Available online at www.sciencedirect.com

ScienceDirect

Computational Statistics & Data Analysis 51 (2006) 918–930

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Automatic dimensionality selection from the scree plot via the use of profile likelihood

Mu Zhu*, Ali Ghodsi

Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ont., Canada N2L 3G1

Received 14 January 2005; received in revised form 21 September 2005; accepted 21 September 2005
Available online 25 October 2005

Abstract

Most dimension reduction techniques produce ordered coordinates so that only the first few coordinates need be considered in subsequent analyses. The choice of how many coordinates to use is often made with a visual heuristic, i.e., by making a scree plot and looking for a "big gap" or an "elbow." In this article, we present a simple and automatic procedure to accomplish this goal by maximizing a simple profile likelihood function. We give a wide variety of both simulated and real examples.

Keywords: Data compression; Denoising; Isomap; Latent semantic indexing; Manifold learning; Principal component analysis (PCA); Resampling methods; Singular value decomposition (SVD)

1. Introduction

In modern applications, we often encounter high-dimensional data. More often than not, the intrinsic dimensionality of the data is much lower, i.e., even though the data are lying in a high-dimensional space, only a few dimensions are actually important for the analysis. Various dimension reduction techniques are available. Suppose $\mathbf{x} \in \mathbb{R}^p$ is a p-dimensional vector for some relatively large p. Most of these dimension reduction techniques will produce ordered mappings from $\mathbb{R}^p$ to $\mathbb{R}$, say $\alpha_1(\mathbf{x}), \alpha_2(\mathbf{x}), \ldots, \alpha_m(\mathbf{x})$ where $m \leqslant p$, such that $\alpha_1(\mathbf{x})$ is the most important coordinate, $\alpha_2(\mathbf{x})$ is the next most important coordinate, and so on. Associated with each mapping is a measure of its relative importance, say $d_1 \geqslant d_2 \geqslant \cdots \geqslant d_m$, so that the resulting coordinates can be ordered in a meaningful way. Dimension reduction is then achieved by selecting only the top few coordinates.

The problem that we shall focus on in this article is that of deciding how many coordinates should be retained. Ideally, if the intrinsic dimensionality is, say 3, we would like to see $d_1, d_2, d_3 > 0$ and $d_j = 0$ for all $j > 3$. However, this seldom happens because the data are often noisy.

To review some of the commonly used techniques for making such a decision, it is convenient to focus on the case of principal component analysis (PCA), which is perhaps the best-known dimension reduction technique of all.

* Corresponding author. Tel.: +1 519 888 4567x6987; fax: +1 519 746 1875.
E-mail addresses: m3zhu@uwaterloo.ca (Mu Zhu), aghodsib@uwaterloo.ca (Ali Ghodsi).

0167-9473/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2005.09.010

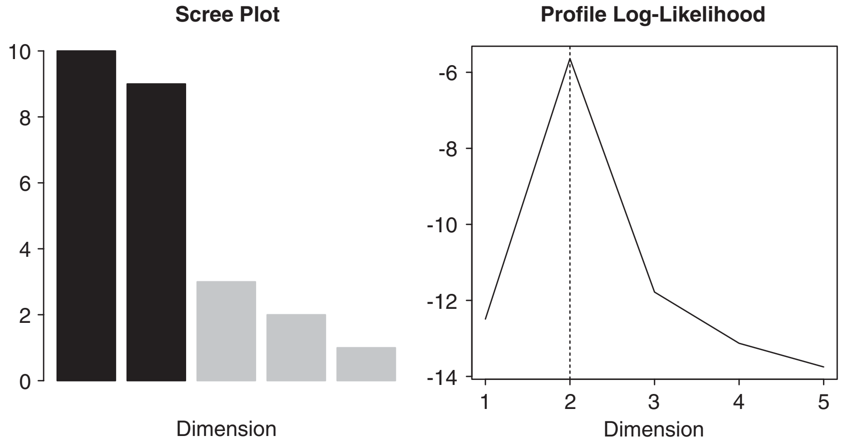

Fig. 1. A first example. Suppose the eigenvalues are 10, 9, 3, 2, 1; there exists a “big gap” between the second and third eigenvalues.

Given data $\mathbf{x} \in \mathbb{R}^p$, the principal components are defined by unit vectors $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_p \in \mathbb{R}^p$ such that

$$\mathrm{Var}\left(\boldsymbol{\alpha}_1^{\mathrm{T}}\mathbf{x}\right) \geqslant \mathrm{Var}\left(\boldsymbol{\alpha}_2^{\mathrm{T}}\mathbf{x}\right) \geqslant \cdots \geqslant \mathrm{Var}\left(\boldsymbol{\alpha}_p^{\mathrm{T}}\mathbf{x}\right)$$

and $\mathrm{Cov}\left(\boldsymbol{\alpha}_i^{\mathrm{T}}\mathbf{x}, \boldsymbol{\alpha}_j^{\mathrm{T}}\mathbf{x}\right) = 0$ for all $i \neq j$.

Therefore, PCA produces mappings from $\mathbb{R}^p$ to $\mathbb{R}$ that are simply projections, i.e., $\alpha_j(\mathbf{x}) = \boldsymbol{\alpha}_j^{\mathrm{T}}\mathbf{x}$, and the importance of mapping j is simply measured by the marginal variance along the projection, i.e., $d_j = \mathrm{Var}\left(\boldsymbol{\alpha}_j^{\mathrm{T}}\mathbf{x}\right)$. Mardia et al. (1979) give a nice overview of PCA as well as other related multivariate techniques. Let $\mathbf{S}$ be the sample variance–covariance matrix of the data; the principal components $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_p$ are simply the eigenvectors of $\mathbf{S}$, and $d_1 \geqslant d_2 \geqslant \cdots \geqslant d_p$ are simply the (ordered) eigenvalues of $\mathbf{S}$.

In order to determine how many principal components are needed, a number of approaches are available; Jolliffe (2002, Chapter 6) gave perhaps the most comprehensive and up-to-date summary on the status of the current practices. These methods can be roughly classified into the following categories although within each category there are still various minor variations:

(1) *Percent variance*: Find the smallest number of components to capture a certain percentage of the total variance, i.e., retain the top q components where q is the smallest integer between 1 and p such that

$$\frac{d_1 + d_2 + \cdots + d_q}{d_1 + d_2 + \cdots + d_p} \geqslant \gamma,$$

where γ is a pre-determined level, say, 80% or 90%.

