A survey of dimension reduction techniques

Imola K. Fodor
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory
P.O. Box 808, L-560, Livermore, CA 94551
fodor1@llnl.gov

June 2002

1 Introduction

Advances in data collection and storage capabilities during the past decades have led to an information overload in most sciences. Researchers working in domains as diverse as engineering, astronomy, biology, remote sensing, economics, and consumer transactions, face larger and larger observations and simulations on a daily basis. Such datasets, in contrast with smaller, more traditional datasets that have been studied extensively in the past, present new challenges in data analysis. Traditional statistical methods break down partly because of the increase in the number of observations, but mostly because of the increase in the number of variables associated with each observation. The dimension of the data is the number of variables that are measured on each observation.

High-dimensional datasets present many mathematical challenges as well as some opportunities, and are bound to give rise to new theoretical developments [11]. One of the problems with high-dimensional datasets is that, in many cases, not all the measured variables are "important" for understanding the underlying phenomena of interest. While certain computationally expensive novel methods [4] can construct predictive models with high accuracy from high-dimensional data, it is still of interest in many applications to reduce the dimension of the original data prior to any modeling of the data.

In mathematical terms, the problem we investigate can be stated as follows: given the p-dimensional random variable $\mathbf{x} = (x_1, \ldots, x_p)^T$, find a lower dimensional representation of it, $\mathbf{s} = (s_1, \ldots, s_k)^T$ with $k \leq p$, that captures the content in the original data, according to some criterion. The components of $\mathbf{s}$ are sometimes called the hidden components. Different fields use different names for the p multivariate vectors: the term "variable" is mostly used in statistics, while "feature" and "attribute" are alternatives commonly used in the computer science and machine learning literature.

Throughout this paper, we assume that we have n observations, each being a realization of the p-dimensional random variable $\mathbf{x} = (x_1, \ldots, x_p)^T$ with mean $\mathrm{E}(\mathbf{x}) = \boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^T$ and covariance matrix $\mathrm{E}\{(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\} = \boldsymbol{\Sigma}_{p \times p}$. We denote such an observation matrix by $\mathbf{X} = \{x_{i,j} : 1 \leq i \leq p, 1 \leq j \leq n\}$. If μ_i and $\sigma_i = \sqrt{\boldsymbol{\Sigma}_{(i,i)}}$ denote the mean and the standard deviation of the ith random variable, respectively, then we will often standardize the observations $x_{i,j}$ by $(x_{i,j} - \hat{\mu}_i)/\hat{\sigma}_i$, where $\hat{\mu}_i = \bar{x}_i = 1/n \sum_{j=1}^{n} x_{i,j}$, and $\hat{\sigma}_i = 1/n \sum_{j=1}^{n} (x_{i,j} - \bar{x}_i)^2$.

We distinguish two major types of dimension reduction methods: linear and non-linear. Linear techniques result in each of the $k \leq p$ components of the new variable being a linear combination of the original variables:

$$s_i = w_{i,1} x_1 + \ldots w_{i,p} x_p, \quad \text{for} \quad i = 1, \ldots, k, \quad \text{or} \tag{1}$$

$$\mathbf{s} = \mathbf{W}\mathbf{x}, \tag{2}$$

where $\mathbf{W}_{k \times p}$ is the linear transformation weight matrix. Expressing the same relationship as

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{3}$$

with $\mathbf{A}_{p\times k}$, we note that the new variables $\mathbf{s}$ are also called the hidden or the latent variables. In terms of an $n \times p$ observation matrix $\mathbf{X}$, we have

$$S_{i,j} = w_{i,1}X_{1,j} + \ldots w_{i,p}X_{p,j}, \quad \text{for} \quad i = 1, \ldots, k, \quad \text{and} \quad j = 1, \ldots, n, \tag{4}$$

where j indicates the jth realization, or, equivalently,

$$\mathbf{S}_{k\times n} = \mathbf{W}_{k\times p}\mathbf{X}_{p\times n}, \tag{5}$$

$$\mathbf{X}_{p\times n} = \mathbf{A}_{p\times k}\mathbf{S}_{k\times n}. \tag{6}$$

Such linear techniques are simpler and easier to implement than more recent methods considering non-linear transforms.

In this paper, we review traditional and current state-of-the-art dimension reduction methods published in the statistics, signal processing and machine learning literature. There are numerous books and articles [41, 17, 5, 14, 19, 46, 13] in the statistical literature on techniques for analyzing multivariate datasets. Advances in computer science, machine learning [43, 50, 44, 2]. Earlier survey papers. [7] reviews several methods, including principal components analysis, projection pursuit, principal curves, self-organizing maps, as well as provides neural network implementations of some of the reviewed statistical models. [22] surveys recent results in independent component analysis, in the context of other dimension reduction methods.

This survey is organized as follows. Sections 2 and 3 review principal component analysis and factor analysis, respectively, the two most widely used linear dimension reduction methods based on second-order statistics. For normal variables (with mean zero), the covariance matrix contains all the information about the data. Second-order methods are relatively simple to code, as they require classical matrix manipulations. However, many datasets of interest are not realizations from Gaussian distributions. For those cases, higher-order dimension reduction methods, using information not contained in the covariance matrix, are more appropriate. Such a linear higher-order method, projection pursuit is reviewed in Section 4. Section 5 summarizes another higher-order linear method called independent component analysis. Although non-linear principal component analysis can be considered as a special case of independent component analysis, Section 5.1.4, it is reviewed separately in Section 6. It uses non-linear objective functions to determine the optimal weights, but the resulting components are still linear combinations of the original variables. Section 7 explains the method of random projections. Section 8 presents some extensions and non-linear dimension reduction techniques.

2 Principal component analysis

Principal component analysis (PCA) is the best, in the mean-square error sense, linear dimension reduction technique [25, 28]. Being based on the covariance matrix of the variables, it is a second-order method. In various fields, it is also known as the singular value decomposition (SVD), the Karhunen-Loève transform, the Hotelling transform, and the empirical orthogonal function (EOF) method.

