American Journal of Computational Mathematics, 2014, 4, 55-72
Published Online March 2014 in SciRes. http://www.scirp.org/journal/ajcm
http://dx.doi.org/10.4236/jasmi.2014.42006

Scientific Research

An Actual Survey of Dimensionality Reduction

Alireza Sarveniazi

Institut fuer Angewandte Forschung (IAF), Karlsruhe, Germany
Email: alireza.sarveniazi@hs-karlsruhe.de

Received 6 November 2013; revised 6 December 2013; accepted 15 December 2013

Abstract

Dimension reduction is defined as the processes of projecting high-dimensional data to a much lower-dimensional space. Dimension reduction methods variously applied in regression, classification, feature analysis and visualization. In this paper, we review in details the last and most new version of methods that extensively developed in the past decade.

Keywords

Dimensionality Reduction Methods

1. Introduction

Any progresses in efficiently using data processing and storage capacities need control on the number of useful variables. Researchers working in domains as diverse as computer science, astronomy, bio-informatics, remote sensing, economics, face recognition are always challenged with the reduction of the number of data-variables. The original dimensionality of the data is the number of variables that are measured on each observation. Especially when signals, processes, images or physical fields are sampled, high-dimensional representations are generated. High-dimensional data-sets present many mathematical challenges as well as some opportunities, and are bound to give rise to new theoretical developments [1].

In many cases, these representations are redundant and the varaibles are correlated, which means that eventually only a small sub-space of the original representation space is populated by the sample and by the underlying process. This is most probably the case, when very narrow process classes are considered. For the purpose of enabling low-dimensional representations with minimal information loss according dimension reduction methods are needed.

Hence, we are reviewing in this paper the most important dimensional reduction methods, including most traditional methods, such as principal component analysis (PCA) and non-linear PCA up to current state-of-art

How to cite this paper: Sarveniazi, A. (2014) An Actual Survey of Dimensionality Reduction. *American Journal of Computational Mathematics*, **4**, 55-72. http://dx.doi.org/10.4236/jasmi.2014.42006

methods published in various areas, such as signal processing and statistical machine learning literature. This actual survey is organized as follows: Section 2 reviews the linear nature of Principal component analysis and its relation with multidimensional scaling (classical scaling) in a comparable way. Section 3 introduces non-linear or Kernel PCA (KPCA) using the kernel-trick. Section 4 is about linear discriminant analysis (LDA), and we give an optimization model of LDA which is a measuring of a power of this method. In Section 5 we summarize another higher-order linear method, namely canonical correlation analysis (CCA)), which finds a low dimensional representation maximizing the correlation and of course its optimization-formulation. Section 6 reviews the relatively new version of PCA, the so-called oriented PCA (OPCA) which is introduced by Kung and Diamantaras [2] as a generalization of PCA. It corresponds to the generalized eigenvalue decomposition of a pair of covariance matrices, but PCA corresponds to the eigenvalue decomposition of only a single covariance matrix. Section 7 introducs principal curves and includes a characterization of these curves with an optimization problem which tell us when a given curve can be a principal curve. Section 8 gives a very compact summary about non-linear dimensional-reduction methods using neural networks which include the simplest neural network which has only three layers:

1) Input Layer
2) Hidden Layer (bottleneck)
3) Output Layer

and an auto-associative neural network with five layers:

1) Input Layer
2) Hidden Layer
3) Bottleneck
4) Hidden Layer
5) Output Layer

A very nice optimizing formulation is also given. In Section 9, we review the Nystroem method which is a very useful and well known method using the numerical solution of an integral equation. In Section 10, we look the multidimensional scaling (MDS) from a modern and more exact consideration view of point, specially a defined objective stress function arises in this method. Section 11 summarizes locally linear embedding (LLE) method which address the problem of nonlinear dimensionality reduction by computing low-dimensional neighborhood preserving embedding of high-dimensional data. Section 12 is about one of the most important dimensional-reduction method namely Graph-based method. Here we will see how the adjacency matrix good works as a powerful tool to obtain a small space which is in fact the eigen-space of this matrix. Section 13 gives a summary on Isomap and the most important references about Dijstra algorithm and Floyd's algorithm are given. Section 14 is a review of Hessian eigenmaps method, a most important method in the so called manifold embedding. This section needs more mathematical backgrounds. Section 15 reviews most new developed methods such as

- vector quantization
- genetic and evolutionary algorithms
- regression

We have to emphasize here the all of given references in the body of survey are used and they are the most important references or original references for the related subject. To obtain more mathematical outline and sensation, we give an appendix about the most important backgrounds on the fractal and topological dimension definitions which are also important to understand the notion of intrinsic dimension.

2. Principal Component Analysis (PCA)

Principal component Analysis (PCA) [3] [4] [5]-[8] is a linear method that it performs dimensionality reduction by embedding the data into a linear subspace of lower dimensional. PCA is the most popular unsupervised linear method. The result of PCA is a lower dimensional representation from the original data that describes as much of the variance in the data as possible. This can be reached by finding a linear basis (possibly orthogonal) of reduced dimensionality for the data, in which the amount of variance in the data is maximal.

In the mathematical language, PCA attempts to find a linear mapping P that maximizes the cost function $tr\left(P^T AP\right)$, where A is the sample covariance matrix of the zero-mean data. Another words PCA maximizes $P^T AP$ with respect to P under the constraint the norm of each column v of P is 1, *i.e.*, $\|v\|^2 = 1$. In fact

PCA solves the eigenvalue problem:

$$AP = \lambda P \quad \text{or} \quad Av = \lambda v \tag{1.1}$$

Why the above optimization Problem is equivalent to the eigenvalue problem (1.1)? consider the convex form $v^T Av + \lambda\left(1 - v^T v\right)$, it is a straightforward calculation that the maximum happens when $Av = \lambda v$.

It is interesting to see that in fact PCA is identical to the multidimensional scaling (classical scaling) [9].

For the given data $\{x_i\}_{i=1}^N$ let $D = \left[d_{ij}\right]$ be the pairwise Euclidean matrix whose entries d_{ij} represent the Euclidean distance between the high-dimensional data points x_i and x_i. multidimensional scaling finds the linear mapping P such that maximizes the cost function:

$$\psi(Y) := \sum_{i,j}\left(d_{ij}^2 - \left\|y_i - y_j\right\|^2\right), \tag{1.2}$$

in which $\left\|y_i - y_j\right\|$ is the Euclidean distance between the low-dimensional data points y_i and y_j, y_i is restricted to be $x_i A$, with $\left\|v_j\right\|^2 = 1$ for all column vector v_j of P. It can be shown [10] [11] that the minimum of the cost function $\psi(Y)$ is given by the eigen-decomposition of the Gram matrix $G = XX^T$ where $X = \left[x_i\right]$. Actually we can obtain the Gram matrix by double-centering the pairwise squared Euclidean distance matrix, *i.e.*, by computing:

$$g_{ij} = -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n}\sum_l d_{il}^2 - \frac{1}{n}\sum_l d_{jl}^2 + \frac{1}{n^2}\sum_{l,m} d_{lm}^2\right). \tag{1.3}$$

Now consider the multiplication of principal eigenvectors of the double-centered squared Euclidean distance matrix (*i.e.*, the principal eigenvectors of the Gram matrix) with the square-root of their corresponding eigenvalues, this gives us exactly the minimum of the cost function in Equation (1.2).

It is well known that the eigenvectors u_i and v_i of the matrices $X^T X$ and XX^T are related through $\sqrt{\lambda_i} v_i = X u_i$ [12], it turns out that the similarity of classical scaling to PCA . The connection between PCA and classical scaling is described in more detail in, e.g., [11] [13]. PCA may also be viewed upon as a latent variable model called probabilistic PCA [14]. This model uses a Gaussian prior over the latent space, and a linear-Gaussian noise model.

