COURSE PROJECT- CS678A: LEARNING WITH KERNELS
UNDER THE SUPERVISION OF PROF. H. C. KARNICK

Dimensionality Reduction : A comparative study

Aayush Mudgal
mudgal@iitk.ac.in
12008

Sheallika Singh
shealika@iitk.ac.in
12665

February 13, 2016

Abstract

We provide a comparative and detailed study on the existing Dimensional Reduction techniques. We did a detailed literature review about the existing techniques and provide the respective details in a succinct manner.

1 INTRODUCTION

Dimension Reduction refers to the mapping of data to a lower dimensional space such that uninformative variance is discarded, or such that a subspace in which the data lives is detected. It has been used for data visualization and data extraction of key low dimensional features. For example the two dimensional orientation of an object, from its high dimensional image representation is often utilized in practical purposes as a first step towards data visualization. It can also lead us to understand and develop better models for inference. If the data lies (approximately) in a low dimensional manifold L that happens to be embedded in a high dimensional manifold H, then modeling the data directly in L rather than in H may turn an unfeasible problem to a feasible problem.

The techniques of dimensional reduction often explicit or implicitly assume that the data often lies on or near a much lower dimensional, curved manifold. And a good way to represent data points is by their low-dimensional co-ordinates. The low-dimensional representation of the data should capture information about high-dimensional pairwise distances.

Dimensional reduction techniques can be categorized into Linear and Non-Linear techniques as shown in Figure1.1. Linear techniques of dimensional reduction assume that the lower dimensional representation of the data is essentially a linear combination of the higher dimensional representation, hence the term linear. Different techniques could also be categorized into Global and local on the basis of the relative importance given to nearby and/or far-by distances. Global methods on one hand assume that all the pairwise distances are of equal importance. It tries to choose the lower-dimensional pairwise distances to fit the high-dimensional ones, using some magnitude or rank-order. On the other hand, local methods assume that only the local distances are reliable in higher-dimensional space and thus put more weight

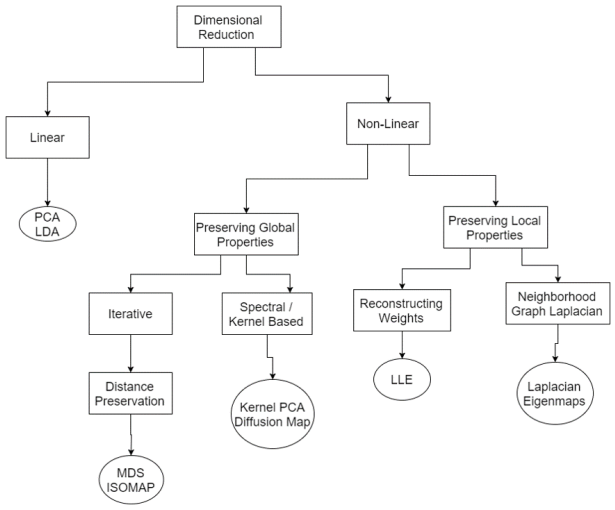


Figure 1.1: Differnet types of dimensional reduction techniques

on modeling the local distances correctly.Global methods assume that all pairwise distances are of equal importance.Choose the low-D pairwise distances to fit the high-dimensional ones (using magnitude or rank order).

These techniques could also be studied under two broad-categories, namely the Projective methods and the Manifold Methods. Projective methods are perhaps the simplest approach. They attempt to find low dimensional projections that extract useful information from the data by maximizing a suitable objective function. Apart from handling higher dimensional data, projection pursuit methods can be robust to noisy or irrelevant features, and have been applied to regression [6] Some of the Projective methods are as follows:

- Independent Component Analysis (ICA)
- Principal Component Analysis (PCA)
- Kernel PCA (KPCA)
- Canonical Correlation Analysis (CCA)

while some of the manifold methods are as follows:

- MultiDimensional Scaling (MDS)
- Locally Linear Embedding (LLE)
- Diffusion Maps (DM)
- Laplacian EigenMaps (LEM)

2 PROJECTIVE METHODS

2.1 INDEPENDENT COMPONENT ANALYSIS (ICA)

Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. [4] mentions that ICA searches for the projections such that the probability distributions of the data along these projections are statistically

independent. Independent Component Analysis draws its motivation from the standard cocktail party problem, which assumes that there are two independent speakers speaking into two microphones, where each microphone captures sound from both the speakers. The task is to separate the two original source signals from their mixtures. The microphone signal can be written as $y = Ax, x, y \in R^2$, where the components of x (assumed statistically independent and 0 mean) are the signals from each individual speaker. The trouble point is that here only y is known and both A, and x are unknown.

