyp detection in ct colonography by nonlinear dimensionality reduction," *Med Phys*, vol. 35, no. 4, pp. 1377–1386, 2008.

[53] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009.

[54] J. Liu, J. Chen, and J. Ye, "Large-scale sparse logistic regression," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 547–556.

[55] X. Zhu, "Semi-supervised learning literature survey," 2006.

[56] K. Bache and M. Lichman, "UCI Machine Learning Repository," 2013.

[57] J. Yang, D. Zhang, Z. Jin, and J.-Y. Yang, "Unsupervised discriminant projection analysis for feature extraction," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1, 2006, pp. 904–907.

[58] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, Oct 2005, pp. 1208–1213.

[59] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences*, vol. 163, 1998.

[60] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," Columbia University, Tech. Rep. CUCS-005-96, 1996.

[61] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," 1997.

[62] L. Tran, D. Banerjee, J. Wang, A. Kumar, F. McKenzie, Y. Li, and J. Li, "High-dimensional mri data analysis using a large-scale manifold learning approach," *Machine Vision and Applications*, vol. 24, no. 5, pp. 995–1014, 2013.

[63] D. Pauleit, K.-J. Langen, F. Floeth, H. Hautzel, M. J. Riemenschneider, G. Reifenberger, N. J. Shah, and H.-W. Muller, "Can the apparent diffusion coefficient be used as a noninvasive parameter to distinguish tumor tissue from peritumoral tissue in cerebral gliomas?" *Journal of Magnetic Resonance Imaging*, vol. 20, no. 5, pp. 758–764, 2004.

[64] M. K. Bode, J. Ruohonen, M. T. Nieminen, and J. Pyhtinen, "Potential of diffusion imaging in brain tumors: A review," *Acta Radiologica*, vol. 47, no. 6, pp. 585–594, 2006.

[65] S. Sinha, M. E. Bastin, I. R. Whittle, and J. M. Wardlaw, "Diffusion tensor mr imaging of high-grade cerebral gliomas," *American Journal of Neuroradiology*, vol. 23, no. 4, pp. 520–527, 2002.

[66] M. Castillo, J. K. Smith, L. Kwock, and K. Wilber, "Apparent diffusion coefficients in the evaluation of high-grade cerebral gliomas," *American Journal of Neuroradiology*, vol. 22, no. 1, pp. 60–64, 2001.

[67] K. Kono, Y. Inoue, K. Nakayama, M. Shakudo, M. Morino, K. Ohata, K. Wakasa, and R. Yamada, "The role of diffusion-weighted imaging in patients with brain tumors," *American Journal of Neuroradiology*, vol. 22, no. 6, pp. 1081–1088, 2001.

[68] N. Bulakbasi, M. Kocaoglu, F. Ors, C. Tayfun, and T. Ucoz, "Combination of single-voxel proton mr spectroscopy and apparent diffusion coefficient calculation in the evaluation of common brain tumors," *American Journal of Neuroradiology*, vol. 24, no. 2, pp. 225–233, 2003.

[69] W. Lam, W. Poon, and C. Metreweli, "Diffusion mr imaging in glioma: Does it have any role in the pre-operation determination of grading of glioma?" *Clinical Radiology*, vol. 57, no. 3, pp. 219 – 225, 2002.

[70] D. Yang, Y. Korogi, T. Sugahara, M. Kitajima, Y. Shigematsu, L. Liang, Y. Ushio, and M. Takahashi, "Cerebral gliomas: prospective comparison of multivoxel 2d chemical-shift imaging proton mr spectroscopy, echoplanar perfusion and diffusion-weighted mri," *Neuroradiology*, vol. 44, pp. 656–666, 2002.

[71] N. Sadeghi, I. Camby, S. Goldman, H.-J. Gabius, D. Balriaux, I. Salmon, C. Decaesteckere, R. Kiss, and T. Metens, "Effect of hydrophilic components of the extracellular matrix on quantifiable diffusion-weighted imaging of human gliomas: Preliminary results of correlating apparent diffusion coefficient values and hyaluronan expression level," *American Journal of Roentgenology*, vol. 181, no. 1, pp. 235–241, 2003.