The probabilistic formulation of PCA leads to an EM-algorithm that may be computationally more efficient for very high-dimensional data. By using Gaussian processes, probabilistic PCA may also be extended to learn nonlinear mappings between the high-dimensional and the low-dimensional space [15]. Another extension of PCA also includes minor components (*i.e.*, the eigenvectors corresponding to the smallest eigenvalues) in the linear mapping, as minor components may be of relevance in classification settings [16]. PCA and classical scaling have been successfully applied in a large number of domains such as face recognition [17], coin classification [18], and seismic series analysis [19].

PCA and classical scaling suffer from two main drawbacks. First, in PCA, the size of the covariance matrix is proportional to the dimensionality of the data-points. As a result, the computation of the eigenvectors might be infeasible for very high-dimensional data. In data-sets in which $n < D$, this drawback may be overcome by performing classical scaling instead of PCA, because the classical scaling scales with the number of data-points instead of with the number of dimensions in the data. Alternatively, iterative techniques such as Simple PCA [20] or probabilistic PCA [14] may be employed. Second, the cost function in Equation (1.2) reveals that PCA and classical scaling focus mainly on retaining large pairwise distances d_{ij}^2, instead of focusing on retaining the small pairwise distances, which is much more important.

3. Non-Linear PCA

Non-linear or Kernel PCA (KPCA) is in fact the reconstruction from linear PCA in a high-dimensional space that is constructed using a given kernel function [21]. Recently , such reconstruction from linear techniques using the kernel-trick has led to the proposal of successful techniques such as kernel ridge regression and Support Vector Machines [22]. Kernel PCA computes the principal eigenvectors of the kernel matrix, rather than those of the covariance matrix. The reconstruction from PCA in kernel space is straightforward, since a

kernel matrix is similar to the inner product of the data-points in the high-dimensional space that is constructed using the kernel function. The application of PCA in the kernel space provides Kernel PCA the property of constructing nonlinear mappings.

Kernel PCA computes the kernel matrix $K=\left[k_{ij}\right]$ of the data-points x_i. The entries in the kernel matrix are defined by

$$k_{ij} := \kappa\left(x_i, x_j\right) \tag{1.4}$$

where κ is a kernel function [22], which may be any function that gives rise to a positive-semi-definite kernel *K*. Subsequently, the kernel matrix $\bar{K}$ is double-centered using the following modification of the entries

$$\bar{k}_{ij} = -\frac{1}{2}\left(k_{ij} - \frac{1}{n}\sum_l k_{il} - \frac{1}{n}\sum_l k_{jl} + \frac{1}{n^2}\sum_{l,m} k_{lm}\right). \tag{1.5}$$

The centering operation corresponds to subtracting the mean of the features in traditional PCA: it subtracts the mean of the data in the feature space defined by the kernel function κ. Hence, the data in the features space defined by the kernel function is zero-mean. Subsequently, the principal d eigenvectors v_i of the centered kernel matrix are computed. The eigenvectors of the covariance matrix a_i (in the feature space constructed by κ) can now be computed, since they are related to the eigenvectors of the kernel matrix v_i (see, e.g., [12]) through

$$a_i = \frac{1}{\sqrt{\lambda_i}} v_i \tag{1.6}$$

In order to obtain the low-dimensional data representation, the data is projected onto the eigenvectors of the covariance matrix $\left(a_i\right)$. The result of the projection (*i.e.*, the low-dimensional data representation $Y=\left(y_i\right)$ is given by:

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

where $a_l^{(j)}$ indicates the j^{th} value in the vector a_l and κ is the kernel function that was also used in the computation of the kernel matrix. Since Kernel PCA is a kernel-based method, the mapping performed by Kernel PCA relies on the choice of the kernel function κ. Possible choices for the kernel function include the linear kernel (making Kernel PCA equal to traditional PCA), the polynomial kernel, and the Gaussian kernel that is given in [12]. Notice that when the linear kernel is employed, the kernel matrix K is equal to the Gram matrix, and the procedure described above is identical to classical scaling (previous section).

An important weakness of Kernel PCA is that the size of the kernel matrix is proportional to the square of the number of instances in the data-set. An approach to resolve this weakness is proposed in [23] [24]. Also, Kernel PCA mainly focuses on retaining large pairwise distances (even though these are now measured in feature space).

Kernel PCA has been successfully applied to, e.g., face recognition [25], speech recognition [26], and novelty detection [25]. Like Kernel PCA, the Gaussian Process Latent Variable Model (GPLVM) also uses kernel functions to construct non-linear variants of (probabilistic) PCA [15]. However, the GPLVM is not simply the probabilistic counterpart of Kernel PCA: in the GPLVM, the kernel function is defined over the low-dimensional latent space, whereas in Kernel PCA, the kernel function is defined over the high-dimensional data space.

4. Linear Discriminant Analysis (LDA)

The main Reference here is [27] see also [28]. The LDA is a method to find a linear transformation that maximizes class separability in the reduced dimensional space. The criterion in LDA is in fact to maximize between class scatter and minimize within-class scatter. The scatters are measured by using scatter matrices. Let we have r class C_i each including n_i points $x_j^i \in \mathbb{R}^l$ and set $X=\left[\tilde{C}_1, \cdots, \tilde{C}_r\right] \in \mathbb{R}^{l\times n}$, where $\tilde{C}_i = \left[x_1^i, \cdots, x_{n_i}^i\right]$ and $n=\sum_{i=1}^r n_i$. Let $\overline{x^i} = \frac{1}{n_i}\sum_{j=1}^{n_i} x_j^i$ and $\bar{x} = \frac{1}{n}\sum_{i=1}^n x^i$.

Now we define three scatter matrices:

The between-class scatter matrix $S_b := \sum_{i=1}^{r} n_i \left(\overline{x^i} - \overline{x}\right)\left(\overline{x^i} - \overline{x}\right)^T$,

The within-class scatter matrix $S_w := \sum_{i=1}^{r}\sum_{j=1}^{n_i}\left(x_j^i - \overline{x^i}\right)\left(x_j^i - \overline{x^i}\right)^T$,

The total scatter matrix $S_t := \sum_{i=1}^{r}\sum_{j=1}^{n_i}\left(x_j^i - \overline{x}\right)\left(x_j^i - \overline{x}\right)^T$. Actually LDA is a method for the following optimization problem:

$$\arg\max_{U\in\mathbb{R}^{l\times m}} \frac{\left|U^T S_b U\right|}{\left|U^T S_w U\right|}$$

Hence in this way the dimension is reduced from l to m by a linear transformation $\overline{U}$ which is the solution of above optimization problem. Although we know from Fukunaga (1990), (see [27] and [29]) that the eigenvectors corresponding to the $r-1$ largest eigenvalues of

$$S_b u = \lambda S_w u$$

form the columns of U as above for LDA.

5. Canonical Correlation Analysis (CCA)

CCA is an old method back to the works of Hotelling 1936 [30], recently Sun *et al.* [31] used CCA as an unsupervised feature fusion method for two feature sets describing the same data objects. CCA finds projective directions which maximize the correlation between the feature vectors of the two feature sets.

Let $X = x_{i_{i=1}}^{n}$ and $Y = y_{i_{i=1}}^{n}$ be two data set of n points in $\mathbb{R}^p$ and $\mathbb{R}^q$ respectively, associate with them we have two matrices:

$$A_X = \left[x_1 - \overline{x}, \cdots, x_n - \overline{x}\right] \in \mathbb{R}^{p\times n}, \quad A_Y = \left[y_1 - \overline{y}, \cdots, y_n - \overline{y}\right] \in \mathbb{R}^{q\times n}$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ are the means of x_i and y_i s, respectively.

Actually CCA is a method for the following optimization problem:

$$\arg\max_{U_X, U_Y} \frac{U_X^T A_X A_Y^T U_Y}{\sqrt{U_X^T A_X A_X^T U_X}\sqrt{U_Y^T A_Y A_Y^T U_Y}}$$

which can be modified as

$$\arg\max_{U_X, U_Y} U_X^T A_X A_Y^T U_Y, \quad U_X^T A_X A_X^T U_X = 1, \quad U_Y^T A_Y A_Y^T U_Y = 1$$

Assume the pair of projective directions $\left(U_X^*, U_y^*\right)$ be the solution of above optimization problem, we can find another pair of projective directions by solving

$$\arg\max_{U_X, U_Y} U_X^T A_X A_Y^T U_Y, \quad U_X^{*T} A_X A_X^T U_X^* = U_Y^{*T} A_Y A_Y^T U_Y^* = 0, \quad U_X^T A_X A_X^T U_X = U_Y^T A_Y A_Y^T U_Y = 1$$

repeating the above process $m-1$ times we obtain a m-dimensional specs of linear combination of these vector-solutions.