To give non-trivial results ICA requires that the in the original signals at most one is Gaussian distributed. ICA components can be estimated by one of the following ways:

- Finding the maximally non-gaussian component
- Finding the component with minimum mutual information

2.2 Principal Component Analysis (PCA)

Principal component analysis (PCA) uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. It tries to capture the direction of maximum variance. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. The principal components being the eigenvectors of the symmetric covariance matrix, are orthogonal. PCA is sensitive to the relative scaling of the original variables. PCA seeks to represent data in the sense that it minimizes squared reconstruction error and preserve the variance expressed in the data, at the same decorelating the data. It also maximizes the mutual information on Gaussian data. The algorithm to compute the principal components is described as below.

Firstly Compute the covariance matrix C, and perform the eigendecomposition of the matrix C. Let A be the diagonal matrix comprising eigenvalues of C, therefore, where $\lambda_i \equiv A_{ii}$ and $\lambda_i \geq \lambda_{i+1} \forall i$, then

$$CE = EA \tag{2.1}$$

Now for some unit vector $n_1 \in R_d$, consider

$$n_1^{'} E^T CE n_1 =_1^{'} An_1 \tag{2.2}$$

The left hand side is the variance of the projections of the data along the unit vector En_1. The right hand side is $\sum_i n_{1_i}^2 \lambda_i$ and since $\sum_i n_{1_i}^2 = 1$, this is a convex optimization problem, which is maximized by chosing the largest λ, i.e. by choosing $n_{1_i} = \lambda_{i,1}$. For that choice of n_1, En_1 is the principal eigenvector, and the variance of the data projected along that direction is just λ_1. Applying the same argument iteratively shows that the eigenvectors of C give the desired directions, and the corresponding variances are the λ. We will take the first k principal eigenvectors, where $\sum_{i=1}^{k} \lambda_i \approx \sum_{i=1}^{d} \lambda_i$. These top k eigenvectors are able to express the maximum variance in the data, and thus forming the basis of the reduced projected space.

2.3 Kernel Principal Component Analysis (K-PCA)

Kernel Principal Component Analysis is an extension of PCA using the kernel trick. Using the kernel trick it allows us to perform PCA is some higher dimensional feautre space (a reproducing Kernel Hilbert Space), resulting in a non-linear relationships in the input space. It requires tuning of Kernel parameters, but is similar to PCA in terms of time and space complexity PCA method uses linear projection which can limit the usefulness of the approach, on the other hand Kernel PCA can overcome this problem as it is non linear method PCA depends only on first and second moments of the data whereas kernel PCA does not. Kernel PCA followed by a linear SVM on a pattern recognition problem has shown to give similar results to using a nonlinear SVM using the same kernel. Moreover Kernel PCA is not affected by the noise in the data

2.4 CANONICAL CORRELATION ANALYSIS (CCA)

Canonical correlation analysis (CCA) is a way of measuring the linear relationship between two multidimensional variables. The basic idea is to find two bases, one for each variable, that are optimal with respect to correlations and, at the same time, finding the corresponding correlations. In other words, it finds the two bases in which the correlation matrix between the variables is diagonal and the correlations on the diagonal are maximized. The dimensionality of these new bases is equal to or less than the smallest dimensionality of the two variables. In other words, CCA simultaneously finds dimension reduction for two feature spaces CCA can be summarised in short as following. Given two X_1, X_2 random vectors, with ranges in R^{d_1}, R^{d_2}, and also assume that expectations of product of random variables that are the components of X's is computable. Further assume the following

$$E[X_{1a}] = E[X_{2b}] = 0 \text{ where a=1,...,} d_1 \text{ b=1,...,} d_2 \tag{2.3}$$