(2) *Scree plot*: Plot the eigenvalues $d_1, d_2, \ldots, d_p$ in descending order (often called a scree plot) and look for a “big gap” or an “elbow” in such a graph. For example, as illustrated in the left panel of Fig. 1, there is a “big gap” between the second and third eigenvalues, so the first two principal components are retained and the rest, discarded.

(3) *Sequential tests*: Sequentially conduct a series of formal hypothesis tests to determine whether the small eigenvalues are equal. For $j = 1, 2, \ldots, p-1$, consider a series of null hypotheses:

$$\mathrm{H}_{0,j} : d_p = d_{p-1} = \cdots = d_{p-j}.$$

We start by testing $\mathrm{H}_{0,1}$, $\mathrm{H}_{0,2}$ and so on until a null hypothesis is first rejected. Suppose $\mathrm{H}_{0,q}$ is the first rejected null hypothesis, then the first $p-q$ components are retained.

(4) *Resampling*: Estimate the null distribution of each d_j by resampling the data repeatedly, e.g., via permutation or the bootstrap. Retain component j if d_j exceeds the 95 percentile of the corresponding null distribution and discard component j if otherwise.

It is clear that all of these methods above depend heavily on the eigenvalues of $\mathbf{S}$, but in principle each d_j does not have to be an eigenvalue for these four approaches to work. For many other techniques the importance index associated with each coordinate can no longer be interpreted as a marginal variance like in PCA, but the four approaches outlined above are still applicable as long as for each coordinate $\alpha_j(\mathbf{x})$ there exists an explicit measure of its relative importance, d_j. In fact, even for PCA it is not uncommon in practice to make a scree plot using $\sqrt{d_j}$ (e.g., Hastie et al., 2001, Section 14.5) or $\log\left(d_j\right)$ (e.g., Jolliffe, 2002, Section 6.1.3) instead of d_j itself.

The drawbacks of each of these methods are fairly obvious and well understood. The choice of the threshold γ in the percent variance approach and the decision of where the "gap" or the "elbow" is in the scree plot approach are both highly subjective. The validity of the sequential tests approach relies on the assumption that the underlying data follow a multivariate normal distribution and is only approximately correct even then; moreover, "it is difficult to get even an approximate idea of the overall significance level because the number of tests done is not fixed but random, and the tests are not independent of each other" (Jolliffe, 2002, Section 6.1.4). The resampling techniques have the clear disadvantage of being computationally expensive and slow, especially for large data sets.

In this article, we shall not be concerned with the respective drawbacks of each of these approaches. Instead, we note that all of these methods can be implemented to work automatically except the scree plot approach where a visual examination by the data analyst is necessary. Even then, it is still very difficult for different analysts to agree upon exactly where the "gap" or the "elbow" is. The main goal of this article is to propose a very simple method to find this "gap" or "elbow" in the scree plot in a fairly objective and fully automatic fashion. So far, we are not aware of any such automatic techniques other than the one outlined by Hastie et al. (2001, Section 14.5), which is based on resampling techniques and hence cumbersome; we will be more specific about the resampling technique below (Section 4.1).

But since the scree plot approach is somewhat ad hoc to begin with, our work here will necessarily inherit this drawback. The reason why we think our work is still valuable is because the scree plot approach is, in fact, one of the most commonly used in practice and a simple and automatic procedure to perform this task will undoubtedly benefit a large number of data analysts.

2. Methodology

The gist of our method (Section 2.2) consists of explicitly constructing a model for the numbers d_j $(j=1,2,\ldots,p)$ and estimating the position of the "gap" or the "elbow" by maximizing a profile likelihood function. Before we proceed with the details, we briefly review of concept of profile likelihood.

2.1. *Profile likelihood*

Suppose $l(\theta,\psi;y)$ is a likelihood function, which depends on two parameters. Suppose further that θ is the main parameter of interest and ψ is a so-called nuisance parameter. To facilitate statistical inference on θ, it is often desirable to get rid of the nuisance parameter. The profile likelihood is a convenient way of doing so. In particular, the profile likelihood for θ is defined as

$$l_\theta(\theta;y)=l\left(\theta,\hat{\psi}_\theta;y\right),$$

where $\hat{\psi}_\theta$ is the maximum likelihood estimate (MLE) of ψ for fixed θ. Alternative approaches of eliminating the nuisance parameter include the use of marginal likelihood or conditional likelihood; Sprott (2000) gives a number of nice examples. The profile likelihood has the disadvantage of not being a proper likelihood function, e.g., its first derivative does not have mean zero. On the other hand, it is always available whereas marginal and conditional likelihoods are only available in very special problems; see McCullagh and Nelder (1989, Chapter 7) and Sprott (2000). It is also clear that the maximum of the profile likelihood l_θ is the same as the overall MLE of θ. Therefore, the use of profile likelihood does not really invite any controversy as far as point estimation is concerned.

2.2. *The method*

Again, let $d_1\geqslant d_2\geqslant\cdots\geqslant d_p>0$ be the ordered measures of importance of the corresponding coordinates. In the case of PCA these are ordered eigenvalues. Our basic idea is very simple. Given a fixed number $1\leqslant q\leqslant p$, write

$\mathscr{S}_1 = \left\{d_1, d_2, \ldots, d_q\right\}$ and $\mathscr{S}_2 = \left\{d_{q+1}, d_{q+2}, \ldots, d_p\right\}$. If a “gap” or an “elbow” exists at position q, then we can think of $\mathscr{S}_1$ and $\mathscr{S}_2$ as samples from two different distributions. Based on this point of view, we proceed to model $\mathscr{S}_1$ as an independent sample from $f(d;\theta_1)$ and $\mathscr{S}_2$ as an independent sample from $f(d;\theta_2)$. The independence assumption here is convenient but somewhat naïve, but, as we shall demonstrate below, we are able to construct an effective procedure despite the naïve independence assumption.