[72] M. E. Bastin, S. Sinha, I. R. Whittle, and J. M. Wardlaw, "Measurements of water diffusion and t1 values in peritumoural oedematous brain," *Neuroreport*, vol. 13, no. 10, pp. 1335–1340, 2002.

[73] S. Lu, D. Ahn, G. Johnson, M. Law, D. Zagzag, and R. I. Grossman, "Diffusion-tensor mr imaging of intracranial neoplasia and associated peritumoral edema: Introduction of the tumor infiltration index," *Radiology*, vol. 232, no. 1, pp. 221–228, 2004.

[74] J. M. Provenzale, P. McGraw, P. Mhatre, A. C. Guo, and D. Delong, "Peritumoral brain regions in gliomas and meningiomas: Investigation with isotropic diffusion-weighted mr imaging and diffusion-tensor mr imaging," *Radiology*, vol. 232, no. 2, pp. 451–460, 2004.

[75] J. Li, J. Wang, Y. Shen, Y. Shen, R. McKenzie, and N. Guha-Thakurta, "Brain tumor progression assessment using multiple mri volumes. radiological society of north america (rsna) 95th scientific assembly and annual meeting, oral presentation. chicago, il."

[76] Y. Shen, D. Banerjee, J. Li, A. Chandler, Y. Shen, F. McKenzie, and J. Wang, "Prediction of brain tumor progression using a machine learning technique," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7624, Mar. 2010.

[77] D. Banerjee, L. Tran, J. Li, Y. Shen, F. McKenzie, and J. Wang, "Prediction of brain tumor progression using multiple histogram matched MRI scans," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7963, Mar. 2011.

[78] P. Drineas and M. W. Mahoney, "On the nyström method for approximating a gram matrix for improved kernel-based learning," *J. Mach. Learn. Res.*, vol. 6, pp. 2153–2175, Dec. 2005.

[79] V. de Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 705–712.

[80] J.-W. Xu and K. Suzuki, "Computer-aided detection of polyps in ct colonography with pixel-based machine learning techniques," in *Proceedings of the Second international conference on Machine learning in medical imaging*, ser. MLMI'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 360–367.

[81] T. Hartkens, D. Rueckert, J. A. Schnabel, D. J. Hawkes, and D. L. G. Hill, "Vtk cisg registration toolkit: An open source software package for affine and nonrigid registration of single- and multimodal 3d images." in *Bildverarbeitung fr die Medizin*, ser. CEUR Workshop Proceedings, M. Meiler, D. Saupe, F. Kruggel, H. Handels, and T. M. Lehmann, Eds., vol. 56. Springer, 2002, pp. 409–412.

[82] U. Bağcı, J. K. Udupa, and L. Bai, "The role of intensity standardization in medical image registration," *Pattern Recogn. Lett.*, vol. 31, no. 4, pp. 315–323, Mar. 2010.

[83] A. Madabhushi, J. K. Udupa, and A. Souza, "Generalized scale: theory, algorithms, and application to image inhomogeneity correction," *Comput. Vis. Image Underst.*, vol. 101, no. 2, pp. 100–121, Feb. 2006.

[84] L. G. Nyúl and J. K. Udupa, "On standardizing the mr image intensity scale," *Magnetic Resonance in Medicine*, vol. 42, no. 6, pp. 1072–1081, 1999.

[85] R. Duda, P. Hart, and D. Stork, *Pattern classification*, ser. Pattern Classification and Scene Analysis: Pattern Classification. Wiley, 2001.

[86] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[87] L. Tran, D. Banerjee, X. Sun, J. Wang, A. J. Kumar, D. Vinning, F. D. McKenzie, Y. Li, and J. Li, "A large-scale manifold learning approach for brain tumor progression prediction," in *Proceedings of the Second international conference on Machine learning in medical imaging*, ser. MLMI'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 265–272.

[88] L. Tran, F. McKenzie, J. Wang, and J. Li, "Large-scale manifold learning using an adaptive sparse neighbor selection approach for brain tumor progression prediction," in *Machine Learning in Medical Imaging*, ser. Lecture Notes in Computer Science, G. Wu, D. Zhang, D. Shen, P. Yan, K. Suzuki, and F. Wang, Eds. Springer International Publishing, 2013, vol. 8184, pp. 219–226.