. Define $U \equiv X_1.w_1, V \equiv X_2.w_2$ for some $w_1 \in R^{d_1}$ and $w_2 \in R^{d_2}$. The goal is to find w_1, w_2 such that the correlation as given below in 2.4 is maximized.

$$\rho \equiv \frac{E[UV]}{\sqrt{E[U^2]E[V^2]}} = \frac{w_1^{'} C_{12} w_2}{\sqrt{(w_1^{'} C_{11} w_1)(w_2^{'} C_{22} w_2)}} = \frac{A_{12}}{\sqrt{A_{11} A_{22}}} \tag{2.4}$$

Among the most important and useful property of the canonical correlations is that they are invariant with respect to affine transformation of the variables, and this is the basic distinguishing feature from the ordinary correlation analysis. CCA can be visualized as an extension of PCA to two paired data sets, and since PCA can be kernelized, and was successfully shown independently by Akaho and Bach and Jordan.Kernel CCA follows Kernel PCA in spirit. CCA also shares with PCA the property that the projections decorrelate the data. For CCA, the projections decorrelate the individual data sets just as for PCA, but also the cross-correlation of the projected data vanishes, and the direction are conjugate with respect to the cross-covariance matrices.

3 MANIFOLD MODELING

Before understanding Manifold Modeling, it is important to understand the two important terms, manifold and embedding. *Manifold*: A manifold is a topological space which is locally Euclidean. In general, any object which is nearly "flat" on small scales is a manifold *Embedding*: is a representation of a topological object, manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved. Manifold modeling techniques generally assume that the data lives on some manifold $M \subset R^{d^{'}}$ embedded in R^d, and the inputs are samples taken in R_d of the underlying manifold M, and the output is the lower dimensional representation which preserves the manifold structure of the data, as defined by some metric of interest.

3.1 MULTIDIMENSIONAL SCALING (MDS)

MultiDimensional Scaling is a manifold modeling algorithm that arose first in behavioral science [3] Multidimensional scaling is one of the several multivariate techniques that aim to reveal the structure of a data set by plotting points in one or two dimensions. MDS starts with a measure of dissimilarity between each pair of data pints in the data set (this measure can be very general and may involve non-vecotrial data). Given the dissimilarity matrix, MDS searches for a mapping of the dissimilarities to a low dimensional Euclidean space such that the (transformed) pairwise dissimilarities become squared distances. Although we may start with a proximity or similarity matrix, it may need to converted to a distance matrix in the course of the analysis, the output is expressed in terms of the distance.

In simpler terms, MDS is used to determine whether the distance matrix may be represented by a map or a configuration in a smaller number of dimensions such that the distances on the map reproduce approximately the original distance matrix.

3.1.1 CLASSICAL MDS

In classical MDS the aim is to find a configuration in a low number of dimensions such that the distances between the points in the configuration d_{ij} are close in value to the observed distances δ_{ij}. It treats the distances as Euclidean Distances. The problem is tackled algebraically, giving a series of approximations starting with one dimension, then two and so on. The quality of the mapping is expressed with the help of the stress function, a measure of the error between the pairwise distances in the low-dimensional and high-dimensional representation of the data. Two important examples of stress functions (for metric MDS) are the raw stress function3.1 and the Sammon cost function 3.2.The raw stress function is given by

$$\phi(Y) = \sum_{ij}(||x_i - x_j|| - ||y_i - y_j||)^2 \tag{3.1}$$

where $||x_i x_j||$ is the euclidean distance between the higher dimensional points x_i and x_j and $||y_i - y j||$ is the euclidean distance between the lower dimensional points. Another common stress function is the Sammon cost function which is given as follows:

$$\phi(Y) = \frac{1}{\sum_{ij}||x_i - x_j||}\sum_{i\neq j}\frac{(||x_i - x_j|| - ||y_i - y_j||)^2}{||x_i - x_j||} \tag{3.2}$$

The Sammon cost in contrast to the raw-stress function puts more importance on retaining distances that were originally very small. The minimization of the stress function can be performed using one of the many methods, such as the eigendecomposition of a pairwise dissimilarity matrix, the conjugate gradient method, or a pseudo-Newton method.