The log-likelihood function under the naïve independence assumption can be written as

$$l(q, \theta_1, \theta_2) = \sum_{i=1}^{q} \log f(d_i; \theta_1) + \sum_{j=q+1}^{p} \log f\left(d_j; \theta_2\right). \tag{1}$$

For any given q, MLEs of θ_1 and θ_2 can be obtained separately from $\mathscr{S}_1$ and $\mathscr{S}_2$. By plugging in these estimates into (1), we obtain a profile log-likelihood for q:

$$l_q(q) = \sum_{i=1}^{q} \log f\left(d_i; \hat{\theta}_1(q)\right) + \sum_{j=q+1}^{p} \log f\left(d_j; \hat{\theta}_2(q)\right). \tag{2}$$

We use the notation $\hat{\theta}_j(q)\ (j=1,2)$ to emphasize that the MLE for θ_1 and θ_2 may depend on q. An estimate of q can then be obtained by maximizing the profile log-likelihood above. To do so, a simple exhaustive search can be used, at least conceptually. That is, we simply compute $l_q(1), l_q(2), \ldots, l_q(p)$ and estimate q with

$$\hat{q} = \underset{k=1,2,\ldots,p}{\operatorname{argmax}}\ l_q(k).$$

For simplicity, we choose f to be the Gaussian distribution:

$$f\left(d; \mu_j, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{\left(d-\mu_j\right)^2}{2\sigma^2}\right\} \quad \text{for } j=1,2.$$

Note here that it is important to use a common scale parameter σ for both $\mathscr{S}_1$ and $\mathscr{S}_2$. If a different σ is used for each model, the model becomes too flexible and it is possible for the profile log-likelihood (2) to become infinite, e.g., when $q=1$ and $q=p-1$. Though elementary, we give the MLEs for μ_1, μ_2 and σ^2 explicitly here for completeness. For any given q, the MLEs for μ_1 and μ_2 are simply the sample averages,

$$\hat{\mu}_1 = \frac{\sum_{d_i \in \mathscr{S}_1} d_i}{q} \quad \text{and} \quad \hat{\mu}_2 = \frac{\sum_{d_j \in \mathscr{S}_2} d_j}{p-q},$$

and the MLE for the common scale parameter σ^2 is the usual pooled estimate

$$\hat{\sigma}^2 = \frac{(q-1)s_1^2 + (p-q-1)s_2^2}{p-2},$$

where s_j^2 is the sample variance of $\mathscr{S}_j$.

In the illustrative example depicted in Fig. 1, the eigenvalues are 10, 9, 3, 2, 1. The left panel of Fig. 1 is a scree plot that plots the eigenvalues in descending order; we can see a clear “gap” between the second and third eigenvalues. The right panel of Fig. 1 plots the profile log-likelihood $l_q(q)$. Our MLE of q is 2, telling us to retain the first two principal components and discard the rest.

3. Simulated examples

3.1. A simple uniform experiment

Here we conduct an experiment that is similar in spirit to the example shown in Fig. 1. Suppose there are a total of 100 dimensions ($p=100$). We generate two different cases:

(a) $d_1, d_2, \ldots, d_{50} \sim$ Uniform[0, 45], $d_{51}, d_{52}, \ldots, d_{100} \sim$ Uniform[55, 100];
(b) $d_1, d_2, \ldots, d_{80} \sim$ Uniform[0, 49], $d_{81}, d_{82}, \ldots, d_{100} \sim$ Uniform[51, 100].

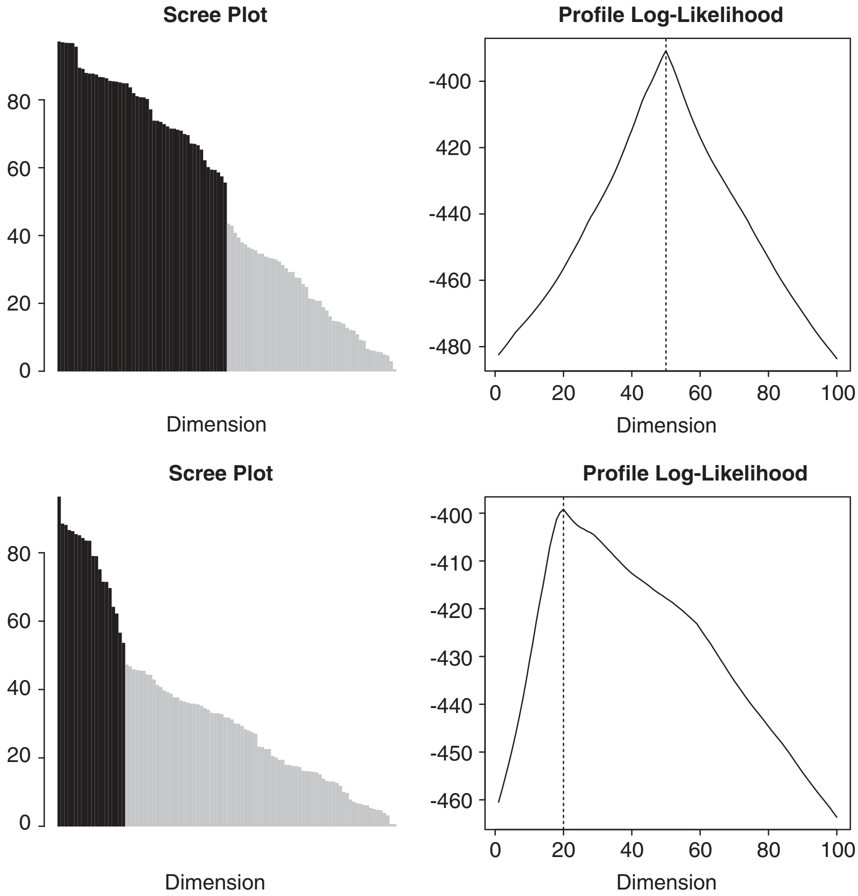

Fig. 2. An experiment. The two rows correspond to the two different cases (a) and (b), in that order. Refer to the text (Section 3.1) for more details.

Table 1
Median (MAD) from 100 repeated experiments

Simulation setting	True answer	Estimated answer
(a) $\leqslant 45$: 50	50	50.00
$\geqslant 55$: 50		(0.00)
(b) $\leqslant 49$: 80	20	20.00
$\geqslant 51$: 20		(0.00)

Clearly (b) is a more difficult case than (a). In case (a), the gap between the two groups of eigenvalues is very large, whereas in case (b), the gap is very small and hard to detect by the human eye. In case (b), we also try to mimic reality by generating fewer eigenvalues from the large group. As a result, the "elbow" effect is clearly noticeable (Fig. 2), but it is still hard to pinpoint visually where exactly the "elbow" is. The results are shown graphically in Fig. 2.