In essence, PCA seeks to reduce the dimension of the data by finding a few orthogonal linear combinations (the PCs) of the original variables with the largest variance. The first PC, s_1, is the linear combination with the largest variance. We have $s_1 = \mathbf{x}^T\mathbf{w}_1$, where the p-dimensional coefficient vector $\mathbf{w}_1 = (w_{1,1}, \ldots, w_{1,p})^T$ solves

$$\mathbf{w}_1 = \arg\max_{||\mathbf{w}=1||}\mathrm{Var}\{\mathbf{x}^T\mathbf{w}\}. \tag{7}$$

The second PC is the linear combination with the second largest variance and orthogonal to the first PC, and so on. There are as many PCs as the number of the original variables. For many datasets, the first several PCs explain most of the variance, so that the rest can be disregarded with minimal loss of information.

Since the variance depends on the scale of the variables, it is customary to first standardize each variable to have mean zero and standard deviation one. After the standardization, the original variables with possibly different units of measurement are all in comparable units. Assuming a standardized data with the empirical covariance matrix

$$\mathbf{\Sigma}_{p\times p} = \frac{1}{n}\mathbf{X}\mathbf{X}^T, \tag{8}$$

we can use the spectral decomposition theorem to write $\mathbf{\Sigma}$ as

$$\mathbf{\Sigma} = \mathbf{U}\Lambda\mathbf{U}^T, \tag{9}$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_p)$ is the diagonal matrix of the ordered eigenvalues $\lambda_1 \leq \ldots \leq \lambda_p$, and $\mathbf{U}$ is a $p \times p$ orthogonal matrix containing the eigenvectors. It can be shown [41] that the PCs are given by the p rows of the $p \times n$ matrix $\mathbf{S}$, where

$$\mathbf{S} = \mathbf{U}^T\mathbf{X}. \tag{10}$$

Comparing (10) to (5), we see that the weight matrix $\mathbf{W}$ is given by $\mathbf{U}^T$. It can be shown [41] that the subspace spanned by the first k eigenvectors has the smallest mean square deviation from $\mathbf{X}$ among all subspaces of dimension k.

As briefly indicated in Section 8.5, PCs can also be obtained by using neural networks with specific architectures and learning algorithms.

Another property of the eigenvalue decomposition is that the total variation is equal to the sum of the eigenvalues of the covariance matrix,

$$\sum_{i=1}^{p} \text{Var}(\text{PC}_\text{i}) = \sum_{i=1}^{p} \lambda_i = \sum_{i=1}^{p} \text{trace}(\mathbf{\Sigma}), \tag{11}$$

and that the fraction

$$\sum_{i=1}^{k} \lambda_i / \text{trace}(\mathbf{\Sigma}) \tag{12}$$

gives the cumulative proportion of the variance explained by the first k PCs. By plotting the cumulative proportions in (12) as a function of k, one can select the appropriate number of PCs to keep in order to explain a given percentage of the overall variation. Such plots are called scree diagram plots in the statistical literature [53]. The number of PCs to keep can also be determined by first fixing a threshold λ_0, then only keeping the eigenvectors such that their corresponding eigenvalues are greater than λ_0. This latter method was found preferable in [26, 27], where the author also suggested keeping at least four variables.

The interpretation of the PCs can be difficult at times. Although they are uncorrelated variables constructed as linear combinations of the original variables, and have some desirable properties, they do not necessarily correspond to meaningful physical quantities. In some cases, such loss of interpretability is not satisfactory to the domain scientists.

An alternative way to reduce the dimension of a dataset using PCA is suggested in [41]. Instead of using the PCs as the new variables, this method uses the information in the PCs to find important variables in the original dataset. As before, one first calculates the PCs, then studies the scree plot to determine the number k of important variables to keep. Next, one considers the eigenvector corresponding to the smallest eigenvalue (the least important PC), and discards the variable that has the largest (absolute value) coefficient in that vector. Then, one considers the eigenvector corresponding to the second smallest eigenvalue, and discards the variable contributing the largest (absolute value) coefficient to that eigenvector, among the variables not discarded earlier. The process is repeated until only k variables remain.

3 Factor analysis

This section follows [41]. Like PCA, factor analysis (FA) is also a linear method, based on the second-order data summaries. First suggested by psychologists, FA assumes that the measured variables depend on some unknown, and often unmeasurable, common factors. Typical examples include variables defined as various test scores of individuals, as such scores are thought to be related to a common "intelligence" factor. The goal of FA is to uncover such relations, and thus can be used to reduce the dimension of datasets following the factor model.

The zero-mean p-dimensional random vector $\mathbf{x}_{p\times 1}$ with covariance matrix $\mathbf{\Sigma}$ satisfies the k-factor model if

$$\mathbf{x} = \Lambda\mathbf{f} + \mathbf{u}, \tag{13}$$

where $\Lambda_{p\times k}$ is a matrix of constants, and $\mathbf{f}_{k\times 1}$ and $\mathbf{u}_{p\times 1}$ are the random common factors and specific factors, respectively. In addition, the factors are all uncorrelated and the common factors are standardized to have variance one:

$$\mathrm{E}(\mathbf{f}) = \mathbf{0}, \quad \mathrm{Var}(\mathbf{f}) = \mathbf{I}, \tag{14}$$

$$\mathrm{E}(\mathbf{u}) = \mathbf{0}, \quad \mathrm{Cov}(u_i, u_j) = 0 \quad \text{for} \quad i \neq j, \tag{15}$$

$$\mathrm{Cov}(\mathbf{f}, \mathbf{u}) = \mathbf{0}. \tag{16}$$

Under these assumptions, the diagonal covariance matrix of $\mathbf{u}$ can be written as $\mathrm{Cov}(\mathbf{u}) = \Psi = \mathrm{diag}(\psi_{11}, \cdots, \psi_{\mathrm{pp}})$.