3.1.2 ORDINAL MDS

In ordinal MDS the aim is to find a configuration such that the d_{ij} are in the same rank order as that of the original δ_{ij}. In ordinal MDS, fitted distances often called *disparities* $\hat{d_{ij}}$ from the d_{ij}, such that the $\hat{d_{ij}}$ are n the same rank order as the δ_{ij}. Least-squares monotonic regression is used for this smoothing process. The aim of the monotonic regression is to fit a monotonic curve to the points (d_{ij}, δ_{ij}), while making the squared vertical deviations as small as possible. The point on the monotonic curve, $\hat{d_{ij}}$, is the fitted value of d_{ij} from the monotonic regression. The stress functions as defined before are obtained by cleverly replacing $\delta i j$ by $\hat{d_{ij}}$ in the formulae.

3.1.3 LANDMARK MDS

When the datasets are extremely large, MDS is not a feasible option since it is computationally expensive. Since the distance matrix is not sparse, the computational complexity of the eigendecomposition is $O(m^3$, making MDS a poor choice for many practical purposes. Landmark MDS significantly reduces this bottleneck. LMDS chooses some q points as landmarks, such as $q > r$ (where r is the rank of the distance matrix) but $q << m$. Classical MDS is performed on these chosen landmarks. The remaining points are mapped to R_d using only their distances to the landmark points. In other words as pointed by Bengio et al. [2] LMDS combines MDS with Nystrom algorithm.This increases the scalibility but at the same time, uses an approximation to map points.

LDMS has two significant advantages, first it reduces the computational complexity form $O(m^3$ to $O(q^2 m)$ and also since it can be applied to any non-landmark point, it gives a method of extending MDS (using Nystrom) to out-of-sample data.

3.2 ISOMETRIC FEATURE MAP (ISOMAP)

Multidimensional scaling has shown to be successful in many applications, but it suffers from the fact that it is based on Euclidean Distances, and does not take into account the distribution of the neighboring data-points.If the high-dimensional data lies on or near a curved manifold, such as in the Swiss roll dataset

(as shown in Figure [3.1]), MDS might consider two data-points as near points, whereas their distance over the manifold is much larger than the typical inter-point distance.

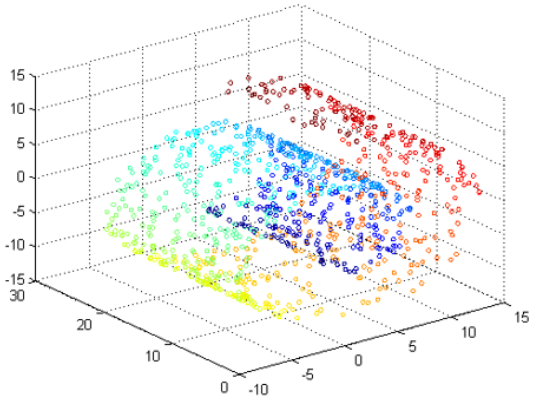


Figure 3.1: Swiss Roll

Though MDS is useful for extracting low dimensional representations for some kinds of data, but it does not attempt to explicitly model the underlying manifold. In contrast to MDS, Isomap and LLE directly attempt to model the manifold. The key assumption made by Isomap is that when comparing two points, the distance along the curve between the two points is of greater importance. Even if the two points are close in R_D, this larger distance along the curve is a better representative of the distance between them.

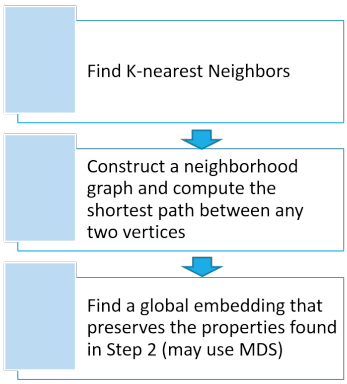


Figure 3.2: General Scheme of ISOMAP technique

Isomap resolves this problem by attempting to preserve the pairwise geodesic (or curvilinear) distances .[1] between the data-points. The basic step is to construct a graph whose nodes are the data-points, where a pair of nodes are adjacent only if the two points are close in $R^{d'}$, and then to approximate the geodesic distance along the manifold between any two points as the shortest path in the graph, are computed using the Floyd algorithm, and is finally followed by MDS to extract the low-dimensional representation as vectors in $R^{d'}$