In order to account for the variability caused by random generation of these two cases, we repeat the above experiments 100 times, each time generating two slightly different cases (a) and (b) using the same mechanism. Table 1 summarizes the results. Reported are the *median* estimate of q over 100 repetitions; the number in the parentheses below are the *median absolute deviation* (MAD) of the 100 estimates. We can see that our very simple procedure is capable of producing accurate estimates of q in both cases. Note also that even though the numbers $d_1, d_2, \ldots, d_{100}$ here are generated from the highly non-Gaussian uniform distribution, our method is quite robust and still works quite well.

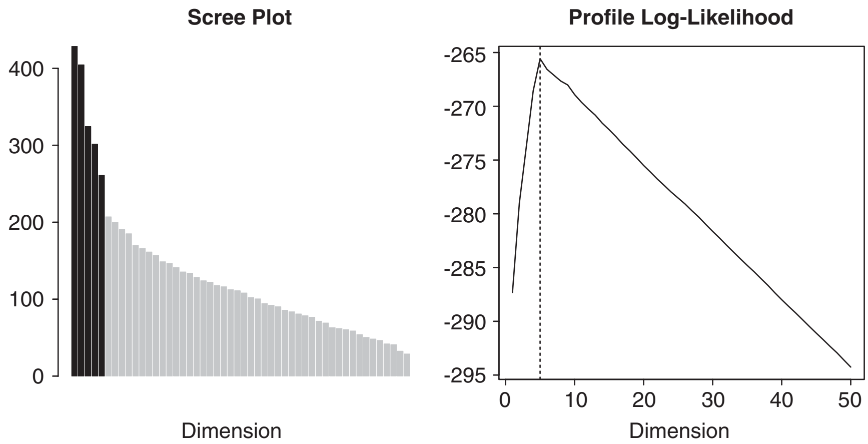

Fig. 3. Simulated linear subspace example. A total of 200 observations in $\mathbb{R}^{50}$ lying close to a 5-dimensional subspace.

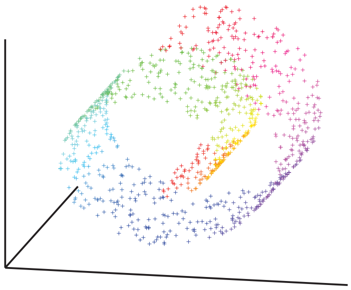

Fig. 4. Swiss roll data example. These three-dimensional data lie close to a two-dimensional manifold taking the shape of a Swiss roll.

3.2. A linear subspace

In this example, we generate data in $\mathbb{R}^{50}$ that lie approximately in a five-dimensional subspace. We first randomly generate 5 basis vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_5 \in \mathbb{R}^{50}$; these five vectors span a five-dimensional subspace in $\mathbb{R}^{50}$, call it $\mathscr{S}$. We then randomly generate 200 data points in $\mathbb{R}^{50}$, project them onto the subspace $\mathscr{S}$, and add a bit of noise. To be specific, we start by randomly generating 10, 000 numbers from the standard Gaussian distribution and arranging them into a 50 × 200 matrix, $\mathbf{Z}$. Then we construct

$$\mathbf{X} = \mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{Z} + \mathbf{E} \quad \text{where} \quad \mathbf{B} = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \cdots \quad \mathbf{b}_5]$$

is a 50 × 5 matrix stacking the basis vectors $\mathbf{b}_i$ as column vectors, and the matrix $\mathbf{E}$ is 50 × 200 with each element generated independently from a Gaussian distribution with mean 0 and standard deviation 0.75. Each column of $\mathbf{X}$ is then a vector in $\mathbb{R}^{50}$ that lies close to the subspace $\mathscr{S}$. Fig. 3 shows the results of applying our profile likelihood method together with PCA to this simulated data. The true dimensionality of five is correctly identified.

3.3. A nonlinear manifold

In this example, we analyze a simulated data set known as the Swiss roll data, made popular by Tenenbaum et al. (2000) and available at the web site `http://isomap.stanford.edu/`. The data are in $\mathbb{R}^3$ but, as Fig. 4 shows, all the data are generated to lie close to a two-dimensional manifold taking the shape of a Swiss roll.

To recover the global internal coordinates of a nonlinear manifold, we use a technique called Isomap (Tenenbaum et al., 2000). Isomap is a nonlinear generalization of multidimensional scaling (MDS; Cox and Cox, 2001). The key idea is to compute the MDS, not in the input space, but in the geodesic space of the manifold. The geodesic distances

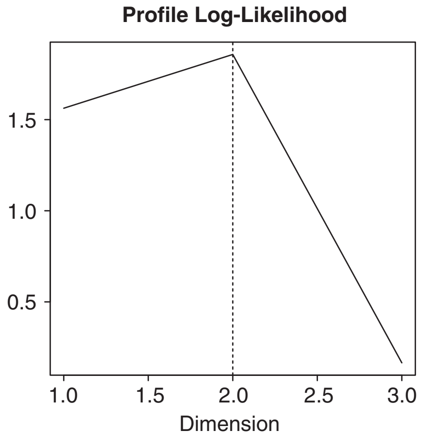

Fig. 5. Swiss roll data example. The profile log-likelihood for selecting the intrinsic dimensionality of this data.

represent the shortest paths along the curved surface of the manifold measured as if the surface were flat. This can be approximated by a sequence of short steps between neighboring sample points. Isomap then applies MDS to the geodesic distances to find a low-dimensional mapping with similar pairwise distances.

We first apply Isomap to the Swiss roll data using exactly the same control parameters as used in the original Isomap paper (Tenenbaum et al., 2000) and then apply our profile likelihood technique to the resulting singular values. Fig. 5 shows that we correctly identify the intrinsic dimensionality of two.

4. Real data examples

In this section, we illustrate the usefulness of our profile likelihood method with a wide variety of real examples, including data compression, image denoising, document retrieval and manifold learning.