If the data covariance matrix can be decomposed as

$$\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi}, \tag{17}$$

then it can be shown that the k-factor model holds. Since x_i can be written as

$$x_i = \sum_{j=1}^{k} \lambda_{ij} f_j + u_i, \quad i = 1, \ldots, p, \tag{18}$$

its variance may be decomposed as

$$\sigma_{ii} = \sum_{j=1}^{k} \lambda_{ij}^2 + \psi_{ii}, \tag{19}$$

where the first part $h_i^2 = \sum_{j=1}^{k} \lambda_{ij}^2$ is called the *communality* and represents the variance of x_i common to all variables, while the second part ψ_{ii} is called the *specific* or *unique* variance and it is the contribution in the variability of x_i due to its specific u_i part, not shared by the other variables. The term λ_{ij}^2 measures the magnitude of the dependence of x_i on the common factor f_j. If several variables x_i have high loadings λ_{ij} on a given factor f_j, the implication is that those variables measure the same unobservable quantity, and are therefore redundant.

Unlike PCA, the factor model does not depend on the scale of the variables. However, the factor model also holds for orthogonal rotations of the factors. Given the orthogonal matrix $\mathbf{G}$, given the model (13), the new model

$$\mathbf{x} = (\Lambda\mathbf{G})(\mathbf{G}^T\mathbf{f}) + \mathbf{u}, \tag{20}$$

also holds, with new factors $\mathbf{G}^T\mathbf{f}$ and corresponding loadings $\Lambda\mathbf{G}$. Therefore, the factors are generally rotated to satisfy some additional constraints, such as

$$\Lambda^T\Psi^{-1}\Lambda \quad \text{is diagonal,} \quad \text{or} \tag{21}$$

$$\Lambda^T\mathbf{D}^{-1}\Lambda \quad \text{is diagonal,} \quad \mathbf{D} = \mathrm{diag}(\sigma_{11}, \ldots, \sigma_{pp}), \tag{22}$$

where the diagonal elements are in decreasing order. There are techniques, such as the varimax method, to rotate the factors to obtain a parsimonious representation with few significantly non-zero loadings (i.e. sparse matrix $\mathbf{\Lambda}$). As explained in [13], ICA (see Section 5) can be thought of as another factor rotation method, where the goal is to find rotations that maximize certain independence criteria.

In many cases, a k-order factor model in (17) provides a better explanation for the data than the alternative full covariance model $\mathrm{Var}(\mathbf{x}) = \mathbf{\Sigma}$. In such cases, it is possible to derive parameter estimates $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{\Psi}}$.

Let $\bar{\mathbf{x}}$, $\mathbf{R}$, and $\mathbf{S}$ denote the sample mean, covariance matrix, and correlation matrix, respectively, of the observed data matrix $\mathbf{X}$. Then, starting with

$$\hat{\sigma}_{ii} = s_{ii}, \quad i = 1, \ldots, p, \tag{23}$$

and using

$$\hat{\mathbf{\Sigma}} = \hat{\mathbf{\Lambda}}\hat{\mathbf{\Lambda}}^T + \hat{\mathbf{\Psi}}, \tag{24}$$

we obtain

$$\hat{\sigma}_{ii} = \sum_{j=1}^{k} \hat{\lambda_{ij}^2} + \hat{\psi}_{ii}. \tag{25}$$

Two different possibilities to derive estimates $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{\Psi}}$ for the model parameters in (13)-(16) are detailed in Sections 3.1 and 3.2.

3.1 Principal factor analysis

Suppose the data is standardized, so that its covariance matrix is equal to the correlation matrix. To obtain estimates $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{\Psi}}$ for the standardized variables, first estimate $\hat{h}_i^2$ for $i = 1, \ldots, p$. Common estimates $\hat{h}_i^2$ include the square of the multiple correlation coefficients of the ith variable with all the other variables, and the largest correlation coefficient between the ith variable and one of the other variables. Next, form the *reduced correlation matrix* $\mathbf{R} - \hat{\mathbf{\Psi}}$, where the diagonal elements of 1 in $\mathbf{R}$ are replaced by the elements $\hat{h}_i^2 = 1 - \hat{\psi}_{ii}$. Then, decompose the reduced correlation matrix in terms of the eigenvalues $a_1 \geq \ldots \geq a_p$ and orthonormal eigenvectors $\gamma_{(\mathbf{1})}, \ldots, \gamma_{(\mathbf{p})}$ as

$$\mathbf{R} - \hat{\mathbf{\Psi}} = \sum_{i=1}^{p} a_i \gamma_{(\mathbf{i})} \gamma_{(\mathbf{i})}^{\mathbf{T}} \tag{26}$$

If the first k eigenvalues are positive, estimate the ith column of $\mathbf{\Lambda}$ by

$$\hat{\lambda}_{(\mathbf{i})} = a_i^{1/2} \gamma_{(i)}, \quad i = 1, \ldots, k. \tag{27}$$

Equivalently,

$$\hat{\mathbf{\Lambda}} = \mathbf{\Gamma_1} \mathbf{A}_1^{1/2}, \tag{28}$$

where $\mathbf{\Gamma_1} = (\gamma_{(\mathbf{i})}, \ldots, \gamma_{(\mathbf{k})})$, and $\mathbf{A}_1 = \text{diag}(a_1, \ldots, a_k)$. The eigenvectors are orthogonal, so the constraint in (22) holds.

Finally, the specific variance estimates are updated as

$$\hat{\psi}_{ii} = 1 - \sum_{i=1}^{k} \hat{\lambda}_{ij}^2, \quad i = 1, \ldots, p. \tag{29}$$

The k-factor model is permissible if all the p terms in (29) are non-negative.

In practice, the number of factors may be determined by looking at the eigenvalues a_i of the reduced correlation matrix, and choosing k as the index where there is a sharp drop in the eigenvalue magnitudes.

As its name suggests, principal factor analysis (PFA) is related to principal component analysis. When the specific variances are all zero, $\mathbf{\Psi} = \mathbf{0}$, comparing Equations (17) and (26) to Section 2 indicates that PFA is equivalent to PCA.

3.2 Maximum likelihood factor analysis

If, in addition to the factor model specified in (13)-(16), we also assume that the factors $\mathbf{f}$ and $\mathbf{u}$ are distributed as multivariate normal variables, then parameters of the model can also be estimated by maximizing the likelihood. In such cases, one can also test the hypothesis that the k-factor model describes the data more accurately than the unconstrained variance model.

