# Locality Preserving Projections

An Introductaty Presentation

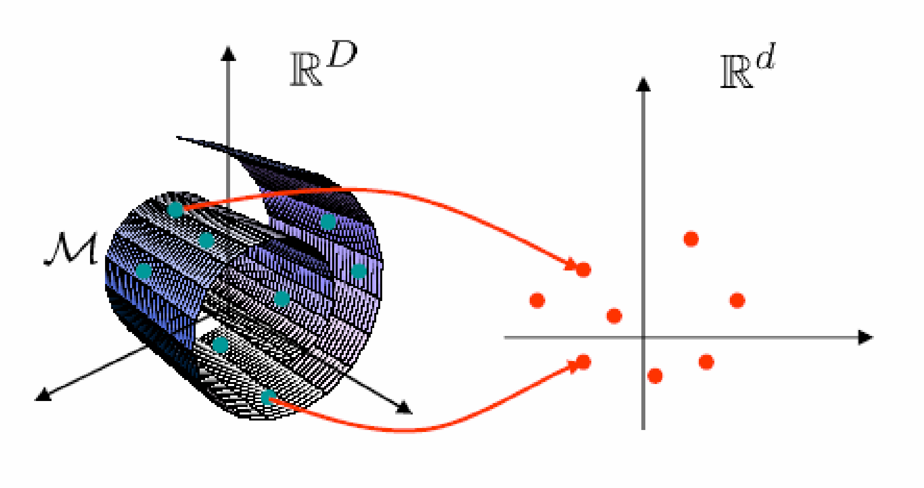
# Introduction

PCA performs dimensionality reduction by projecting the original n-dimensional data onto a $k \ll n$ dimensional linear subspace spanned by the leading eigenvectors of the data’s covariance matrix

# Introduction

LDA searches for the projective axes on which the data points of different classes are far from each other (maximize between class scatter), while constraining the data points of the same class to be as close to each other as possible (minimizing within class scatter).

# The GOAL of LPP

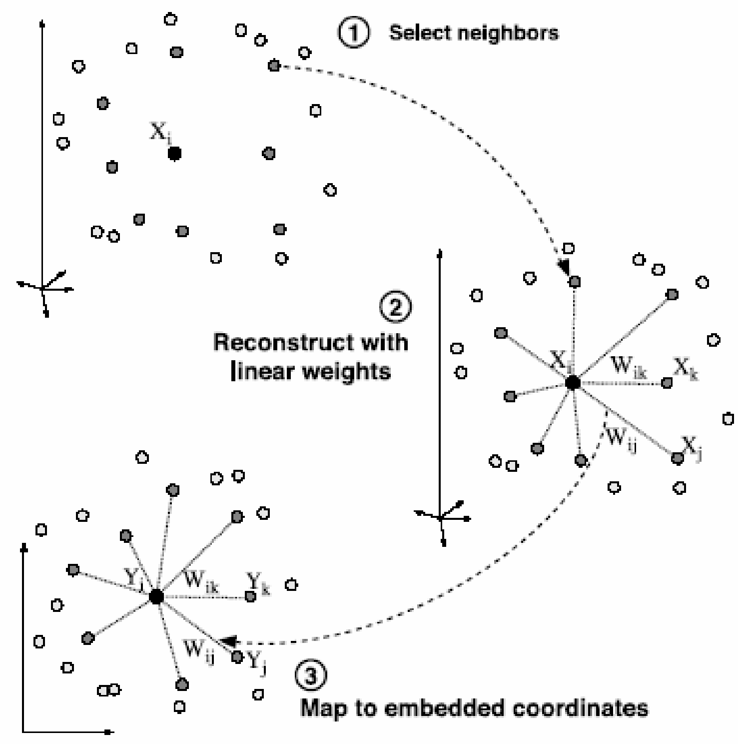



***"Preserving locality information by finding a projection that Minimize the sum of the squared distance from one sample to its neighborhood samples after projection."***

# LPP

•LPP builds a graph incorporating neighborhood information of the data set.

•Using the notion of the Laplacian of the graph, we then compute a transformation matrix which maps the data points to a subspace.



# LPP algorithm

- Constructing the adjacency graph
- Choosing the weights
- Eigenmaps

# Constructing the adjacency graph

1. **Constructing the adjacency graph**: Let $G$ denote a graph with $k$ nodes. We put an edge between nodes $i$ and $j$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close". There are two variations:

   **(a)** $\epsilon$-neighborhoods. [parameter $\epsilon \in \mathbf{R}$] Nodes $i$ and $j$ are connected by an edge if $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon$ where the norm is the usual Euclidean norm in $\mathbf{R}^l$.

   **(b)** $n$ nearest neighbors. [parameter $n \in \mathbf{N}$] Nodes $i$ and $j$ are connected by an edge if $i$ is among $n$ nearest neighbors of $j$ or $j$ is among $n$ nearest neighbors of $i$.

# Choosing the weights

2. **Choosing the weights**: Here, as well, we have two variations for weighting the edges:

   **(a)** Heat kernel. [parameter $t \in \mathbf{R}$]. If nodes $i$ and $j$ are connected, put

$$W_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}$$

   The justification for this choice of weights can be traced back to [1].

   **(b)** Simple-minded. [No parameter]. $W_{ij} = 1$ if and only if vertices $i$ and $j$ are connected by an edge.

# Eigenmaps

3. **Eigenmaps**: Compute the eigenvectors and eigenvalues for the generalized eigenvector problem:

$$XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a}$$

where $D$ is a diagonal matrix whose entries are column (or row, since $W$ is symmetric) sums of $W$, $D_{ii} = \Sigma_j W_{ji}$. $L = D - W$ is the Laplacian matrix. The $i^{th}$ column of matrix $X$ is $\mathbf{x}_i$.

Let $\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{k-1}$ be the solutions of equation (1), ordered according to their eigenvalues, $\lambda_0 < \lambda_1 < \cdots < \lambda_{k-1}$. Thus, the embedding is as follows:

$$\mathbf{x}_i \rightarrow \mathbf{y}_i = A^T\mathbf{x}_i$$

$$A = \begin{pmatrix} \mathbf{a}_0, & \mathbf{a}_1, & \cdots, & \mathbf{a}_m \end{pmatrix}$$

where $\mathbf{y}_i$ is a $m$-dimensional vector, and $A$ is a $l \times m$ matrix.

# Finally

Finally the images are represented by A which is the d eigen vectors corresponding the d smallest eigen values of XLXT

Thus it is simillar to the PCA algorithm with these eigen vectors.

# Bibilography


- Locality Preserving Projections (LPP) by Xiaofei He and Partha Niyogi
- Incremental Semi-Supervised Subspace Learning for Image Retrieval by Xiaofei He