4.1. Data compression

We first consider an interesting application of PCA as a data compression technique to a real data set of handwritten digits (Hastie et al., 2001, Section 14.5). The entire data set consists of 16 × 16 images of handwritten digits, 0, 1, 2, …, 9. Following Hastie et al. (2001), we only work with a subset of the data here, one that consists of all the images for the digit 8. The choice of the digit 8 is due to it being a lucky number rather than anything else. There are altogether 542 observations; each image is treated as a vector in $\mathbb{R}^{256}$. In other words, our data matrix $\mathbf{X}$ is 542 × 256. Fig. 6 shows a random selection of some of the images in our subset. The variation in the data is apparent.

For such a high-dimensional problem, it is generally believed that the intrinsic dimensionality q is much smaller. Hastie et al. (2001) outlined a resampling technique that can be used to estimate q for this problem. First of all, each column of the data matrix $\mathbf{X}$ is permuted; call the resulting matrix $\mathbf{X}'$. PCA is then applied to the original data $\mathbf{X}$ and the permuted data $\mathbf{X}'$. The resulting eigenvalues are then compared.

More often in practice, the operation of permuting the columns of $\mathbf{X}$ is repeated a number of times in order to obtain an empirical estimate of the null distribution for each eigenvalue when PCA is performed on structureless (permuted) data. The comparison is then made with the 95 percentile of the null distribution; the components whose eigenvalues are larger than the 95 percentile of the corresponding null distributions are retained (see, e.g., Jolliffe, 2002, Section 6.1.3).

The left panel of Fig. 7 shows such a comparison for this particular example. The 95 percentile of the null distributions are obtained by permuting the data matrix 100 times. Note that no big gap seems to exist but the “elbow” effect is apparent. However, it is hard to determine visually where exactly the “elbow” is located. By comparing the observed eigenvalues with the 95 percentile of the corresponding null distributions, we note that the crossover occurs at 20, giving an estimate of $\hat{q} = 20$. The profile log-likelihood (2) based on our procedure is shown in the right panel of Fig. 7.

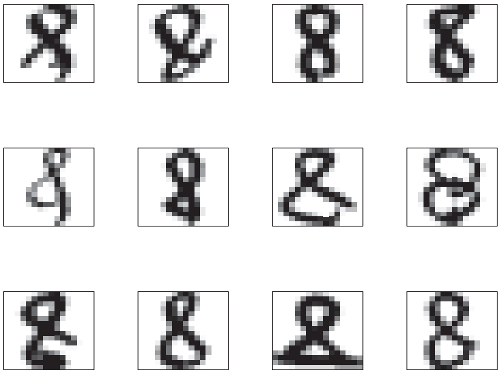

Fig. 6. A random selection of 12 images of the handwritten digit “8”.

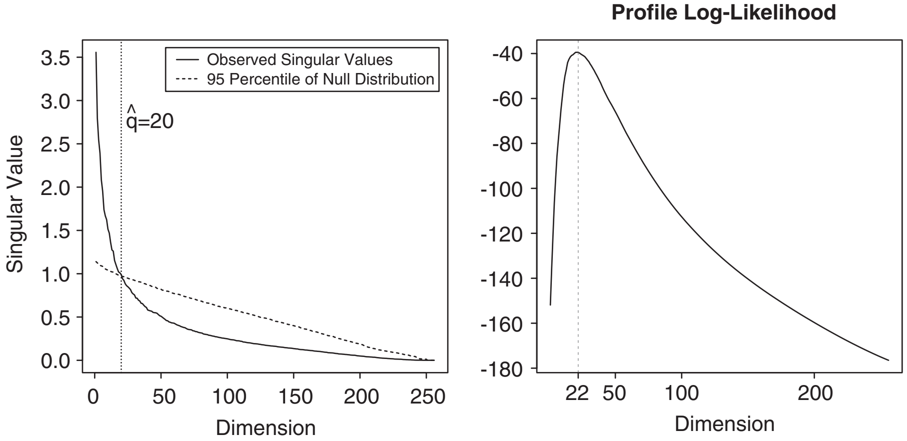

Fig. 7. Handwritten digit example. Left: Finding the intrinsic dimensionality by resampling. Right: The profile log-likelihood $l_q(q)$ for selecting the intrinsic dimensionality.

The maximum occurs at 22, giving an estimate of $\hat{q} = 22$, which is very close to the result obtained from the resampling technique.

4.2. Image denoising

In PCA, an observation **x** can be reconstructed by $\mathbf{V}_k\mathbf{V}_k^{\mathrm{T}}\mathbf{x}$, where $\mathbf{V}_k$ is a matrix consisting of the top k eigenvectors of the covariance matrix of **x** (Hastie et al., 2001, Section 14.5). If $\mathbf{x} \in \mathbb{R}^p$ is a noisy observation generated from a low-dimensional structure f whose dimensionality is $k < p$, then intuitively the first k eigenvectors should contain most of the information about f and the remaining eigenvectors should just contain noise. Therefore, PCA can be used as a data denoising technique. Suppose the observed data are noisy. If one finds the correct intrinsic dimensionality of the data k and reconstructs the data by using only the first k significant eigenvectors, one should expect to be able to filter out the noise while still capturing the important information in the data.

As an experiment, we use a data set studied by Roweis and Saul (2000) and available at the web site `http://www.cs.toronto.edu/~roweis/data.html`. The data set consists of 1965 20-pixel-by-28-pixel grey-scale images. We distort each image by adding Gaussian noises to each pixel with $\sigma = 25$. The top row of Fig. 8 shows five examples of these noisy images. We then apply PCA to the noisy data and use our profile likelihood technique to select the intrinsic dimensionality. We obtain an answer of $k = 10$ as the intrinsic dimensionality. Computer-reconstructed images using the top $k = 10$ eigenvectors are shown in the third row of Fig. 8.