The log-likelihood function can be written as

$$l = -\frac{1}{2}\, n\, \log|2\pi\mathbf{\Sigma}| - \frac{1}{2}\, n\, \text{tr}\mathbf{\Sigma}^{-1}\mathbf{S}, \tag{30}$$

and the goal is to maximize it with respect to the parameters $\mathbf{\Lambda}$ and $\mathbf{\Psi}$, subject to the constraint in (22) on $\mathbf{\Lambda}$. Under the factor model, $\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi}$.

The optimization is carried out by noting that the function

$$F(\mathbf{\Lambda}, \mathbf{\Psi}) = F(\mathbf{\Lambda}, \mathbf{\Psi}; \mathbf{S}) = \mathrm{tr}\mathbf{\Sigma}^{-1}\mathbf{S} - \log|\mathbf{\Sigma}^{-1}\mathbf{S}| - p \tag{31}$$

is a linear function of the log-likelihood l, with a maximum in l corresponding to a minimum in F. Also, in terms of the arithmetic mean a and the geometric mean g of the eigenvalues of $\mathbf{\Sigma^{-1}S}$, we have

$$F = p(a - \log g - 1). \tag{32}$$

Minimizing $F(\mathbf{\Lambda}, \mathbf{\Psi})$ proceeds in two stages: first, the minimization over $\mathbf{\Lambda}$ for a fixed $\mathbf{\Psi}$ has an analytical solution, then, the minimization over $\mathbf{\Psi}$ is carried out numerically.

4 Projection pursuit

Projection pursuit (PP) is a linear method that, unlike PCA and FA, can incorporate higher than second-order information, and thus is useful for non-Gaussian datasets. It is more computationally intensive than second-order methods.

Given a projection index that defines the "interestingness" of a direction, PP looks for the directions that optimize that index. As the Gaussian distribution is the least interesting distribution (having the least structure), projection indices usually measure some aspect of non-Gaussianity. If, however, one uses the second-order maximum variance, subject that the projections be orthogonal, as the projection index, PP yields the familiar PCA. Writing the optimization criterion as

$$Q(\mathbf{x}, \mathbf{w}) = \mathrm{Var}\{\mathbf{x}^T\mathbf{w}\}, \tag{33}$$

according to (7), the direction $\mathbf{w}_1$ of the first PC solves $\arg\max_{||\mathbf{w}=1||} Q(\mathbf{x}, \mathbf{w})$, and the corresponding first PC is $s_1 = \mathbf{x}^T\mathbf{w}_1$.

A commonly used higher-order projection index is based on the negative Shannon entropy [20]. Given the random variable $\mathbf{x}$ with probability distribution f, its negative entropy is defined as

$$Q(\mathbf{x}) = \int f(\mathbf{x}) \log f(\mathbf{x}) d(\mathbf{x}). \tag{34}$$

The Gaussian distribution minimizes this measure, so it makes sense to find directions $\mathbf{w}$ that maximize the entropy of the projected data $Q(\mathbf{x}, \mathbf{w})$ with respect to $\mathbf{w}$, subject to having constant variance of $\mathbf{x}^T\mathbf{w}$.

Other projection indices include indices based on higher-order cumulants and on the Fisher information [7, 22]. However, all of these measures depend on the unknown probability distribution of $\mathbf{x}^T\mathbf{w}$, which can be difficult to estimate. Alternative indices based on approximations, and on different measures of non-normality have also been proposed in the literature [22].

The FastICA algorithm for independent components in Section 5.3 can also be used to find projection pursuit directions.

5 Independent component analysis

This section is based on [22], a recent survey on independent component analysis (ICA). More information (and software) on this currently very popular method can be found at various websites, including [6, 24, 49]. Books summarizing the recent advances in the theory and application of ICA include [1, 48, 15, 38].

ICA is a higher-order method that seeks linear projections, not necessarily orthogonal to each other, that are as nearly statistically independent as possible. Statistical independence is a much stronger condition than uncorrelatdness. While the latter only involves the second-order statistics, the former depends on all the higher-order statistics. Formally, the random variables $\mathbf{x} = \{x_1, \ldots, x_p\}$ are uncorrelated, if for $\forall i \neq j, 1 \leq i, j \leq p$, we have

$$\mathrm{Cov}(x_i, x_j) = \mathrm{E}\{(x_i - \mu_i)(x_j - \mu_j)\} = \mathrm{E}(x_i x_j) - \mathrm{E}(x_i)\mathrm{E}(x_j) = 0. \tag{35}$$

In contrast, independence requires that the multivariate probability density function factorizes, and can be written as

$$f(x_1, \ldots, x_p) = f_1(x_1) \ldots f_p(x_p). \tag{36}$$

Independence always implies uncorrelatedness, but not vice versa in general. Only if the distribution $f(x_1, \ldots, x_p)$ is multivariate normal, are the two equivalent. For Gaussian distributions, the PCs are independent components. Following [22], the noise-free ICA model for the p-dimensional random vector $\mathbf{x}$ seeks to estimate the components of the k-dimensional vector $\mathbf{s}$ and the $p \times k$ full column rank mixing matrix $\mathbf{A}$ in (3),

$$(x_1, \ldots, x_p)^T = \mathbf{A}_{p \times k}(s_1, \ldots, s_k)^T \tag{37}$$

such that the components of $\mathbf{s}$ are as independent as possible, according to some definition of independence. At least one of the hidden independent components s_i has to be non-Gaussian to ensure the identifiability of the model [22]. The noisy ICA contains an additive random noise component,

$$(x_1, \ldots, x_p)^T = \mathbf{A}_{p \times k}(s_1, \ldots, s_k)^T + (u_1, \ldots, u_p)^T \tag{38}$$

but estimation of such models is still an open research issue [22]. In this survey, we only consider the noiseless model as specified in (37).

There are overcomplete versions of ICA, where the number k of ICs is larger than the number of original variables p [22]. In this paper, we will assume that there are as many independent components as there are original variables, i.e. $k = p$. In contrast with PCA, the goal of ICA is not necessarily dimension reduction. To find $k < p$ independent components, one needs to first reduce the dimension of the original data p to k, by a method such as PCA.