Isomap like other manifold mapping techniques does not provide a direct functional form for the mapping $L : R^d \rightarrow R^{d'}$ that can be simply be applied to the new data. Thus out of sample complexity of the

[1] Geodesic distance is the distance between two points measured over the manifold

algorithm is an issue. Most computational intensive part of this is the eignevector computation which is $O(m^3)$, which is because of the MDS step. An important weakness of the Isomap algorithm is its topological instability, it may construct erroneous connections in the neighborhood graph G. Such short-circuiting can severely impair the performance of Isomap. Several approaches have been proposed to overcome the problem of short-circuiting, e.g., by removing datapoints with large total flows in the shortest path-algorithm [5] or by removing nearest neighbors that violate local linearity of the neighborhood graph. A second weakness is that Isomap may suffer from 'holes' in the manifold. This problem can be dealt with by tearing manifolds with holes [7] A third weakness of Isomap is that it can fail if the manifold is non convex [15] . Despite these three weaknesses, Isomap was successfully applied on tasks such as wood inspection [8], visualization of biomedical data[7], and head pose estimation [10].

3.3 Locally Linear Embedding (LLE)

Locally Linear Embedding models the manifold by assuming that the manifold is approximately linear when viewed locally. It models the manifold by treating it as a union of linear patches. [9][12] LLE is a type of graph based algorithm, and all almost all of the graph based algorithms follow a certain patter as shown in figure 3.3.

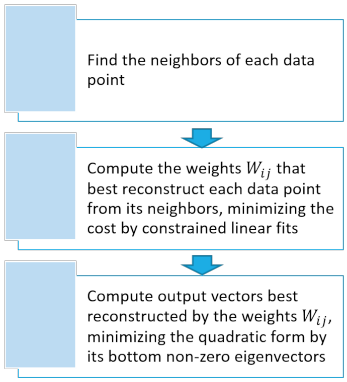


Figure 3.3: General Scheme of LLE technique

Suppose that each point $x_i \in R^d$ has a small number of close neighbors indexed by the set $N(i)$, and let $y_i \in R^{d'}$ be the corresponding low dimensional representation of x_i. The motivation behind LLE is to express each of the x_i as a linear combination of its neighbors and then construct its corresponding lower dimensional representation using its corresponding neighbors. For simplification, let us consider only 'k' nearest neighbors of a point. The condition on the $x^{'}s$ can be expressed as finding the appropriate $W \in M_{mn}$ that minimizes the reconstruction error given by 3.3

$$\sum_i \left\| (x_i - \sum_{j \in N(i)} W_{ij} * x_j) \right\|^2 \tag{3.3}$$

$$E_i = \sum_i \left\| (x_i - \sum_{j \in N(i)} W_{ij} * x_j) \right\|^2 \tag{3.4}$$

But Each of the reconstruction error E_i must be unaffected by any global translation of x_i, which gives the following condition 3.5

$$\sum_{j \in N(i)} W_{ij} = 1 \forall i \tag{3.5}$$

The objective function after enforcing the constraints through Lagrange multipliers λ_i is as given below 3.6

$$F \equiv \sum_i F_i \equiv \sum_i \frac{1}{2} (\left\| (x_i - \sum_{j \in N(i)} W_{ij} * x_j) \right\|^2 - \lambda_i (\sum_{j \in N(i)} W_{ij} - 1)) \tag{3.6}$$

Given the $W^{'}s$ the second step is to find a set of representative lower dimensional vectors $y_i \in R^{d^{'}}$ that can be expressed in terms of each other using the same weight vectors $W^{'}s$. Again no exact solution may exist and so $\sum_i \left\| y_i - \sum_{j \in N(i)} W_{ij} * y_j \right\|^2$ is minimized with respect to the y's keeping the same weight matrix. The objective function to be minimized in this condition is given by 3.7

$$F = \frac{1}{2} \sum_i \left\| y_i - \sum_j W_{ij} y_j \right\|^2 - \frac{1}{2} \sum_{\alpha\beta} \lambda_{\alpha\beta} \left(\sum_i \frac{1}{m} Y_{i\alpha} Y_{i\beta} - \delta_{\alpha\beta} \right) \tag{3.7}$$