[89] M.-A. Belabbas and P. J. Wolfe, "Spectral methods in machine learning and new strategies for very large datasets," *Proceedings of the National Academy of Sciences*, vol. 106, no. 2, pp. 369–374, 2009.

[90] M.-A. Belabbas and P. Wolfe, "On landmark selection and sampling in high-dimensional data analysis," *Philosophical Transactions of the Royal Society A: Mathematical,Physical and Engineering Sciences*, vol. 367, pp. 4295–4312, 2009.

[91] K. Zhang and J. T. Kwok, "Clustered nystrom method for large scale manifold learning and dimension reduction," *Trans. Neur. Netw.*, vol. 21, no. 10, pp. 1576–1587, Oct. 2010.

[92] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering in large graphs and matrices," in *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA '99. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999, pp. 291–299.

Appendix A

REUSE LICENSES

Figure 52: Reuse license for Figure 13, page 1

ELSEVIER LICENSE
TERMS AND CONDITIONS

Mar 28, 2014

This is a License Agreement between Loc Tran ("You") and Elsevier ("Elsevier") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

Supplier	Elsevier Limited The Boulevard,Langford Lane Kidlington,Oxford,OX5 1GB,UK
Registered Company Number	1982084
Customer name	Loc Tran
Customer address	11 Lake Ovide Ct
	Hampton, VA 23669
License number	3357810874193
License date	Mar 28, 2014
Licensed content publisher	Elsevier
Licensed content publication	Neurocomputing
Licensed content title	Sample-dependent graph construction with application to dimensionality reduction
Licensed content author	Bo Yang,Songcan Chen
Licensed content date	December 2010
Licensed content volume number	74
Licensed content issue number	1-3
Number of pages	14
Start Page	301
End Page	314
Type of Use	reuse in a thesis/dissertation
Intended publisher of new work	other
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Format	both print and electronic

Figure 53: Reuse license for Figure 13, page 2

Are you the author of this Elsevier article?	No
Will you be translating?	No
Title of your thesis/dissertation	High Dimensional Data Set Analysis using a Large-Scale Manifold Learning Approach
Expected completion date	Jun 2014
Estimated size (number of pages)	100
Elsevier VAT number	GB 494 6272 12
Permissions price	0.00 USD
VAT/Local Sales Tax	0.00 USD / 0.00 GBP
Total	0.00 USD

Terms and Conditions

INTRODUCTION

1. The publisher for this copyrighted material is Elsevier. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

GENERAL TERMS

2. Elsevier hereby grants you permission to reproduce the aforementioned material subject to the terms and conditions indicated.

3. Acknowledgement: If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication/copies. Suitable acknowledgement to the source must be made, either as a footnote or in a reference list at the end of your publication, as follows:

"Reprinted from Publication title, Vol /edition number, Author(s), Title of article / title of chapter, Pages No., Copyright (Year), with permission from Elsevier [OR APPLICABLE SOCIETY COPYRIGHT OWNER]." Also Lancet special credit -"Reprinted from The Lancet, Vol. number, Author(s), Title of article, Pages No., Copyright (Year), with permission from Elsevier."

4. Reproduction of this material is confined to the purpose and/or media for which permission is hereby given.

5. Altering/Modifying Material: Not Permitted. However figures and illustrations may be altered/adapted minimally to serve your work. Any other abbreviations, additions, deletions and/or any other alterations shall be made only with prior written authorization of Elsevier Ltd. (Please contact Elsevier at permissions@elsevier.com)

6. If the permission fee for the requested use of our material is waived in this instance, please be

Figure 54: Reuse license for figures and text in Chapter 5

Rightslink Printable License

SPRINGER LICENSE
TERMS AND CONDITIONS

Mar 28, 2014

This is a License Agreement between Loc Tran ("You") and Springer ("Springer") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Springer, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	3357820564075
License date	Mar 28, 2014
Licensed content publisher	Springer
Licensed content publication	Machine Vision and Applications
Licensed content title	High-dimensional MRI data analysis using a large-scale manifold learning approach
Licensed content author	Loc Tran
Licensed content date	Jan 1, 2013
Volume number	24
Issue number	5
Type of Use	Thesis/Dissertation
Portion	Full text
Number of copies	1
Author of this Springer article	Yes and you are a contributor of the new work
Order reference number	
Title of your thesis / dissertation	High Dimensional Data Set Analysis using a Large-Scale Manifold Learning Approach
Expected completion date	Jun 2014
Estimated size(pages)	100
Total	0.00 USD
Terms and Conditions	