As the problem is stated, there is no order among the ICs. Once they are estimated, they can be ordered according to the norms of the columns of the mixing matrix (similar to the ordering in PCA), or according to some non-Gaussianity measure (similar to ordering in PP).

ICA can be considered a generalization of the PCA and the PP concepts. While PCA seeks uncorrelated variables, ICA seeks independent variables. The noise-free ICA is a special case of PP, with independence being the "interestingness" in the projection pursuit index definition. The noisy ICA model is equivalent to the FA model in (13) assuming non-Gaussian data.

ICA has been applied to many different problems, including exploratory data analysis, blind source separation, blind deconvolution, and feature extraction. In the feature extraction context, the columns of the matrix $\mathbf{A}$ represent features in the data, and the components s_i give the coefficient of the ith feature in the data. Several authors used ICA to extract meaningful features from natural images [22].

Estimation of the model in (37) consists of two steps: specifying the objective function (also called the contrast, the loss function, the cost function), and the algorithm to optimize the objective function. Objective functions can be categorized into two groups: "multi-unit" contrast functions that estimate all p independent components at once, and "one-unit" contrast functions that estimate a single independent component at a time [22]. They are detailed in Section 5.1 and in Section 5.2, respectively. Section 5.3 lists several optimization algorithms.

5.1 Multi-unit objective functions

There are many different ways to specify objective functions. This section lists several possibilities. It has been shown, that despite their different formulations, they all closely related, and under certain conditions, some are equivalent [22].

Under certain conditions (the distribution of the independent components is known with sufficient accuracy), the mutual information method is essentially equivalent to maximum likelihood principle, and so is the non-linear PCA method. Under the same conditions, cumulant-based methods are approximations to the mutual information.

Cumulant and general contrast-based methods, however, can be used for any non-Gaussian data, without knowing the underlying distributions.

5.1.1 Maximum likelihood and network entropy

This method specifies the likelihood of the noise-free ICA model, and uses the maximum likelihood principle to estimate the parameters. Under some conditions, it is equivalent to the "infomax" network entropy maximization concept in the neural network literature.

The advantages of this method include the asymptotic efficiency of maximum likelihood estimates under regularity conditions. However, it requires knowledge of the distribution of the independent components, it is sensitive to outliers, and it is computationally intensive, which make it undesirable in many practical situations.

5.1.2 Mutual information and Kullback-Leibler divergence

Mutual information I measures the dependence among m random variables y_i as

$$I(y_1, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y}), \tag{39}$$

where H is the differential entropy, $H(\mathbf{y}) = -Q(\mathbf{y})$ in (34). The mutual information is always non-negative, and is zero if and only if the variables are statistically independent. It therefore makes sense to find the variables that minimize the mutual information among the components.

Mutual information is also equal to the Kullback-Leibler divergence

$$\delta(f_1, f_2) = \int f_1(\mathbf{y}) \log \frac{f_1(\mathbf{y})}{f_2(\mathbf{y})} d\mathbf{y} \tag{40}$$

between the joint density $f(\mathbf{y})$ and the factorized $\tilde{f}(\mathbf{y}) = f_1(y_1) \ldots f_m(y_m)$. The Kullback-Leibler divergence measures a "closeness" of two distributions. If the components are independent, the actual density $f(\mathbf{y})$ factorizes just like $\tilde{f}(\mathbf{y})$, and results in zero divergence.

Mutual information is hard to estimate, imposing difficulties on using it as an objective function. As summarized in [22], several approximations, based on polynomials, on higher-order cumulants, and on the maximum entropy principle, have been proposed.

5.1.3 Non-linear cross-correlations

This principle is based on canceling non-linear cross-correlations of the form $\mathrm{E}\{g_1(y_i)g_2(y_j)\}$, where g_1 and g_2 are non-linearities specified by the user. Assuming that y_i and y_j are independent, such cross-correlations are zero. Oftentimes, there are no explicit objective functions associated with the chosen cross-correlation, so that they are only implicitly specified.

5.1.4 Non-Linear PCA

This method indicates the strong connection between ICA and non-linear PCA. By introducing non-linearities g based on the probability densities of the independent components into the PCA objective function in (7), we obtain the ICA model

$$\mathbf{w}_1 = \arg\max_{||\mathbf{w}=1||} \mathrm{Var}\{g(\mathbf{x}^T\mathbf{w})\}. \tag{41}$$

As with the non-linear cross-correlation method, there might not exist explicit contrast functions.

5.1.5 Higher-order cumulant tensors

The ICA model can also be estimated by solving for the eigenvectors of eigenmatrices corresponding to the linear operator T defined by the fourth-order cumulant as

$$T(\mathbf{K}_{ij}) = \sum_{k,l} \mathrm{cum}(x_i, x_j, x_k, x_l)\mathbf{K}_{kl}. \tag{42}$$

The linear operator T maps the space of $k \times k$ matrices to itself, and has k^2 eigenvalues corresponding to eigenmatrices. This procedure does not need to know the probability densities of the independent components, but suffers from suboptimal statistical properties characteristic to cumulant-based estimators.

5.2 One-unit objective functions

One-unit contrast functions seek a single vector $\mathbf{w}$ such that the linear combination $\mathbf{x}^T\mathbf{w}$ is equal to one of the independent components s_i. It is desirable when not all the PCs are needed, it can be used iteratively to find more PCs, and it tends to result in computationally simple solutions.

The contrast functions in this section are closely related. Both cumulants and general contrast functions can be used to approximate the negentropy. The objective function based on the kurtosis (fourth-order cumulant) is a special case of general contrast functions.

5.2.1 Negentropy

Differential entropy is not invariant under scale transformations. The negentropy, or negative normalized entropy

$$J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y}) \tag{43}$$

where H is the differential entropy, $H(\mathbf{y}) = -Q(\mathbf{y})$ in (34), and $\mathbf{y}_{\text{gauss}}$ is a Gaussian random vector with the same covariance matrix as $\mathbf{y}$, is a linearly invariant version of the entropy. It is non-negative, and zero if and only if $\mathbf{y}$ is Gaussian. Finding the direction of maximum negentropy is equivalent to finding the representation with minimum mutual information. The directions that maximize the negentropy can also be found by using differential entropy as a projection index in PP.