LLE have a very desirable property that they will result in the same weight W if the data is rotated, scaled, reflected and/or translated. It requires a two step process

- Finding the W's has $O(n^3 m)$ computational complexity
- The second step requires eigen-decomposing of two sparse matrices in M_m

The only free parameter is the dimensionality of the latent space $d^{'}$ and the number of neighbors that are used to determine the local weights. The n x n matrix is sparse. The convex optimization problem also guarantees the global minimum and thus doesn't require multiple tries to solve the objective function. The major drawback of this technique is that it might not be optimizing the right things and moreover it has no incentive to keep widely separated data points far apart in the low-dimensional space. LLE also suffers from the collapsing problem. For example, consider the case where the neighborhood graph has several disconnected pieces, the unit variance constraint could be satisfied but still resulting in collapses. Even if the graph is fully connected, it may be possible to collapse all the densely connected regions and satisfy the variance constraint by paying a high cost for a few outliers.

3.4 LAPLACIAN EIGENMAPS (LEM)

Similar to the Locally Linear Embedding, Laplacian EigenMaps finds a low-dimensional data representation by preserving the local properties of the manifold. Here, the local properties are based on the pairwise distances between the neighbors. LEM compute a low dimensional representation of the data in which the distances between a data-point and its 'k' nearest neighbors is minimized in a weighted manner. In simpler words the distance in the low-dimensional data representation between a data-point and its first nearest neighbor contributes more to the cost function than any other neighbor of that data-point. Using spectral graph theory, the minimization of the cost function is defined as an eigenvalue problem. The algorithmic overview is as provided below.

It first constructs a neighborhood graph G in which every data-point x_i is connected to its k nearest neighbors. For all points x_i and x_j in the neighborhood graph, that are connected by an edge, the weight of the edge is computed using the guassian kernel function as shown in 3.8 or simply as shown in 3.9, leading to a sparse adjacency matrix W. In the computation of the low-dimensional representations y_i the cost function that is minimized is given by 3.10

$$w_{ij} = \epsilon^{\frac{\|x_i - x_j\|^2}{2\sigma^2}} \text{ if } x_i \text{ and } x_j \text{ are connected} \tag{3.8}$$

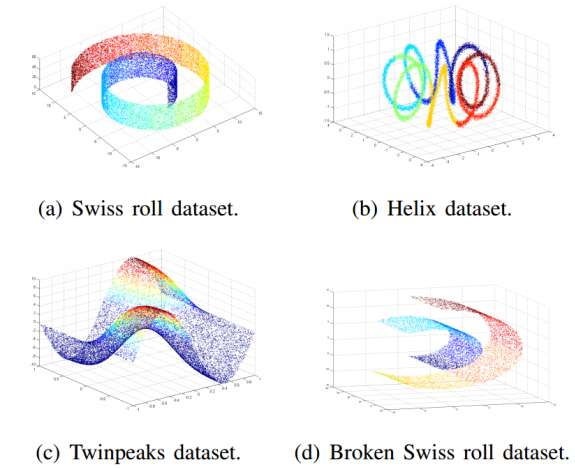
$$w_{ij} = 1 \text{ if } x_i \text{ and } x_j \text{ are connected} \tag{3.9}$$

$$\phi(Y) = \sum_{ij} (y_i - y_j)^2 w_{ij} \tag{3.10}$$

Larger the weight w_{ij}, smaller is the distance between the points x_i and x_j. Hence the difference between their low-dimensional representations y_i and y_j highly contributes to the cost function. As a result, nearby points in the higher dimensional space are closer together even in the low-dimensional space.

The computation of the degree matrix M and the graph Laplacian L of the graph W allows for formulating the minimization problem as an eigenproblem The degree matrix M of W is a diagonal matrix of which the entries are the row sums of W (i.e. $m_{ii} = \sum_j w_{ij}$. The graph Laplacian L is computed by L = M-W. Equation 3.10 could be shown to be equivalent to as shown in 3.14

$$\phi(Y) = \sum_{ij} (y_i^2 + y_j^2 - 2 y_i y_j) w_{ij} \tag{3.11}$$



(a) Swiss roll dataset. (b) Helix dataset.