In fact we can obtain this m-dimensional space with solving of the paired eigenvalue problem:

$$A_X A_Y^T \left(A_Y A_Y^T\right)^{-1} U_X = \lambda A_X A_X^T U_X, \quad A_Y A_X^T \left(A_X A_X^T\right)^{-1} U_Y = \lambda A_Y A_Y^T U_y$$

and the eigenvectors $\left(U_X^{(i)}, U_Y^{(i)}\right), i = 1, \cdots, m$ corresponding to the m largest eigenvalues are the pairs of projective directions for CCA see [31]. Hence

$$\left\{U_X^{(i)T} A_X, i = 1, \cdots, m\right\} \text{ and } \left\{U_Y^{(i)T} A_Y, i = 1, \cdots, m\right\}$$

compose the feature sets extracted from A_X and A_Y by CCA. It turns out that the number m is determined as the number of nonzero eigenvalue.

6. Oriented PCA (OPCA)

Oriented PCA is introduced by Kung and Diamantaras [2] as a generalization of PCA. It corresponds to the generalized eigenvalue decomposition of a pair of covariance matrices in the same way that PCA corresponds to the eigenvalue decomposition of a single covariance matrix. For the given pair of vectors u and v the objective function maximized by OPCA is given as follows:

$$\arg\max_{w} \frac{E\left(w^T u\right)^2}{E\left(w^T v\right)^2} = \frac{w^T A_u w}{w^T A_v w}$$

where $A_u := E\left(uu^T\right)$, $A_v := E\left(vv^T\right)$. A solution w_1^* of above optimization problem is called Principal oriented component and it is the generalized eigenvector of matrix pair $\left[A_u, A_v\right]$ corresponding to maximum generalized eigenvalue λ_1. Since A_u and A_v are symmetric all the generalized eigenvalues are real and thus they can be arranged in decreasing order, as with ordinary PCA. Hence we will obtain the rest generalized eigenvectors $w_2^*, w_3^*, \cdots, w_m^*$, as second , third, ···, m^{th} oriented principal components. All of these solutions are the solutions under the orthogonality constraint:

$$w_i^{*T} A_u w_j = w_i^{*T} A_v w_j^* = 0, \text{ for } i \neq j$$

7. Principal Curves and Surfaces

By the definition, principal curves are smooth curves that pass through the middle of multidimensional data sets, see [32]-[34] as main references and also [35] and [36].

Given the n-dimensional random vector $x = \left(x_1, \cdots, x_n\right) \in \mathbb{R}^n$ with probability density function $p(x)$. Let $\boldsymbol{f} : \mathbb{R} \to \mathbb{R}^n$ be the given smooth curve which can be parametrized by a real value θ (actually we can choose $\theta \in [0,1]$). Hence we have $\boldsymbol{f}(\theta) = \left(f_1(\theta), \cdots, f_n(\theta)\right)$.

We can associate to the curve $\boldsymbol{f}$ the projection index $\Theta_f : \mathbb{R}^n \to \mathbb{R}$ geometrically as the value of θ corresponding to the point on the curve $\boldsymbol{f}$ that under Euclidean metric is the closet point to x.

We say $\boldsymbol{f}$ is self-consistent if each point $\boldsymbol{f}(\theta)$ is the mean of all points in the support of density function p that are projected on θ, *i.e.*,

$$E\left[x \middle| \Theta_f(x) = \theta\right] = \boldsymbol{f}(\theta).$$

It is shown in [32] that the set of principal curves do not intersect themselves and they are self-consistent. Most important fact about principal curves which proved in [32] is a characterization of these curves with an optimization Problem:

Theorem 1 *A curve* $\boldsymbol{f}$ *is a principal curve* (*associate with the data set* $\left\{x^{(i)}\right\}_{i=1}^N$) *iff it solves fallowing optimization problem*

$$\min_{\boldsymbol{f}, \boldsymbol{\Theta}_f} \sum_{i=1}^{N} \left\| x^{(i)} - \boldsymbol{f}\left(\boldsymbol{\Theta}_f\left(x^{(i)}\right)\right)\right\|^2, \tag{1.7}$$

Of course to solve (or even estimate) minimization (0.7) is a complex problem, to estimate $\boldsymbol{f}$ and θ in [32] an iterative algorithm has given. It started with $\boldsymbol{f}(\theta) = E(x) + \theta, \boldsymbol{u}_1$, where $\boldsymbol{u}_1$ is the first eigenvector of covariance matrix of x and $\theta = \Theta_f(x)$. Then it iterates the two steps:

- For a fixed θ, minimize $\left\|x - \boldsymbol{f}(\theta)\right\|$ by setting

$$\boldsymbol{f}_j(\theta) = E\left[x_j \middle| \Theta_f(x) = \theta\right] \text{ for each } j$$

- Fix $\boldsymbol{f}$ and set $\Theta_f(x)=\theta$ for each x until the change in $\|x-\boldsymbol{f}(\theta)\|$ is less than a threshold.

One can find in [37] another formulation of the principal curves, along with a generalized EM algorithm for its estimation under Gaussian pdf $p(x)$. Unfortunately except for a few special cases, it is an open problem for what type of distributions do principal curves exist, how many principal curves there exist and which properties the have see [36]. in recent years the concept of principal curves has been extended to higher dimensional principal surfaces, but of course the estimation algorithms are not smooth as the curves.

8. Non-Linear Methods Using Neural Networks

Given Input variables $\{x_1, x_2, \cdots, x_N\}$, neural networks getting this input and gives output variables $\{y_1, y_2, \cdots, y_m\}$ with

$$y_j = y_j(x, w),\ j = 1, \cdots, m$$

where the weights w are determined by training the neural network using a set of given instances and a cost function see [38]. Over the last two decades there are several developments based on a ring architectures and learning algorithms of dimensional reduction techniques could be implemented using neural networks, see [35] [36] [38]-[40]. Consider the simplest neural network which has only three layers:

1) Input Layer
2) Hidden Layer (bottleneck)
3) Output Layer

there are two steps here:

- In order to obtain the data at node k of the hidden layer, we have to consider any inputs x_i in combination with their associated weight's w_{ik} along with a threshold term (or called bias in some references) ρ_k, Now they are ready passing through to the corresponding activation φ_k, hence we are building up the expression $\varphi_k\left(\rho_k + \sum_i w_{ik} x_i\right)$.
- Here we have to repeat step (1) with changing original data x_i with new one namely $\varphi_k\left(\rho_k + \sum_i w_{ik} x_i\right)$, of course according the threshold ρ_j and possibly new output function φ_{out}. Hence we have:

$$y_j = \varphi_{out}\left(\rho_j + \sum_k w_{kj}\varphi_k\left(\rho_k + \sum_i w_{ik} x_i\right)\right).$$

We observe that the first part of network reduces the input data into the lower-dimensional space just as same as a linear PCA, but the second part decodes the reduced data into the original domain [36] [35]. Note that only by adding two more hidden layers with nonlinear activation functions, one between the input and the bottleneck, the other between the bottleneck and the output layer, the PCA network can be generalized to obtain non-linear PCA. One can extend this idea from the feed-forward neural implementation of PCA extending to include non-linear activation function in the hidden layers [41],. In this framework, the non-linear PCA network can be considered of as an auto-associative neural network with five layers:

1) Input Layer
2) Hidden Layer
3) Bottleneck
4) Hidden Layer
5) Output Layer

If $\boldsymbol{\Theta}_f : \mathbb{R}^n \to \mathbb{R}^l$ be the function modeled by layers (1), (2) and (3), and $\boldsymbol{f} : \mathbb{R}^l \to \mathbb{R}^n$ be the modeled function by layers (3), (4) and (5), in [35] have been shown that weights of the non-linear PCA network are determined such that the following optimization Problem solved:

$$\min_{\boldsymbol{f}, \boldsymbol{\Theta}_f} \sum_{i=1}^{N} \left\| x_i - \boldsymbol{f}\left(\boldsymbol{\Theta}_f(x_i)\right)\right\|^2,$$

As we have seen in the last section the function $\boldsymbol{f}$ must be Principal curve(surface). In the thesis [42], one

can find comparison between PCA, Vector Quantization and five layer neural networks, for reducing the dimension of images.