Negentropy is difficult to estimate. Approximations based on higher-order cumulants are explained in 5.2.2, and ones based on general contrast functions in 5.2.3.

5.2.2 Higher-order cumulants

One higher-order cumulant often used as a measure of non-Gaussianity is the fourth-order cumulant, also called the kurtosis. By definition, the kurtosis $\text{kurt}(x)$ of a random variable x is

$$\text{kurt}(x) = \text{E}(x^4) - 3[\text{E}(x^2)]^2. \tag{44}$$

The kurtosis is zero for a Gaussian variable, it is positive for heavy-tailed super-Gaussian distributions, and it is negative for light-tailed sub-Gaussian distributions. Independent components can be derived by maximizing the modulus of the kurtosis.

Cumulant-based estimators can be poor in terms of robustness and asymptotic variance. They only consider the tails of the distribution, and are sensitive to outliers.

5.2.3 General contrast functions

In contrast with the contrast functions introduced earlier, general contrast functions are formulated to have good statistical properties without requiring knowledge of the distributions, and to allow simple interpretation and algorithmic implementation. Such contrast functions J measure non-Gaussianity of the standardized random variable y by comparing it to a standard Gaussian variable ν via a smooth non-quadratic even function G by

$$J_G(y) = |\text{E}_y[G(y)] - \text{E}_\nu[G(\nu)]|^p, \tag{45}$$

where p is usually taken to be 1 or 2. Taking $G(y) = y^4$, J_G is simply the kurtosis. For suitable choices of G, such as

$$G(y) = \log\cosh(a_1 u) \quad \text{or} \quad G(y) = \exp(-a_2 u^2/2), \tag{46}$$

with constants $a_1, a_2 \geq 1$, estimators based on optimizing generalized contrast functions have superior statistical properties than cumulant-based estimators. Being the log-density of a super-Gaussian distribution, G_1 is related to maximum likelihood estimation.

5.3 Optimization algorithms

Most optimization algorithms either require that the data be sphered, or they converge better for sphered data. Sphering is a linear transformation that maps $\mathbf{x}$ into a new variable $\mathbf{v}$ with unit covariance matrix:

$$\mathbf{v} = \mathbf{Q}\mathbf{x}, \quad \mathrm{E}(\mathbf{v}\mathbf{v}^T) = \mathbf{I}. \tag{47}$$

In terms of $\mathbf{v}$, the ICA model in (37) can be written as

$$\mathbf{v} = \mathbf{B}\mathbf{s}. \tag{48}$$

Assuming unit-variance independent components, we have $\mathbf{I} = \mathrm{E}(\mathbf{v}\mathbf{v}^T) = \mathbf{B}\mathrm{E}(\mathbf{s}\mathbf{s}^T)\mathbf{B}^T = \mathbf{B}\mathbf{B}^T$, and therefore $\mathbf{B}$ is orthogonal. The problem then translates to finding an appropriate orthogonal matrix $\mathbf{B}$ from the sphered $\mathbf{v}$. Once such a $\mathbf{B}$ is found, the independent components are obtained via

$$\hat{\mathbf{s}} = \mathbf{B}^T\mathbf{v}. \tag{49}$$

Several algorithms have been proposed to estimate independent components. As [22] summarizes, there are two major types: adaptive and batch-mode (block) algorithms.

Adaptive methods use stochastic gradient-type algorithms. Likelihood or other multi-unit contrast functions are optimized using gradient ascent of the objective function. One-unit implementations use straightforward stochastic gradient methods that optimize negentropy or approximations of it.

Examples of adaptive algorithms include the Jutten-Herault algorithm, which is based on canceling non-linear cross-correlations and converges only under harsh restrictions; other algorithms based on non-linear decorrelations that are more stable and computationally tractable than the Jutten-Herault method; algorithms for maximum likelihood estimation; non-linear PCA algorithms; neural one-unit learning rules; and exploratory projection pursuit algorithms.

Batch-mode algorithms are much more computationally efficient than adaptive algorithms, and are more desirable in many practical situations where there is no need for adaptation. The FastICA is such a batch-mode algorithm using fixed-point iteration. It was introduced in [23] using the kurtosis, but was subsequently extended to general contrast functions in [21]. A MATLAB implementation is available from [24]. It can also be used for projection pursuit analysis described in Section 4.

6 Non-linear principal component analysis

Non-linear PCA introduces non-linearity in the objective function, but the resulting components are still linear combinations of the original variables. This method can also be thought of as a special case of independent component analysis, Section 5.1.4. As indicated in [31], there are different formulations of the non-linear PCA.

A non-linear PCA criterion for the data vector $\mathbf{x} = (x_1, \ldots, x_p)^T$ searches for the components $\mathbf{s} = (s_1, \ldots, s_p)^T$ in the form of $\mathbf{s} = \mathbf{W}^T\mathbf{x}$ by minimizing

$$J(\mathbf{W}) = E\{||\mathbf{x} - \mathbf{W}\mathbf{g}(\mathbf{W}^T\mathbf{x})||^2\} \tag{50}$$

with respect to the $p \times p$ weight matrix $\mathbf{W}$ [31], where $\mathbf{g}(\mathbf{y})$ denotes the component-wise application of the non-linear function $g()$ to the elements of the vector $\mathbf{y}$. Commonly used such non-linear functions are odd functions like $g(y) = \tanh(y)$ and $g(y) = y^3$.

The optimization in (50) can be carried out either by the stochastic gradient descent algorithm with the learning parameter c and the $\Delta\mathbf{W}$ update matrix of $\mathbf{W}$ below,

$$\Delta\mathbf{W} = c[\mathbf{x} - \mathbf{W}\mathbf{g}(\mathbf{W}^T\mathbf{x})]\mathbf{g}(\mathbf{x}^T\mathbf{W}), \tag{51}$$

or by an approximate recursive least squares (RLS) algorithm [31]. The RLS method converges faster than the corresponding gradient descent method, has good final accuracy, but slightly higher computational load.