(c) Twinpeaks dataset. (d) Broken Swiss roll dataset.

Figure 4.1: Artificial Data-sets

$$\phi(Y) = \sum_i y_i^2 m_{ii} + \sum_j y_j^2 m_{jj} - 2\sum_{ij} y_i y_j w_{ij} \tag{3.12}$$

$$\phi(Y) = 2Y^{'} MY - 2Y^{'} WY \tag{3.13}$$

$$\phi(Y) = 2Y^{'} LY \tag{3.14}$$

Hence, minimizing $\phi(Y)$ is proportional to minimizing $Tr(Y^{'} LY)$ subject to the condition that $Y^{'} DY = 1$.

$$Lv = \lambda Mv \tag{3.15}$$

The low-dimensional data representation Y can thus be found by solving the generalized eigenvalue problem 3.15 for the d smallest nonzero eigenvalues. The d eigenvectors corresponding to the smallest nonzero eigenvalues from the low-dimensional representation.

4 DO EXISTING NON-LINEAR TECHNIQUES REALLY HELP?

A systematic empirical comparison of the performance of the linear and non-linear techniques for dimensional reduction by [16] conclude that existing non-linear techniques are yet to outperform the good old PCA techniques on many of the natural datasets. They perform the evaluation by measuring generalization errors in classification tasks on two types of datasets: (1) artificial datasets and (2) natural datasets.

4.1 EXPERIMENTS ON THE ARTIFICIAL DATASETS

[16] perform their analysis on 5 artificial datasets which are represented in the Figure 4.1 The data sets were specifically selefcted to investigate how the dimensionality reduction techniques deal with:

- Data that lies on or near a low-dimensional manifold that is or is not isometric to Euclidean space
- Data that lies on or near an discontinuous manifold
- Data forming a manifold with a high intrinsic dimensionality

Table 4.1 presents the generalization errors obtained by [16] of 1-nearest neighbor classifiers trained on the low-dimensional data representations obtained from the dimensional reduction techniques. They reported that the non-linear techniques that employ neighborhood graphs i.e. LLE, Isomap, LEM), outperform the other techniques on standard manifold learning problems such as the Swiss roll dataset. The performance of Isomap is still very strong on manifolds that are not isometric to the Euclidean space (twin peaks and the helix dataset). Thirdly the results of the broken swiss roll indicate that most of the non-linear techniques can not deal with discontinuous manifolds. Lastly, the results on the HD dataset, suggest that most nonlinear techniques have major problems when faced with a dataset with a high intrinsic dimensionality. In particular, local dimensionality reduction techniques perform disappointing on a dataset with a high intrinsic dimensionality. On the HD dataset, PCA is only outperformed by Isomap

Table 4.1: Generalization Errors of 1-NN classifiers trained on artificial datasets

Dataset (d)	None	PCA	Isomap	KPCA	LLE	LEM
Swiss roll (2D)	3.68%	30.56%	3.28%	29.30%	7.44%	10.16%
Helix (1D)	1.24%	38.56%	1.22%	44.54%	20.38%	10.34%
TwinPeak (2D)	0.40%	0.18%	0.30%	0.08%	0.54%	0.52%
Broken Swiss (2D)	2.14%	27.62%	14.24%	27.06%	37.06%	26.08%
HD (5D)	24.19%	22.14%	20.45%	29.25%	35.81%	41.70%

4.2 EXPERIMENTS ON THE NATURAL DATASETS

[16] perform their analysis on 5 natural datasets, that represents tasks from variety of domains

- MNIST dataset: Dataset of 60,000 handwritten digits, wherein each image is of size 28x28 pixels, and can thus be considered as points in a 784 dimensional space
- COIL20 dataset: Dataset of 20 different objects, depicted from 72 viewpoints, leading to a total of 1,440 images, wherein each image is of size 32x32 pixels, and can thus be considered as points in a 1024 dimensional space
- NiSIS dataset: Dataset for pedestrian detection, consists of 3,675 grayscale images of size 36x18 pixels, and can thus be considered as points in a 648 dimensional space
- ORL dataset: Dataset of 400 grayscale images for face detection, wherein each image is of size 112x92 pixels, and can thus be considered as points in a 1,617 dimensional space