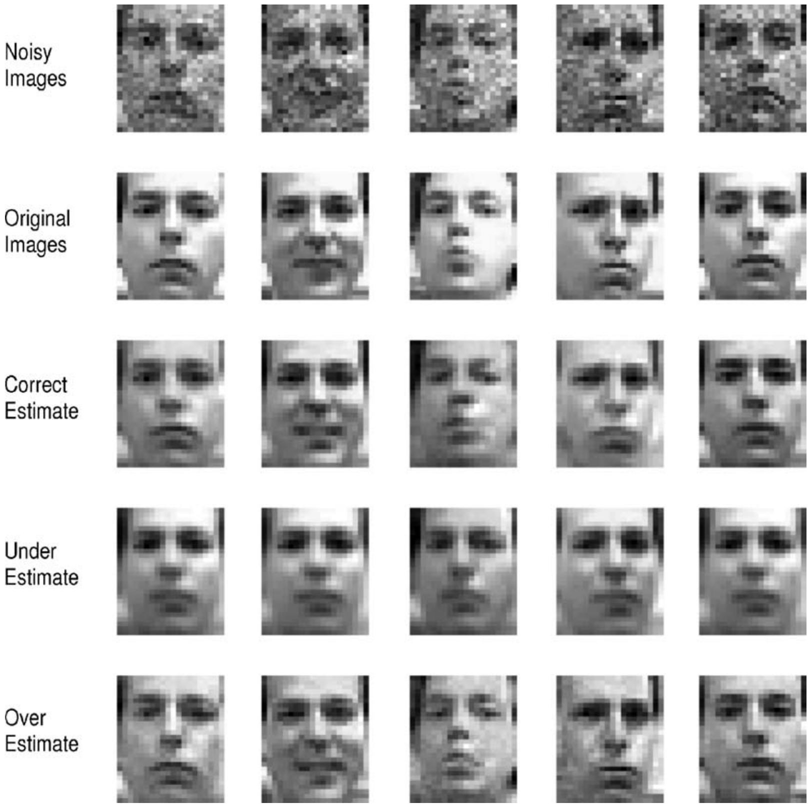

Fig. 8. Image denoising example. A sample from a total of 1965 images. Images in the first row are distorted by adding Gaussian noise with $\sigma = 25$. The second row are the corresponding original images. Images in the third row are reconstructions using the first k eigenvectors, where $k = 10$ is selected by our profile likelihood technique. The last two rows are reconstructions using fewer and more than 10 eigenvectors, respectively.

The last two rows of Fig. 8 show the reconstruction of the same images when the intrinsic dimensionality k is under- and over-estimated, respectively. It is clear from Fig. 8 that when k is underestimated we start to lose important features in the reconstruction (e.g., the smile in the second image, the protruding lips in the third image and the frown in the fourth image) whereas when k is overestimated we start to reconstruct noise. Fig. 9 shows the first 30 eigenvectors; these eigenvectors serve as basis images for the reconstruction. We can see the first 10 basis images all contain important facial features whereas noises start to become more and more visible in the subsequent basis images.

4.3. Latent semantic indexing

In this section, we analyze a benchmark data set in document retrieval known as the MEDLINE data set. References to this data set are numerous, including but certainly not limited to Dumais (1991) and Salton and Buckley (1988).

We first give some background. Given a collection of n documents and a list of p words, a term-document matrix $\mathbf{X}$ (n-by-p) can be constructed such that the element x_{ij} counts the number of times word j occurs in document i (Table 2). The ith row of $\mathbf{X}$ is a p-vector, say $\mathbf{x}_i \in \mathbb{R}^p$, that can be thought of as the profile of document i. A query can also be represented as a p-vector $\mathbf{q}$ where q_j counts the number of times word j occurs in the query. A document can then be said to be relevant to a query if its profile $\mathbf{x}$ is close to the query $\mathbf{q}$ in the sense that

$$\frac{\mathbf{x}^{\mathrm{T}}\mathbf{q}}{\|\mathbf{x}\|\,\|\mathbf{q}\|}$$

is large. Therefore the computation of $\mathbf{Xq}$ is a key operation for document retrieval.

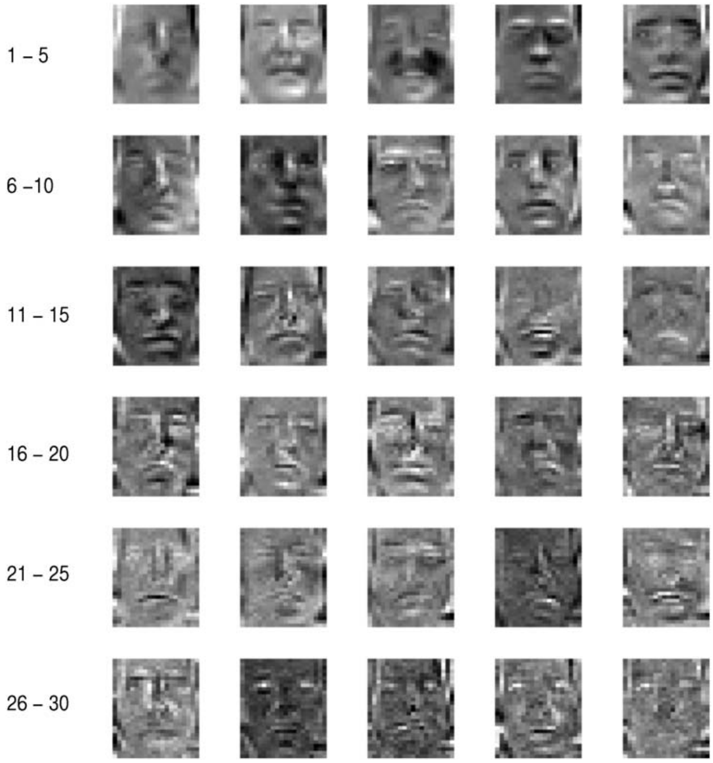

Fig. 9. Image denoising example. The first 30 eigenvectors.

Table 2
A term-document matrix

Document	...	“bar”	...	“music”	...	“piano”	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
15	...	5	...	0	...	1	...
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
623	...	0	...	3	...	3	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Of course, it is easy to understand that the p words and the n documents should not be treated identically. For example, some words are more specific and only appear in a small number of documents whereas others are more general and appear more frequently over the entire collection; some documents are longer than others and contain more words; and so on. In practice, the elements in the matrix $\mathbf{X}$ and the query vector $\mathbf{q}$ are often weighted to account for these facts. There are many different weighting schemes (see, e.g., Salton and Buckley, 1988; Dumais, 1991). We will not go into details here as they are not directly relevant to this article.