Before applying the algorithms, the data needs to be pre-whitened by $\mathbf{v} = \mathbf{V}\mathbf{x}$, where $E\{\mathbf{v}\mathbf{v}^T\} = \mathbf{I}$. By denoting

$$\mathbf{y} = \mathbf{W}^T\mathbf{v} = \mathbf{W}^T\mathbf{V}\mathbf{x} = \mathbf{B}\mathbf{x}, \tag{52}$$

the optimization in (51) can be written as

$$\Delta \mathbf{W} = c[\mathbf{v} - \mathbf{W}\mathbf{g}(\mathbf{y})]\mathbf{g}(\mathbf{y}^T), \tag{53}$$

where, after convergence, $\mathbf{y}$ contains the sought $\mathbf{s}$ vector.

Assuming that the components of $\mathbf{s}$ all have variance equal to one, the final $\mathbf{y}$ estimates are standardized to have $E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{I}$, resulting in the matrix $\mathbf{W}$ being orthogonal ($E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{W}^T E\{\mathbf{v}\mathbf{v}^T\}\mathbf{W} = \mathbf{I}$). Under this condition, it can be shown [31] that

$$J(\mathbf{W}) = E\{||\mathbf{v} - \mathbf{W}\mathbf{g}(\mathbf{W}^T\mathbf{v})||^2\} = E\{||\mathbf{y} - \mathbf{g}(\mathbf{y})||^2\} = \sum_{i=1}^{p} E\{[y_i - g(y_i)]^2\}. \tag{54}$$

As indicated in Section 8.5, [37] proposed a neural network architecture with non-linear activation functions in the hidden layers to estimate non-linear PCAs.

7 Random projections

The method of random projections is a simple yet powerful dimension reduction technique that uses random projection matrices to project the data into lower dimensional spaces [47, 32, 33, 35]. The original data $\mathbf{X} \in \mathcal{R}^p$ is transformed to the lower dimensional $\mathbf{S} \in \mathcal{R}^k$, with $k \ll p$, via

$$\mathbf{S} = \mathbf{R}\mathbf{X}, \tag{55}$$

where the columns of $\mathbf{R}$ are realizations of independent and identically distributed (i.i.d.) zero-mean normal variables, scaled to have unit length. The method was proposed in the context of clustering text documents, where the initial dimension p can be on the order of 6000, and the final dimension k is still relatively large, on the order of 100. Under such circumstances, even PCA, the simplest alternative linear dimension reduction technique, can be computationally too expensive. Random projections are applied as a data pre-processing step, then, the resulting lower dimensional data is clustered. It has been shown empirically that results with the random projection method are comparable with results obtained with PCA, and take a fraction of the time PCA requires [33, 35]. To reduce the computational burden of the random projection method, at a slight loss in accuracy, the random normal projection matrix $\mathbf{R}$ may be replaced by thresholding its values to -1 and +1, or by matrices whose rows have a fixed number of 1s (at random locations) and the rest 0s [35].

If the similarity between two vectors is measured by their inner product (giving the cosine of their angle for unit-length vectors), [33] showed that if the dimension of the reduced space d is large, random projection matrices preserve the similarity measure: on the average, the distortion of the inner products is zero, and its variance is at most the inverse of twice d.

8 Non-linear methods and extensions

8.1 Non-linear independent component analysis

Non-linear methods, such as principal curves, self organizing maps and topographic maps, can, in principle, be incorporated into ICA.

Given the p-dimensional zero-mean non-Gaussian variable $\mathbf{x}$, the non-linear ICA model replaces the linear transformation in (3) by

$$(x_1, \ldots, x_p)^T = \mathbf{f}(s_1, \ldots, s_k)^T, \tag{56}$$

where $\mathbf{f}$ is an unknown real-valued p-component vector function. In general, $k = p$.

The identifiability and other properties of the general non-linear ICA model makes its estimation difficult. A few publications considering special cases are mentioned in [22]. An overview of the problem, along with a maximum likelihood and a Bayesian ensemble learning method for estimation can be found in [30].

8.2 Principal curves

Principal curves are smooth curves that pass through the "middle" of multidimensional data sets [18, 40, 7]. Linear principal curves are in fact principal components, while non-linear principal curves generalize the concept.

Given the p-dimensional random vector $\mathbf{y} = (y_1, \ldots, y_p)$ with density $\mathbf{g}(\mathbf{y})$, and the smooth curve $\mathbf{f}(s) = (f_1(\lambda), \ldots, f_p(\lambda)) \in \mathcal{R}^p$ parameterized by the real-valued λ, define the projection index $\lambda_{\mathbf{f}}(\mathbf{y})$ to be value of λ corresponding to the point on $\mathbf{f}(s)$ that is closest (in Euclidean distance) to $\mathbf{y}$. The set of principal curves is defined in [18] as the curves that do not intersect themselves and are self-consistent with respect to the data. By definition, a curve is self-consistent if each point $\mathbf{f}(\lambda)$ is the mean of all points in the support of $\mathbf{g}$ that are projected on λ.

$$E[\mathbf{y}|\lambda_{\mathbf{f}}(\mathbf{y}) = \lambda] = \mathbf{f}(\lambda). \tag{57}$$

It was shown in [18] that a curve $\mathbf{f}$ is a principal curve if and only if it solves

$$\min_{\mathbf{f}} \sum_{i=1}^{N} ||\mathbf{y}_i - \mathbf{f}(\lambda_{\mathbf{f}}(\mathbf{y}_i)||^2, \tag{58}$$

where $\mathbf{y}_i$ is the ith instance of the p-dimensional vector, and the composition of functions $\mathbf{f}(\lambda_{\mathbf{f}}(\mathbf{y}_i)$ gives the p-dimensional coordinates of the projection of $\mathbf{y}_i$ onto the curve $\mathbf{f}$.