Introduction

The publisher for this copyrighted material is Springer Science + Business Media. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

Figure 55: Reuse license for figures and text in Chapter 6

SPRINGER LICENSE
TERMS AND CONDITIONS

Mar 28, 2014

This is a License Agreement between Loc Tran ("You") and Springer ("Springer") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Springer, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	3357820855146
License date	Mar 28, 2014
Licensed content publisher	Springer
Licensed content publication	Springer eBook
Licensed content title	Large-Scale Manifold Learning Using an Adaptive Sparse Neighbor Selection Approach for Brain Tumor Progression Prediction
Licensed content author	Loc Tran
Licensed content date	Jan 1, 2013
Type of Use	Thesis/Dissertation
Portion	Full text
Number of copies	100
Author of this Springer article	Yes and you are a contributor of the new work
Order reference number	
Title of your thesis / dissertation	High Dimensional Data Set Analysis using a Large-Scale Manifold Learning Approach
Expected completion date	Jun 2014
Estimated size(pages)	100
Total	0.00 USD
Terms and Conditions	

Introduction

The publisher for this copyrighted material is Springer Science + Business Media. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

Limited License

With reference to your request to reprint in your thesis material on which Springer Science and

VITA

Loc Tran

Department of Electrical and Computer Engineering

Old Dominion University

Norfolk, VA 23529

Loc Tran attended Old Dominion University for his undergraduate degree. He completed his undergraduate degree in Electrical Engineering in May 2008. He continued his education at Old Dominion University in pursuit of a Ph.D. Degree in Electrical and Computer Engineering. During his graduate work, he was a researcher at the Old Dominion University Computer Vision Laboratory along with the MIDA lab. He was funded during his first four years of graduate work by the NSF Marine Engineering Scholarship.

Typeset using LaTeX.

LIST OF PUBLICATIONS

JOURNAL ARTICLES

1. L. Tran, D. Banerjee, J. Wang, A. Kumar, F. McKenzie, Y. Li, and J. Li, "High-dimensional MRI data analysis using a large-scale manifold learning approach," *Machine Vision and Applications*, vol. 24, no. 5, pp. 995-1014, 2013.

CONFERENCE PUBLICATIONS

1. L. Tran, F. McKenzie, J. Wang, and J. Li, "Large-scale manifold learning using an adaptive sparse neighbor selection approach for brain tumor progression prediction," in *Machine Learning in Medical Imaging*, ser. Lecture Notes in Computer Science, G. Wu, D. Zhang, D. Shen, P. Yan, K. Suzuki, and F. Wang, Eds. Springer International Publishing, 2013, vol. 8184, pp. 219-226.

2. L. Tran, D. Banerjee, X. Sun, J. Wang, A. J. Kumar, D. Vinning, F. D. McKenzie, Y. Li, and J. Li, "A large-scale manifold learning approach for brain tumor progression prediction," in *Proceedings of the Second international conference on Machine Learning in Medical Imaging*, ser. MLMI'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp 265-272.

3. L. Tran, D. Banerjee, Xiaoyan Sun, Jihong Wang, Jiang Li, et al., "A large-scale manifold learning approach for brain tumor progression prediction," *MICCAI MLMI workshop*, Toronto, Canada 2011 (Oral presentation, acceptance rate: 20

4. D. Banerjee, L. Tran, J. Li, Y. Shen, F. McKenzie, and J. Wang, "Prediction of brain tumor progression using multiple histogram matched MRI scans," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7963, Mar. 2011.

5. L. Tran, J. Selfridge, G. Hou, and J. Li, "Marine Buoy Detection using Circular Hough Transform," *AUVSI Unmanned Systems North America 2011*, Washington DC, August 16-18, 2011.

6. J. Selfridge, L. Tran, and G. Hou, "Autonomous Vehicle Path Planning and Tracking: A Vision Based Approach," *AUVSI Unmanned Systems North America 2011*, Washington DC, August 16-18, 2011.