9. Nystroem Method

The Nystroem Method is a well known technique for finding numerical approximations of generic integral equation and specially to eigenfunction problems of the following form:

$$\int_a^b W(x,y)\, f(y)\mathrm{d}y = \lambda f(x)$$

We can divide the interval $[a,b]$ into n points $\theta_1,\cdots,\theta_n$ where

$$\theta_{i+1} = \theta_i + i\Delta \quad \text{and} \quad \Delta = \frac{b-a}{n},\ \theta_1 = a,\ \theta_n = b.$$

Now consider the simple quadrature rule:

$$\frac{b-a}{n}\sum_{j=1}^{n} W(x,\theta_j)\,\tilde{f}(\theta_j) = \lambda \tilde{f}(x) \tag{1.8}$$

which $\tilde{f}$ approximates f, for $x=\theta_i$ we obtain a system of n equations:

$$\frac{b-a}{n}\sum_{j=1}^{n} W(\theta_i,\theta_j)\,\tilde{f}(\theta_j) = \lambda_i \tilde{f}(\theta_i), i = 1,\cdots,n$$

without loss of generality we can shift interval $[a,b]$ to unit interval $[0,1]$ and change the above system of equations to the following eigenvalue problem:

$$\tilde{A}(f) = nD\tilde{\boldsymbol{f}} \tag{1.9}$$

where $A = \left[W(\theta_i,\theta_j)\right]$, $\tilde{\boldsymbol{f}} = \left(\tilde{f}_1,\cdots,\tilde{f}_n\right)$ and $D = \mathrm{diag}(\lambda_1,\cdots,\lambda_n)$, substituting back into 0.8 yields the Nstroem extension for each $\tilde{f}_i$:

$$\tilde{f}_i(x) = \frac{1}{n\lambda_i}\sum_{j=1}^{n} W(x,\theta_j)\,\tilde{f}_i(\theta_j)$$

We can extend above arguments for $x \in \mathbb{R}^n$ and $n>1$, see [42].

Motivated from 0.9 our main question is if A be a given $n\times n$ real symmetric matrix with small rank r, *i.e.*, $r \ll n$, can we approximate the eigenvectors and eigenvalues of A using those of a small sub-matrix of A?

Nystroem method gives a positive answer to this question. Actually we can assume that the r randomly chosen samples come first and the $n-r$ samples come next. Hence the matrix A in 0.9 can have following form:

$$A = \begin{pmatrix} E & B \\ B^T & C \end{pmatrix}$$

Hence E represents the sub-block of weights among the random samples, B contains the weights from the random samples to the rest of samples and C contains the weights between all of remaining samples. Since $r \ll n$, C must be a large matrix. Let $\bar{U}$ denote the approximate eigenvectors of A, the Nystroem extension method gives:

$$\bar{U} = \begin{pmatrix} U \\ B^T U D^{-1} \end{pmatrix}$$

where U and D are eigenvectors and diagonal matrix associate with E, *i.e.*, $E = U^T D U$. Now the associated approximation of A, which we denote it with $\tilde{A}$, then we have:

$$\tilde{A} = \bar{U}D\bar{U}^T = \begin{pmatrix} U \\ B^T U D^{-1} \end{pmatrix} D\left(U^T \quad D^{-1}U^T B\right) = \begin{pmatrix} UDU^T & B \\ B^T & B^T E^{-1} B \end{pmatrix} = \begin{pmatrix} E & B \\ B^T & B^T E^{-1} B \end{pmatrix} = \begin{pmatrix} E \\ B^T \end{pmatrix} E^{-1}\begin{pmatrix} A & B \end{pmatrix}$$

The last equation is called “bottleneck” form. There is a very interesting application of this form in Spectral

Grouping which it was possible to construct the exact eigen-decomposition of A using the eigen-decomposition of smaller matrix rank r. Also Fowlkes et al have given an application of the Nystroem method t NCut Problem, see [43].

10. Multidimensional Scaling (MDS)

Given N point $X =: \{x_1, x_2, \cdots, x_N\} \subset \mathbb{R}^n$ and build up the distance matrix $\Delta = [d_{ij}]$ where $d_{ij} = \|x_i - x_j\|$, (or in general $d_{ij} = d(x_i, x_j)$ for some metric which defined d) MDS (better to say a m-dimensional MDS) is a technique that produces output points $\{y_1, y_2, \cdots, y_N\} \subset \mathbb{R}^m$ such that the distances d_{ij} are as close as possible to a function f of the corresponding proximity's $f(d_{ij})$. From [36], whether this function ϕ is linear or non-linear, MDS is called either metric or non-metric. Define an objective stress function

MDS-PROCEDURE:

- Define an objective stress function and stress factor α, that it depends on $\sum_{i,j} f(d_{ij})^2$ or on $\sum_{i,j} d_{ij}^2$

$$\Phi_S^f(\Delta, X, f) := \sqrt{\frac{\sum_{i,j}\left(f(d_{ij}) - d_{ij}\right)^2}{\alpha}} \tag{1.10}$$

- Now if for a given X as above, find f^* that minimize 0.10, *i.e.*,

$$\Phi_S(\Delta, X, f^*) = \min_f \Phi_S^f(\Delta, X, f)$$

- Determine the optimal data set $\tilde{X}$ by

$$\Phi_S(\Delta, \tilde{X}, f^*) = \min_X \Phi_S^f(\Delta, X, f^*)$$

If we use Euclidean distance and take $f = id$ in Equation (1.10) the produced output data set should be coincide to the Principal component of cov(X)(without re-scaling to correlation), hence in this special case MDS and PCA are coincide (see [44]) There exist an alternative method to MDS, namely Fast Map see[45] [46].

11. Locally Linear Embedding (LLE)

Locally linear embedding is an approach which address the problem of nonlinear dimensionality reduction by computing low-dimensional neighborhood preserving embedding of high-dimensional data. A data set of dimensionality n, which is assumed to lie on or near a smooth nonlinear manifold of dimensionality $m \ll n$, is mapped into a single **global** coordinate system of lower-dimensionality m. The global nonlinear structure is recovered by locally linear fits.As usual given a Data set of N points on a n-dimensional points $\{x_i\}_{i=1}^N$ from some underlying manifold. Without loss of generality we can assume each data point and its neighbors lie on are close to a locally linear sub-manifold. By a linear transform, consisting of a translation, rotation and rescaling, the high-dimensional coordinates of each neighborhood can be mapped to global internal coordinates on the manifold. In order to map the high-dimensional data to the single global coordinate system of the manifold such that the relationships between neighboring points are preserved. This proceeds in three steps:

- Identify neighbors of each data point x_i. this can be done by finding the K nearest neighbors, or choosing all points within some fixed radius ε.
- Compute the weights $[w_{ij}]$ that best linearly reconstruct x_i from its neighbors.
- Find the low-dimensional embedding vector y_i which is the best reconstructed by the weights determined in the previous step.