In latent semantic indexing (e.g., Deerwester et al., 1990), instead of $\mathbf{Xq}$ one uses $\mathbf{X}_k\mathbf{q}$ where $k \leqslant \min(n, p)$ and $\mathbf{X}_k$ is the best rank-k approximation to the matrix $\mathbf{X}$, which is given by the singular value decomposition (SVD) of $\mathbf{X}$. That is, if $\mathbf{X} = \mathbf{UDV}^{\mathrm{T}}$ is the SVD of $\mathbf{X}$, then

$$\mathbf{X}_k = \mathbf{U}_k\mathbf{D}_k\mathbf{V}_k^{\mathrm{T}},$$

where $\mathbf{U}_k$, $\mathbf{V}_k$ are matrices consisting of the first k columns of $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{D}$ is a diagonal matrix whose diagonal entries consist of the first k singular values only. It is well-known that using a low-rank matrix $\mathbf{X}_k$ usually yields better retrieval

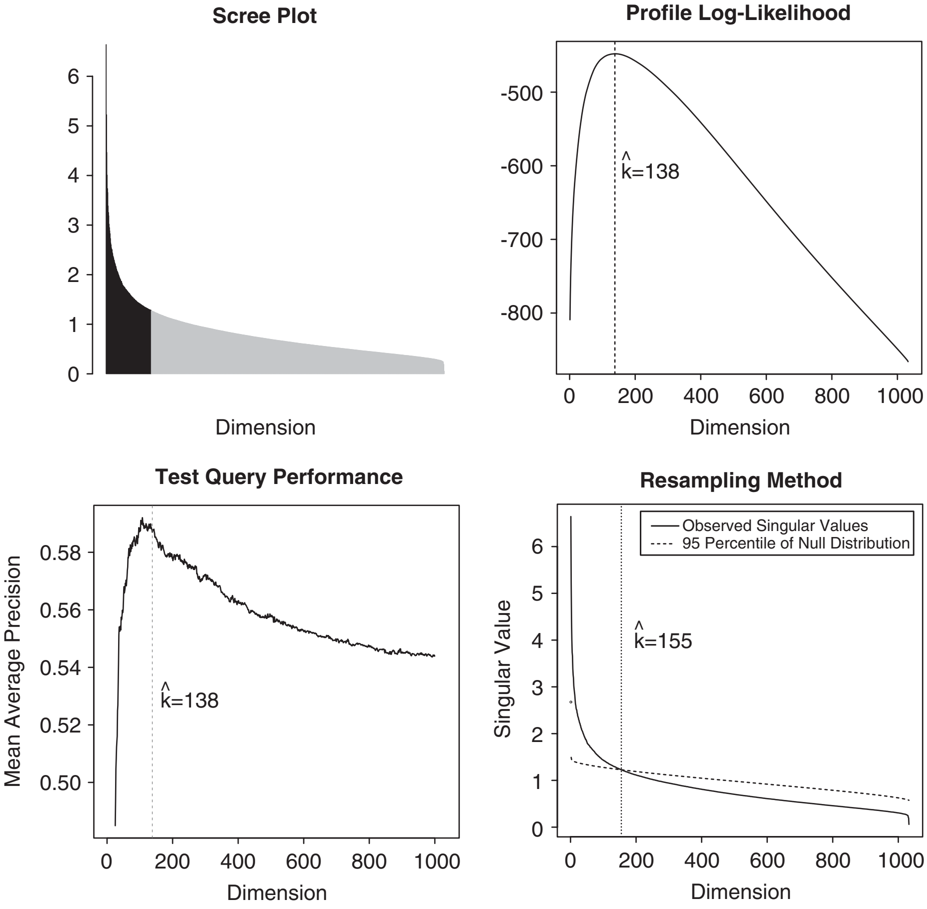

Fig. 10. Latent semantic indexing example using the MEDLINE data. Top left: The scree plot. Top right: The profile log-likelihood for selecting the best low-rank term-document matrix. Bottom left: Mean retrieval performances for the 30 test queries, where the retrieval result of each test query is evaluated by its average precision. Only the range between $k = 25$ and $k = 1000$ is plotted. Bottom right: Selecting the best low-rank term-document matrix by resampling.

results than using the full matrix **X**. The question we are interested in is: what is the best choice of k? Here a scree plot of the singular values of **X** can be used to make this decision. We will choose k by applying our profile likelihood method as well as the resampling method described in Section 4.1.

The MEDLINE data set contains 1033 documents and 30 test queries. The original data set consists of text files; the numeric term-document matrix **X** and query vectors $\mathbf{q}_i$ $(i = 1, 2, \ldots, 30)$ that we use here are constructed using a Matlab tool box called TMG (Zeimpekis and Gallopoulos, 2004) and was kindly provided to us by Professor Dimitris Zeimpekis of the University of Patras in Greece, who also suggested that we use the logarithmic local weighting and the so-called "GfIdf" global weighting for this data set. These weighting schemes are defined precisely in Dumais (1991); again we omit the details here. The data set from Professor Zeimpekis indexes all documents and queries using 5735 unique words, i.e., $\mathbf{x}, \mathbf{q} \in \mathbb{R}^{5735}$.

Various results are displayed in Fig. 10. The "elbow" in the scree plot is apparent (top left). By maximizing the profile log-likelihood, we obtain an estimate of $\hat{k} = 138$ (top right) whereas the resampling strategy (bottom right) yields an estimate of $\hat{k} = 155$. For practical purposes these two answers are quite close to each other given that rank$(\mathbf{X}) = 1033$. However, the computational burden of the resampling technique on a data set of this size (1033×5735) is much heavier.

The bottom left panel of Fig. 10 shows the mean retrieval performance for the 30 test queries; the performance for each test query is measured by the average precision of its retrieval results, a standard performance measure used in

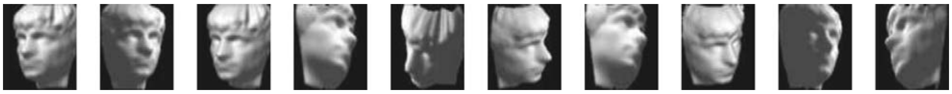

Fig. 11. A random sample of ten images from the face data studied by Tenenbaum et al. (2000).