Table 4.2: Generalization Errors of 1-NN classifiers trained on artificial datasets

Dataset (d)	None	PCA	Isomap	KPCA	LLE	LEM
MNIST (20D)	5.11%	5.06%	28.54%	65.48%	19.21%	19.45%
COIL20 (5D)	0.14%	3.82%	14.86%	7.78%	9.86%	14.79%
ORL (8D)	2.50%	4.75%	44.20%	5.50%	9.00%	12.50%
NiSIS (15D)	8.24%	8.73%	20.57%	11.70%	28.71%	43.08%
HIVA (15D)	4.63%	5.05%	4.97%	5.07%	5.23%	5.23%

Table 4.2 presents the generalization errors obtained by [16] of 1-nearest neighbor classifiers trained on the low-dimensional data representations obtained from the dimensional reduction techniques. [16] observed that the performance of nonlinear techniques for dimensional reduction on the natural datasets is not quite promising as the case of their performance on the Swiss roll data-set. In particular PCA, outperforms all nonlinear technique in 4 of the five natural datasets. Especially the local non-linear techniques perform quite poor. On the other hand Kernel PCA perform strongly on almost all datasets.

5 DISCUSSION

The results of the experiments performed by [16] show that the performance of popular techniques based on neighborhood graphs is rather disappointing on many datasets. The main reasons for this is attributed to the fact that local dimensional reduction techniques suffer from the curse of dimensionality of the embedded manifold, because the number of data-points required to characterize a manifold properly grows exponentially with the intrinsic dimension of the manifold. These local methods perform well on most of the artificial datasets with low intrinsic dimension. Also the low performance of these local non-linear techniques also arises from the eigenproblems that they solve. Except Isomaps, most of them tend to solve for the smallest eigenvalue (around 10^{-7}), such problems are extremely hard to solve. The eigensolver thus provides with an approximate value, thus causing error in the calculations. These techniques are also questionable because the local properties of a manifold do not necessarily follow the global structure of the problem in the presence of noise around the manifold. In short, local methods suffer from over-fitting.

Kernel-based techniques for dimensional reduction do not suffer from the weakness of neighborhood graph based techniques. [16] suggest that the poor performance of K-PCA on swiss roll dataset is probably because of the incapability in selecting the desired kernel function that could model such complex manifolds. Construction of a proper kernel remains an important obstacle for the successful application of Kernel PCA.

These results suggest that the focus of research should shift towards the development of techniques that have objective functions that can be optimized well in practice.

REFERENCES

[1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[2] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. 2004.

[3] Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[4] Christopher JC Burges. Dimension reduction: A guided tour. *Machine Learning*, 2(4):275–365, 2009.

[5] Heeyoul Choi and Seungjin Choi. Robust kernel isomap. *Pattern Recognition*, 40(3):853–862, 2007.

[6] Jerome H Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

[7] Hongyu Li, Li Teng, Wenbin Chen, and I-Fan Shen. Supervised learning on local tangent space. In *Advances in Neural Networks–ISNN 2005*, pages 546–551. Springer, 2005.

[8] Matthew Partridge and Rafael Calvo. Fast dimensionality reduction and simple pca. 1997.

[9] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[10] Sam T Roweis, Lawrence K Saul, and Geoffrey E Hinton. Global coordination of local linear models. 2002.

[11] Lawrence K Saul and Sam T Roweis. An introduction to locally linear embedding. *unpublished. Available at: http://www. cs. toronto. edu/~ roweis/lle/publications. html*, 2000.

[12] Lawrence K Saul and Sam T Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155, 2003.

[13] Ameet Talwalkar, Sanjiv Kumar, and Henry Rowley. Large-scale manifold learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[14] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[15] Michael E Tipping and Cambridge Cb Nh. Sparse kernel principal component analysis. 2001.

[16] Laurens JP van der Maaten, Eric O Postma, and H Jaap van den Herik. Dimensionality reduction: A comparative review.