After finding the nearest neighbors in the first step, the second step must compute a local geometry for each locally linear sub-manifold. This geometry is characterized by linear coefficients that reconstruct each data point from its neighbors.

$$\min_w \sum_{i=1}^{n} \left\| x_i - \sum_{j=1}^{K} w_{ij} x_{N_{i(j)}} \right\|^2 \tag{1.11}$$

where $N_i(j)$ is the index of the j^{th} neighbor of the point. It then selects code vectors so as to preserve the

reconstruction weights by solving

$$\min_{Y} \sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{K} w_{ij} y_{N_{i(j)}} \right\|^2 \tag{1.12}$$

This objective can be restate as

$$\min_{Y} Tra\left(Y^T Y L\right) \tag{1.13}$$

where $L = (I - W)T(I - w)$.

The solution for Y can have an arbitrary origin and orientation. In order to make the problem well-posed, those two degree of freedom must be removed. Requiring the coordinates to be centered on origin ($\sum_{i=1} y_i = 0$, and constructing the embedding vectors to have unit covariance $\left(Y^T Y = I\right)$, removes the first and second degrees of freedom respectively. The cost function can be optimized initially by the second of those two constraints. Under this constraint, the cost is minimized when the column of Y^T (rows of Y) are the eigenvectors with the lowest eigenvalues of L. Discarding the eigenvector associated with eigenvalue 0 satisfies the first constraint.

12. Graph-Based Dimensionality Reduction

As before given a data set X include N points in $\mathbb{R}^n$, *i.e.*, $X = \{x_1, x_2, \cdots, x_N\}$, we associate to X a weighted undirected graph with N vertices and use the Laplacian matrix which defined see [47]. In order to define an undirected graph we need define a pair $(V;E)$ of sets, V the set of vertices and E the set of edges. we follows here the method introduced in [48].

we say $v_i \in V$ if $x_i \in X$ and $(v_i, v_j) \in E$ iff x_i and x_j are **close**. But what it means to be **close**? there are two variations define it:

- ε-neighborhoods, which ε is a positive small real number.
  x_i and x_j are **close** iff $\|x_i - x_j\|^2 \le \varepsilon$, where the norm is as usual the Euclidean norm in $\mathbb{R}^n$.
- K nearest neighbors. Here K is a natural number.
  x_i and x_j are **close** iff x_i is among K nearest neighbors of x_j or x_j is among K near-es neighbors of x_i. that means this relation is a symmetric relation.

To associate the weights to edges, as well, there is two variations:

- Heat kernel, which γ is a real number.

$$w_{ij} = \begin{cases} \exp\left(-\dfrac{\|x_i - x_j\|^2}{\gamma}\right), & \text{if } x_i \text{ and } x_j \text{ are close} \\ 0, & \text{otherwise} \end{cases}$$

- Simple adjacency with parameter $\gamma = \infty$.

$$w_{ij} = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ are close} \\ 0, & \text{otherwise} \end{cases}$$

We assume our graph, defined as above, is connected, otherwise proceed following for each connected component. Set $d_{ii} = \sum_{j=1}^{N} w_{ij}$ and $d_{ij} = 0$ if $i \ne j$, $D = [d_{ij}]$, $W = [w_{ij}]$. $L = D - W$ is the Laplacian matrix of the
graph, which is a symmetric, positive sewmi-definite matrix, so can be thought of as an operator on the space of real functions defined on the vertices set V of Graph.

Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$L\boldsymbol{f} = \lambda D\boldsymbol{f}$$

Let $\boldsymbol{f}_0, \boldsymbol{f}_1, \cdots, \boldsymbol{f}_{N-1}$ be the solutions of the above eigenvalue problem, ordered acording to their eigenvalues,

$$
\begin{aligned}
L\mathbf{f}_0 &= \lambda D\mathbf{f}_0 \\
L\mathbf{f}_1 &= \lambda D\mathbf{f}_1 \\
&\vdots \\
L\mathbf{f}_{N-1} &= \lambda D\mathbf{f}_{N-1} \\
0 = \lambda_0 &\le \lambda_1 \le \cdots \le \lambda_{N-1}
\end{aligned}
$$

We leave out the eigenvector (trivial eigenfuntion) corresponding to eigenvalue o, which is a vector with all component equal to 1 and use next m eigenvectors for embedding in m-dimensional Euclidean space:

$$x_i \mapsto \left(\mathbf{f}_1^{(i)}, \cdots, \mathbf{f}_m^{(i)}\right)$$

which $\mathbf{f}^{(i)}$ means i^{th} component of the vector $\mathbf{f}$. This called the Laplacian Eigenmap embedding by Belkin and Niogi, see [48].

13. Isomap

Like LLE the Isomap algorithm proceeds in three steps:

- Find the neighbors of each data point in high-dimensional data space.
- Compute the geodesic pairwise distances between all points.
- Embed the data via MDS so as preserve those distances

Again like LLE, the first, the first step can be performed by identifying the K-nearest neighbors, or by choosing all points within some fixed radius, ε. These neighborhood relations are represented by graph G in which each data point is connected to its nearest neighbors, with edges of weights $d_X(i, j)$ between neighbors.

The geodesic distances $d_X(i, j)$ between all pairs of points on the manifold M are then estimated in the second step. Isomap approximates $d_M(i, j)$ as the shortest path distance $d_G(i, j)$ in the graph G. This can be done in different ways including Dijstra algorithm [49] and Floyd's algorithm [50]

14. Hessian Eigenmaps Method

High dimensional data sets arise in many real-world applications. These data points may lie approximately on a low dimensional manifold embedded in a high dimensional space. Dimensionality reduction (or as in this case, called manifold learning) is to recover a set of low-dimensional parametric representations for the high-dimensional data points, which may be used for further processing of the data. More precisely consider a d-dimensional parametrized manifold $\mathcal{M}$ embedded in $\mathbb{R}^n$ where $(d < n)$ characterized by a nonlinear map $\psi : \mathcal{X} \subset \mathbb{R}^d \mapsto \mathbb{R}^n$, where $\mathcal{X}$ is a compact and connected subset of $\mathbb{R}^d$. Here $\mathbb{R}^n$ is the high-dimensional data space with $\mathcal{M} = \psi(\mathcal{X})$ being the manifold containing data points and $\mathbb{R}^d$ is the low-dimensional parameter space. Suppose we have a set of data points $x_1, x_2, \cdots, x_N$ sampled from the manifold $\mathcal{M}$ with

$$x_i = \psi(y_i), i = 1, 2, \cdots, N,$$

for some $y_i \in \mathcal{X}$. Then the dimensionality reduction problem is to recover the parameter points y_i s and the map ψ from y_i s.

Of course, this problem is not well defined for a general nonlinear map ψ. However, as is shown by Donoho and Grimes in the derivation of the Hessian Eigenmaps method [51], if ψ is a local isometric map, then $y = \psi^{-1}(x)$ is uniquely determined up to a rigid motion and hence captures the geometric structure of the data set.

Given that the map ψ defined as above, is a local isometric embedding, the map $\phi = \psi^{-1} : \mathcal{M} \subset \mathbb{R}^n \mapsto \mathbb{R}^d$ provides a locally) isometric coordinate system for $\mathcal{M}$. Each component of ϕ is a function defined on $\mathcal{M}$ that provides one coordinate. The main idea of the Hessian Eigenmaps is to introduce a Hessian operator and a functional called the $\mathcal{H}$-functional defined for functions on $\mathcal{M}$, for which the null space consists of the d coordinate functions and the constant function. Let $f : \mathcal{M} \mapsto \mathbb{R}$ be a function defined on $\mathcal{M}$ and let x_0 be an interior point of manifold $\mathcal{M}$. We can define a function $g : \mathcal{X} \mapsto \mathbb{R}$ as $g(y) = f(\psi(y))$, where