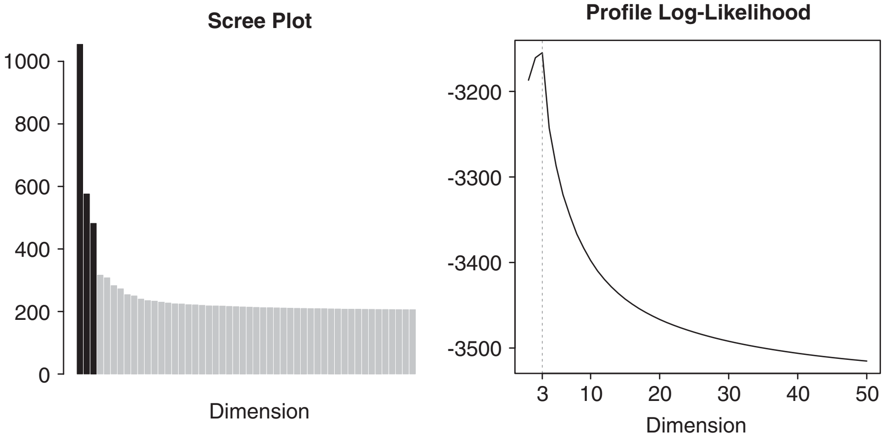

Fig. 12. Face data example. Left: The scree plot of singular values from Isomap. There are altogether 698 singular values; only the top 50 are plotted. Right: The corresponding profile log-likelihood function for selecting the intrinsic dimensionality of this data.

the information retrieval community (e.g., Peng et al., 2003; Dumais, 1991). Here we see that as far as the retrieval performance on these test queries are concerned, the optimal k is actually very close to our estimate of $\hat{k} = 138$.

4.4. Manifold learning

Finally, we consider a real data example where the low-dimensional structure of the data is a manifold rather than a subspace. The data set is used as a main example in the original Isomap paper (Tenenbaum et al., 2000) and consists of a total of 698 64-pixel-by-64-pixel images of the same face; each image is generated by using three free parameters: lighting direction, horizontal and vertical orientation. Therefore it is argued that these 4096-dimensional observations lie on an intrinsically three-dimensional manifold (Tenenbaum et al., 2000). A few example of these images are displayed in Fig. 11; the entire data set is available from the web site `http://isomap.stanford.edu/`.

We first apply Isomap to the data, again using exactly the same control parameters as used in the original Isomap paper (Tenenbaum et al., 2000) and then apply the profile likelihood method to the resulting singular values. Fig. 12 indicates that our technique correctly recognizes the intrinsic dimensionality of three, as indicated by the maximum of the profile log-likelihood function.

5. Discussion

The method we have used to maximize the profile log-likelihood (2) is very simple: we simply evaluate (2) for every possible $q = 1, 2, \ldots, p$ and pick the q that gives the largest profile log-likelihood. In Section 2.2, we emphasized that this exhaustive search was viable “at least conceptually,” which implied an initial intention to develop perhaps more efficient procedures. However, we have found such intention highly unnecessary in practice. Note that the original log-likelihood function (1) is based on one-dimensional “data”: $d_1, d_2, \ldots, d_p$. Even for very large p, such an exhaustive search over q is computationally *trivial* with the current technology. In all the experiments we have conducted so far, the most computationally expensive step is the calculation of $d_1, d_2, \ldots, d_p$. In PCA, for example, this step involves performing a SVD of the $n \times p$ data matrix $\mathbf{X}$, which is an expensive operation for large n and p. The amount of time required to maximize the profile log-likelihood (2) is a tiny fraction of the SVD operation.

6. Conclusion

We have presented a simple and automatic method to find, via maximum profile likelihood, the position of the "big gap" or the "elbow" in the scree plot. Various experiments using both simulated and real examples have confirmed that our method is useful and easy-to-apply in practice.

For relatively simple problems, e.g., Fig. 1 and case (a) in Section 3.1, it can be said that our method merely formalizes and automates what can be easily accomplished by the human eye. For example, anyone with minimal working knowledge of PCA will have no difficulty making the right decision by looking at the scree plot of the eigenvalues. However, for case (b) in Section 3.1 as well as some of the real data examples we have shown above, it is not clear at least to the authors' eyes where the correct cutoff should be. That we can get a good estimate in these more difficult situations without having to resort to computationally expensive resampling techniques is precisely why the proposed procedure can be quite useful in practice.

Acknowledgements

This research is partially supported by the Natural Science and Engineering Research Council of Canada. The authors would like to thank Professor Wenqing He for lending them the new edition of Jolliffe's book, Professor Dimitris Zeimpekis for providing the term-document matrix and the query vectors for Section 4.3, and Professor R.W. Oldford for his very helpful discussions and comments.

References

Cox, T.F., Cox, M.A.A., 2001. Multidimensional Scaling. second ed. Chapman & Hall, New York.

Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A., 1990. Indexing by latent semantic analysis. J. Soc. Inform. Sci. 41 (6), 391–407.

Dumais, S.T., 1991. Improving the retrieval of information from external sources. Behavior Res. Methods Instrum. Comput. 23 (2), 229–236.

Hastie, T.J., Tibshirani, R.J., Friedman, J.H., 2001. The Elements of Statistical Learning: Data-Mining, Inference and Prediction. Springer, Berlin.

Jolliffe, I.T., 2002. Principal Component Analysis. second ed. Springer, Berlin.

Mardia, K.V., Kent, J.T., Bibby, J.M., 1979. Multivariate Analysis. Academic Press, New York.

McCullagh, P., Nelder, J.A., 1989. Generalized Linear Models. second ed. Chapman & Hall, New York.

Peng, F., Schuurmans, D., Wang, S., 2003. Augmenting naive Bayes classifiers with statistical language models. Inform. Retrieval 7 (3), 317–345.

Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326.

Salton, G., Buckley, C., 1988. Term weighting approaches in automatic text retrieval. Inform. Process. Management 24 (5), 513–523.

Sprott, D.A., 2000. Statistical Inference in Science. Springer,

Tenenbaum, J.B., de Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323.

Zeimpekis, D., Gallopoulos, E., 2004. TMG: a MATLAB toolbox for generating term-document matrices from text collections. Technical Report HPCLAB-SCG 1/6-04, Computer Engineering & Informatics Department, University of Patras, Greece.