$\mathcal{X}=\phi(\mathcal{M})\subset\mathbb{R}^d$ and $y=[y_1,y_2,\cdots,y_d]^{\mathrm{T}}\in\mathcal{X}]\cdot h$ is called a pullback of f to $\mathcal{X}$. Let $y_0=\phi(x_0)$. We call the Hessian matrix of g at y_0 the Hessian matrix of function f at x_0 in the isometric coordinate and we denote it by $H_f^{iso}(x_0)$. Then $\left(H_f^{iso}f\right)_{i,j}(x_0)=\dfrac{\partial^2 g(y_0)}{\partial y_i\partial y_j}$. From the Hessian matrix, we define a $\mathcal{H}$-functional of f in isometric coordinates, denoted by $\mathcal{H}_f^{iso}$, as

$$\mathcal{H}_f^{iso}=\int_{\mathcal{M}}\left\|H_f^{iso}f(x)\right\|_F^2\,\mathrm{d}x, \tag{1.14}$$

where $\mathrm{d}x$ is a probability measure on $\mathcal{M}$ which has strictly positive density everywhere on the interior of $\mathcal{M}$. It is clear that $\mathcal{H}_f^{iso}$ of the d component functions of ϕ are zero as their pullbacks to $\mathcal{X}$ are linear functions. Indeed, $\mathcal{H}_f^{iso}(\cdot)$ has a $(d+1)$-dimensional null space, consisting of the span of the constant functions and the d component functions of ϕ; see [51] (Corollary 4). The Hessian matrix and the $\mathcal{H}$-functional in isometric coordinates introduced above are unfortunately not computable without knowing the isometric coordinate system ϕ first. To obtain a functional with the same property but independent of the isometric coordinate system ϕ, a Hessian matrix and the $\mathcal{H}$-functional in local tangent coordinate systems are introduced in [51]. Qiang Ye and Weifeng Zhi [52] developed a discrete version of the Hessian Eigenmaps method of Donoho ad Grims.

15. Miscellaneous

15.1. Vector Quantization

The main references for vector quantization are [40] and [53]. In [53] it is introduced a hybrid non-linear dimension reduction method based on combining vector quantization for first clustering the data, after constructing the Voronoi cell clusters, applying PCA on them. In [40] both non-linear method *i.e.*, vector quantization and non-linear PCA (using a five layer neural network) on the image data set have been used. It turns out that the vector quantization achieved much better results than non-linear PCA.

15.2. Genetic and Evolutionary Algorithms

These algorithms introduced in [54] are in fact optimization algorithms based on Darwinian theory of evolution which uses natural selection and genetics to find the optimized solution among members of competing population. There are several references for genetic and evolutionary algorithms [55], see [56] for more detail. An evolutionary algorithm for optimization is different from classical optimization methods in several ways:

- Random Versus Deterministic Operation
- Population Versus Single Best Solution
- Creating New Solutions Through Mutation
- Combining Solutions Through Crossover
- Selecting Solutions Via “Survival of the Fittest”
- Drawbacks of Evolutionary Algorithms

In [55] using genetic and evolutionary and algorithms combine with a k-nearest neighbor classifier to reduce the dimension of feature set. Here Input is population matrices which are in fact random transformation matrices $\{W_{m\times N}\}^{(i)}$, then algorithms will find output $Y_{m\times N}$ so that the k-nearest neighbor classifier using the new features $B_{m\times r}=Y_{m\times N}X_{N\times r}$ classifies the training data most accurately.

15.3. Regression

We can use Regression methods for dimension reduction when we are looking for a variable function $y=f(x_1,\cdots,x_n)$ for a given data set variables $\{x_i\}$. Under assumption that the x_i s are uncorrelated and relevant to expanding the variation in y. Of course in modern data mining applications however such assumptions rarely hold. Hence we need a dimension reduction for such a case. We can list well-known dimension

reduction methods as follows:

- The Wrapper method in machine learning community [57]
- Projection pursuit regression [36] [58]
- Generalized linear models [59] [60]
- Adaptive models [61]
- Neural network models and sliced regression and Principal hessian direction [62]
- Dimension reduction for conditional mean in regression [63]
- Principal manifolds and non-linear dimension reduction [64]
- Sliced regression for dimension reduction [65]
- Canonical correlation [66]

Acknowledgements

Our research has received funding from the (European Union) Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n [314329]. we would like to thank Eu-Commission for the support.

References

[1] Donoho, D.L. (2000) High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. Lecture Delivered at the "Mathematical Challenges of the 21st Century" Conference of the American Math. Society, Los Angeles. http://www-stat.stanford.edu/donoho/Lectures/AMS2000/AMS2000.html

[2] Diamantaras, K.I. and Kung, S.Y. (1996) Principal Component Neural Networks: Theory and Applications. John Wiley, NY.

[3] Person, K. (1901) On Lines and Planes of Closest Fit to System of Points in Space. *Philiosophical Magazine*, **2**, 559-572. http://dx.doi.org/10.1080/14786440109462720

[4] Jenkins, O.C. and Mataric, M.J. (2002) Deriving Acion and Behavior Primitives from Human Motion Data. *International Conference n Robots and Systems*, **3**, 2551-2556.

[5] Jain, A.K. and Dubes, R.C. (1962) Algorithms for Clastering Data. Prentice Hall, Upper Saddle River.

[6] Mardia, K.V., Kent, J.T. and Bibby, J.M. (1995) Multivariate Analysis Probability and Mathematical Statistics. Academic Press, Waltham.

[7] (2002) Francesco Camastra Data Dimensionality Estimation Methods, a Survey INFM-DISI, University of Genova, Genova.

[8] Fukunaga, K. (1982) Intrinsic Dimensionality Extraction, in Classification, Pattern Recognition and Reduction of Dimensionality, Vol. 2 of Handbook of Statistics, North Holland, 347-362.

[9] Torgerson, W.S. (1952) Multidimenmsional Scaling I: Theory and Methode. *Psychometrika*, **17**, 401-419. http://dx.doi.org/10.1007/BF02288916

[10] Teng, L., Li, H., Fu, X., Chen, W. and Shen, I-F. (2005) Dimension Reduction of Microarrey Data Based on Local Tangent Space Aligment. *Proceedings of the 4th IEEE international Conference on Cogenitive Informatics*, 154-159.

[11] Williams, C.K.I. (2002) On a Connection between Kernel PCA and Metric Multidimensional Scaling. *Machine Learning*, **46**, 11-19. http://dx.doi.org/10.1023/A:1012485807823

[12] Chatfield, C. and Collins, A.J. (1980) Introduction to Multivariate Analysis. Chapman and Hill. http://dx.doi.org/10.1007/978-1-4899-3184-9

[13] Platt, J.C. (2005) FastMap, MetricMap, and Landmark MDS are all Nyström algorithms. *Proceddings of the 10th International Workshop on Artificial Intelligence and Statistics*, **15**, 261-268.

[14] Roweis, S.T. (1997) EM Algorithms for PCA and SPCA. *Advances in Neural Information Processing Systems*, **10**, 626-632.

[15] Lawrence, N.D. (2005) Probabilistic Non-Linear Proncipal Component Analysis with Gaussian Process Latent Variable Models. *Journal of Machine Learning Research*, **6**, 1783-1816.

[16] Welling, M., Rosen-Zvi, M. and Hinton, G. (2004) Exponential Family Harmoniums with an Application to Information Retrieval. *Advances in Neural Information Processing Systems*, **17**, 1481-1488.

[17] Turk, M.A. and Pentland, A.P. (1991) Face Recognition Using Eigenfaces. *Proceedings of the Computer Vision and Pattern Recognition* 1991, Maui, 586-591. http://dx.doi.org/10.1109/CVPR.1991.139758

[18] Huber, R., Ramoser, H., Mayer, K., Penz, H. and Rubik, M. (2005) Classification of Coins Using an Eigenspace Ap-

proach. *Pattern Recognition Letters*, **26**, 61-75. http://dx.doi.org/10.1016/j.patrec.2004.09.006

[19] Posadas, A.M., Vidal, F., de Miguel, F., Alguacil, G., Pena, J., Ibanez, J.M. and Morales, J. (1993) Spatialtemporal Analysis of a Seismic Series Using the Principal Components Method. *Journal of Geophysical Research*, **98**, 1923-1932. http://dx.doi.org/10.1029/92JB02297

[20] Partridge, M. and Calvo, R. (1997) Fast Dimensionality Reduction and Simple PCA. *Intelligent Data Analysis*, **2**, 292-298.

[21] Schölkopf, B., Smola, A. and Müller, K.R. (1998) Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, **10**, 1299-1319.

[22] Shawe-Taylor, J. and Christianini, N. (2004) Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge.

[23] Tipping, M.E. (2000) Sparse Kernel Principal Component Analysis. *Advances in Neural Information Processing Systems*, **13**, 633-639.

[24] Kim, K.I., Jung, K. and Kim, H.J. (2002) Face Recognition Using Kernel Principal Component Analysis. *IEEE Signal Processing Letters*, **9**, 40-42. http://dx.doi.org/10.1109/97.991133

[25] Hoffmann. H. (2007) Kernel PCA for Novelty Detection. *Pattern Recognition*, **40**, 863-874. http://dx.doi.org/10.1016/j.patcog.2006.07.009

[26] Lima, A., Zen, H. Nankaku, Y. Miyajima, C. Tokuda, K. and Kitamura. T. (2004) On the Use of Kernel PCA for Feature Extraction in Speech Recognition. *IEICE Transactions on Information Systems*, **E87-D**, 2802-2811.

[27] Duda, R.O., Hart, P.E. and Stork, D.G. (2001) Pattern Classification, Wiley Interscience, New York.

[28] Shin, Y.J. and Park, C.H. (2011) Analysis of Correlation Based Dimension Reduction Methods. *International Journal of Applied Mathematics and Computer Science*, **21**, 549-558.

[29] Fukunaga, K. (1990) Introduction to Statistical Pattern Recognition. 2nd Edition, Academic Press, San Diego.

[30] Hotelling, H. (1936) Relations between Two Sets of Vertices. *Biometrika*, **28**, 321-377.

[31] Sun, Q., Zeng, S., Liu, Y., Heng, P. and Xia, D. (2005) A New Methode of Feature Fusion and Its Application in Image Recognition. *Pattern Recognition*, **38**, 2437-2448. http://dx.doi.org/10.1016/j.patcog.2004.12.013

[32] Hastie, T. and Stuezle, W. (1989) Principal Curves. *Journal of the American Statistical Association*, **84**, 502-516.

[33] Kegl, B. and Linder, T. (2000) Learning and Design of Principal Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 281-297.

[34] Ozertem, U. and Erdogmus, D. (2011) Locally Defined Principal Curves and Surfaces. *Journal of Machine Learning Research*, **12**, 1249-1286.

[35] Malthouse, E. (1996) Some Theoretical Results on Nonlinear Principal Component Analysis. citeseer.nj.net.com/malthouse96some.html

[36] Carreira-Perpinan, M.A. (1997) A Review of Dimension Reduction Tecniques. Technical Report CS-96-09. Department of Computer Science, University of Sheffield, Sheffield.

[37] Tibshirani, R. (1992) Principal Curves Revisited. *Statistics and Computing*, **2**,183-190. http://dx.doi.org/10.1007/BF01889678

[38] Bishop, C.M. (1995) Neural Networks for Pattern Recognition. Oxford University Press, New York.

[39] Ripley, B.D. (1996) Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.

[40] Spierenburg, J.A. (1997) Dimension Reduction of Images Using Neural Networks. Master's Thesis, Leiden University, Leiden.

[41] Kramer, M.A. (1991) Non-Linear Principal Component Analysis Using Associative Neural Networks. *AIChE Journal*, **37**, 233-243. http://dx.doi.org/10.1002/aic.690370209

[42] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vettering, W.T. (1992) Numerical Recips in C: The Art of Scientific Computing. 2nd Edition, Cambridge University Press, Cambridge.

[43] Fowlkers, C., Belongie, S., Chung, F. and Malik, J. (2004) Specral Grouping Using the Nysroem Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 214-225.

[44] Marida, K.V., Kent, J.T. and Bibby, J.M. (1995) Multivariate Analysis. Probability and Mathematical Statistics. Academic Press, Waltham.

[45] Faloutsos, C. and Lin, K.I. (1995) FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In: Carey, M.J. and Schneider, D.A., Eds., *Proceedings of the* 1995 *ACM SIGMOD International Conference on Management of Data*, San Jose, 163-174. http://dx.doi.org/10.1145/223784.223812

[46] Fodor, I.K. (2002) A Survey of Dimension Reduction Techniques. Center for Applied Scientific Computing, Livermore National Laborary, Livermore.

[47] Chung, F.R.K. (1997) Spectral Graph Theory. American Mathematical Society. *CBMS Regional Conference Series in Mathematics in American Mathematical Society*, **212**, 92.

[48] Belkin, M. and Niyogi, P. (2003) Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, **15**, 1373-1396. http://dx.doi.org/10.1162/089976603321780317

[49] Rivest, R., Cormen, T., Leiserson, C. and Stein, C. (2001) Introduction to Algorithms. MIT Press, Cambridge.

[50] Kumar, V., Grama, A., Gupta, A. and Karypis, G. (1994) Introduction to Parallel Computing. Benjamin-Cummings, Redwood City.

[51] Donoho, D. and Grimes, C. Hessian Eigenmaps: Locally Linear Embedding Techniques for High-Dimensional Data. *Proceedings of National Academy of Sciences*, **100**.

[52] Ye, Q. and Zhi, W.F. (2003) Discrete Hessian Eigenmaps Method for Dimensionality Reduction.

[53] Kamhaltla, N. and Leen, T.K. (1994) Fast Non-Linear Dimension Reduction. In: *Advances in Neural Information Processing Systems*, Morgan Kaufmann Publishers, Inc., Burlington, 152-159.

[54] Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machin Learning. Addisn Wesley, Reading.

[55] Raymer, M.L., Goodman, E.D., Kuhn, L.A. and Jain, A.K. (2000) Dimensionality Reduction Using Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*, **4**, 164-171. http://dx.doi.org/10.1109/4235.850656

[56] Jones, G. (2002) Published Online: 15 APR. University of Sheffield, Sheffield.

[57] Kohavi, R. and John, G. (1998) The Wrapper Approach. In: Liu, H. and Motoda, H., Eds., *Feature Extraction, Construction and Selection*: *A Data Mining Perspective*, Springer Verlag, Berlin, 33-50. http://dx.doi.org/10.1007/978-1-4615-5725-8_3

[58] Huber, P.J. (1985) Projection Persuit. *Annals of Statistics*, **13**, 435-475. http://dx.doi.org/10.1214/aos/1176349519

[59] McCullagh, P. and Nelder, J.A. (1989) Generalized Linear Models. Chapman and Hall, Boca Raton. http://dx.doi.org/10.1007/978-1-4899-3242-6

[60] Dobson, A.J. (1990) An Introduction to Generalized Linear Models. Chapman and Hall, London. http://dx.doi.org/10.1007/978-1-4899-7252-1

[61] Leathwick, J.R. Elith, J. and Hastie, T. (2006) Comparative Performance of Generalized Additive Models and Multivariate Adaptive Regression Splines for Statistical Modelling of Species Distributions. *Ecological Modelling*, 188-196. http://www.stanford.edu/~hastie/Papers/Ecology/leathwick_etal_2006_mars_ecolmod.pdf

[62] Li, K.C. (2000) High Dimensional Data Analysis via SIR/PHD Approach. Lecture Note in Progress. http://www.stat.ucla.edu/kcli/

[63] Dennis Cook, R. and Li, B. (2002) Dimension Reduction for Conditional Mean in Regression. *Annals of Statistics*, **30**, 455-474. http://dx.doi.org/10.1214/aos/1021379861

[64] Zhang, Z. and Zha, H. (2002) Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. http://arxiv.org/pdf/cs.LG/0212008.pdf

[65] Wang, H. and Xia, Y. (2008) Sliced Regression for Dimension Reduction. Peking University & National University of Singapore, *Journal of the American Statistical Association*, **103**, 811-821.

[66] Feng, W.K., He, X. and Shi, P. (2002) Dimension Reduction Based on Canonical Correlation. *Statistica Sinica*, **12**, 1093-1113.

[67] Lectures on Fractals and Dimension Theory. http://homepages.warwick.ac.uk/masdbl/dimensiontotal.pdf

Appendix. Fractal and Topological Dimension

The main Reference for this appendix is [67]. Local (or topological) Methods (1): The definition of topological dimension was given by Brouwer in 1913: **A. Heyting, H. Freudenthal, Collected Works of L.E.J Brouwer, North Holland Elsevier, 1975.**

To begin at the very beginning: How can we best define the dimension of a closed bounded set $\Omega \subset \mathbb{R}^d$, say?

- When Ω is a manifold then the value of the dimension is an integer which coincides with the usual notion of dimension;
- For more general sets Ω we can have fractional dimensional
- Points, and countable unions of points, have zero dimension.

Local (or topological) Methods (2): The earliest attempt to define the dimension:

Definition 1 *We can define the Topological dimension* $\left(\dim_T \Omega\right)$ *by induction. We say that* Ω *has zero dimension if for every point* $x \in \Omega$ *every sufficiently small ball about* x *has boundary not intersecting* Ω. *We say that* Ω *has dimension* d *if for every point* $x \in \Omega$ *every sufficiently small ball about* x *has boundary intersecting* Ω *in a set of dimension* $d-1$.

Local (or topological) Methods (3):

Definition 2 *Given* $\epsilon > 0$, *let* $N(\epsilon)$ *be the smallest number of* ϵ*-balls needed to cover* Ω. *The Box dimension is*

$$\dim_B \Omega := \limsup_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log\left(\frac{1}{\epsilon}\right)}$$

Example 1 *For* $\Omega = \left\{\frac{1}{n} : n \geq 1\right\} \cup \{0\}$

$$\dim_B \Omega = \frac{1}{2}$$

Local (or topological) Methods (4): The Hausdorff dimension $\dim_H \Omega$ for a closed bounded set $\Omega \subset \mathbb{R}^d$ is defined as follows:

Definition 3 *Consider a cover* $\mathcal{U} = \{U_i\}$ *for* Ω *by open sets. For* $\delta > 0$ *we can define*

$$H_\epsilon^\delta(\Omega) = \inf_{\mathcal{U}} \left\{\sum_i \operatorname{diam}(U_i)^\delta\right\}$$

where the infimum is taken over all open covers $\mathcal{U} = \{U_i\}$ such that $\operatorname{diam}(U_i) \leq \epsilon$. Then $H^\delta(\Omega) = \lim_{\epsilon \to 0} H_\epsilon^\delta(\Omega)$ and finally,

$$\dim_H \Omega := \inf\left\{\delta : H^\delta(\Omega) = 0\right\}$$

- Fact1: For any countable set Ω we have $\dim_H \Omega = 0$
- Fact2: $\dim_H \Omega \leq \dim_B \Omega$

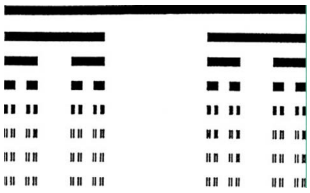
Local (or topological) Methods (4) as shown in **Figure 1**.

Local (or to pological) Methods (5):

Example 2 (von Koch curve: [ ]) The von Koch curve is a standard fractal construction. Starting from $\Omega_0 = [0,1]$, we associate to each piecewise linear curve Ω_n in the plane (which is a union of 4^n segments of length 3^{-n}) a new one Ω_{n+1}. This is done by replacing the middle third of each line segment by the other two sides of an equilateral triangle bases there. Alternatively, one can start from an equilateral triangle and apply this iterative procedure to each of the sides one gets a snowflake curve.

For Ω = von Koch curve, both the box dimension and Hausdorff dimension are equal in fact, as shown in **Figure 2**:

$$\dim_H(\quad) = \dim_B(\quad) = \frac{\log 4}{\log 3}$$

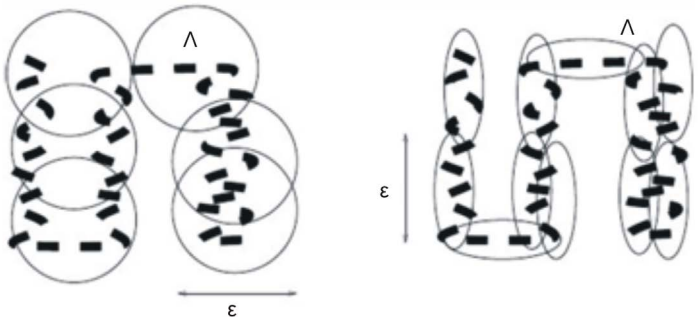


Example 3 (Ω : the Middle third Cantor set E_2 , This is the set of closed set of points in the unit interval whose triadic expansion does not contain any occurrence of the digit 1:

$$\Omega := \left\{ \sum_{k=1}^{\infty} \frac{i_k}{3^k} : i_k \in \{0,2\} \right\}$$

For the middle third Cantor set both the Box dimension and the Hausdorf dimension are $\frac{\log 2}{\log 3} = 0.690\cdots$

The set E_2 is the set of points whose continued fraction expansion contains only the terms 1 and 2. Unlike the Middle third Cantor set, the dimension of this set is not explicitly known in a closed form and can only be numerically estimated to the desired level of accuracy. as shown in **Figure 3,** For the Sierpinski carpet both the Box dimension and the Hausdorff dimension are equal to $\frac{\log 8}{\log 3} = 1.892\cdots$



(I) COVER BY BALLS (FOR DIM_B (Λ);
(II) COVER BY OPEN SETS (FOR DIM_H (Λ))

Figure 1. (I) Cocer by balls, (II) Cover by open sets.

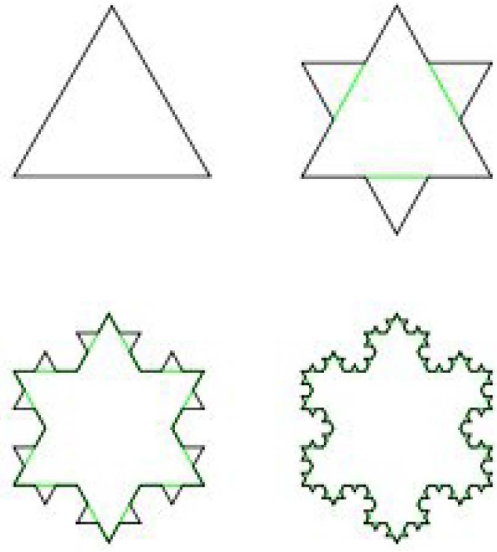


Figure 2. The construction of von Koch-curve.

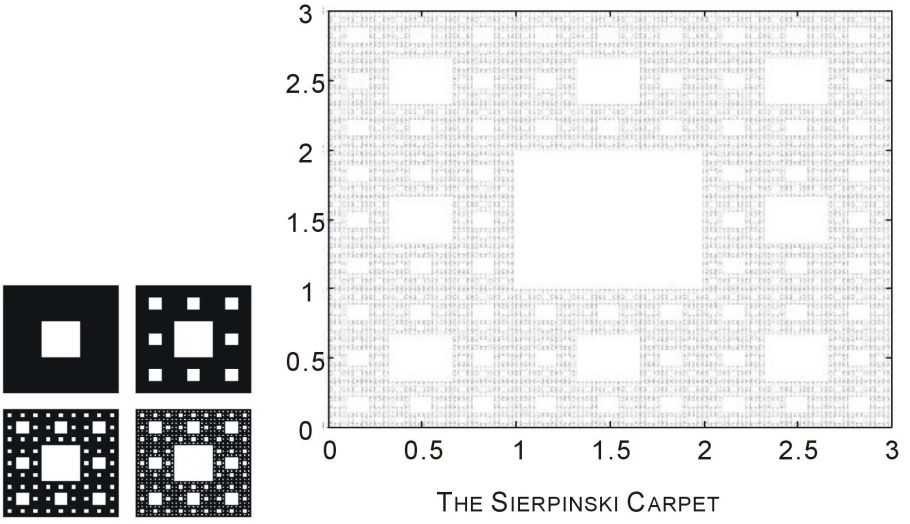


Figure 3. The construction Sierpinski Carpet.