To estimate $\mathbf{f}$ and λ, [18] proposed an iterative algorithm. It starts with $\mathbf{f}(\lambda) = E(\mathbf{y}) + \mathbf{d}\lambda$, where $\mathbf{d}$ is the first eigenvector of the covariance matrix of $\mathbf{y}$ and $\lambda = \lambda_{\mathbf{f}}(\mathbf{y})$. Then it iterates the two steps

1. For a fixed λ, minimize $E||\mathbf{y} - \mathbf{f}(\lambda)||^2$ by setting $\mathbf{f}_j(\lambda) = E(y_j|\lambda_{\mathbf{f}}(\mathbf{y}) = \lambda)$ for each j
2. Fix $\mathbf{f}$ and set $\lambda = \lambda_{\mathbf{f}}(\mathbf{y})$ for each $\mathbf{y}$

until the change in $E||\mathbf{y} - \mathbf{f}_\lambda||^2$ is less than a threshold.

An alternative formulation of the principal curves method, along with a generalized EM algorithm for its estimation under Gaussian distribution of $\mathbf{g}()$, is presented in [52].

In general, for the model $\mathbf{y} = \mathbf{f}(\lambda) + \boldsymbol{\epsilon}$, where $\mathbf{f}$ is smooth and $E(\boldsymbol{\epsilon}) = \mathbf{0}$, $\mathbf{f}$ is not necessarily a principal curve. Except for a few special cases, it is not known in general for what type of distributions do principal curves exist, how many principal curves there are, and what their properties are [7].

The concept of principal curves can be extended to higher dimensional principle surfaces, but the estimation procedure gets more complicated.

8.3 Multidimensional scaling

Given n items in a p-dimensional space and an $n \times n$ matrix of proximity measures among the items, multidimensional scaling (MDS) produces a k-dimensional, $k \leq p$, representation of the items such that the distances among the points in the new space reflect the proximities in the data [8, 7, 41]. The proximity measures the (dis)similarities among the items, and in general, it is a distance measure: the more similar two items are, the smaller their distance is. Popular distance measures include the Euclidean distance (L_2 norm), the manhattan distance (L_1, absolute norm), and the maximum norm. Results of MDS are indeterminate with respect to translation, rotation, and reflection.

MDS has been typically used to transform the data into two or three dimensions, and visualizing the result to uncover hidden structure in the data. A rule of thumb to determine the maximum number of k, is to ensure that there are at least twice as many pairs of items then the number of parameters to be estimated, resulting in $p \geq 4k + 1$ [7].

Given the n items $\{\mathbf{x_i}\}_{i=1}^n \in \mathcal{R}^p$ and a symmetric distance matrix $\Delta = \{\delta_{ij}, i, j = 1, \ldots, n\}$, the result of a k-dimensional MDS will be the set of points $\{\mathbf{y_i}\}_{i=1}^n \in \mathcal{R}^k$ such that the distances $d_{ij} = d(\mathbf{y_i}, \mathbf{y_j})$ are as close as possible to a function f of the corresponding proximities $f(\delta_{ij})$.

In [41], MDS methods that incorporate the given distances δ_{ij} into their calculations are called metric methods, while the ones that only use the rank ordering of the distances are called non-metric methods. In contrast, [7] states that depending on whether f is linear or non-linear, MDS is called either metric or non-metric, correspondingly.

Following [7], the steps for the most general estimation procedure are as follows. First, define the stress as an objective function to be minimized by f

$$\text{stress}_f(\Delta, \mathbf{X}, f) = \sqrt{\frac{\sum_{i,j}[f(\delta_{ij}) - d_{ij}]^2}{\text{scale factor}}}, \tag{59}$$

where the scale factor is usually based on $\sum_{i,j}[f(\delta_{ij})]^2$ or on $\sum_{i,j} d_{ij}^2$. Next, for a given $\mathbf{X} = \{\mathbf{x_i}\}_{i=1}^n$, find $\hat{f}$ that minimizes (59),

$$\text{stress}(\Delta, \mathbf{X}, \hat{f}) = \min_f \text{ stress}_f(\Delta, \mathbf{X}, f), \tag{60}$$

then determine the optimal $\hat{\mathbf{X}}$ by

$$\text{stress}(\Delta, \hat{\mathbf{X}}, \hat{f}) = \min_{\mathbf{X}} \text{ stress}(\Delta, \mathbf{X}, \hat{f}). \tag{61}$$

The special case of using Euclidean distance and f as the identity in (59) leads to the principal coordinates of $\mathbf{X}$ in k dimensions as the solution, which are equivalent to the first k principal components of $\mathbf{X}$ (without re-scaling to correlations) [41].

An alternative to MDS is FastMap [12], a computationally efficient algorithm that maps high-dimensional data into lower-dimensional spaces, while preserving distances between objects.

8.4 Topologically continuous maps

There are several methods based on finding a continuous map to transform a high-dimensional data into a lower-dimensional lattice (latent) space of fixed dimension [7]. Such techniques could be called self-organizing maps, but that name is most often associated with one particular such method, namely, Kohonen's self-organizing maps. To avoid confusion, we follow the review [7], and refer to these methods collectively as methods that use topologically continuous maps.

8.4.1 Kohonen's self-organizing maps

Given the data vector $\{\mathbf{t_n}\}_{n=1}^N \in \mathcal{R}^D$, Kohonen's self-organizing maps (KSOM) [36] learn, in an unsupervised way, a map between the data space and a 2-dimensional lattice. The method can be extended to L-dimensional topological arrangements as well. Let d_L and d_D denote distances (typically Euclidean) in the lattice and in the data space, respectively, and define a neighborhood function h_{ij} on the lattice space, such that it is symmetric, has values in the $[0, 1]$ interval, $h_{ii} = 1$ for any node i in the lattice, and the further node j is from i in the lattice, the smaller h_{ij} is. The neighborhood of node i consists of the nodes for which h_{ij} greater than a threshold. Typical neighborhood function is $h_{ij} = \exp(-d_L^2(i, j)/2/\sigma^2)$, where σ specifies the range of the neighborhood.

Kohonen's rule uses an initially random set of reference vectors $\{\boldsymbol{\mu}_i\}_{i=1}^m$ in the data space $\mathcal{R}^p$, then updates them iteratively according to the data distribution such that the final reference vector will be dense in regions of $\mathcal{R}^D$ where the data is common. Kohonen's rule iterates the following procedure over all the data points until convergence occurs.

- For a given data $\mathbf{t}_n$, find the closest vector $\boldsymbol{\mu}_{i^*}$ to it in the lattice space:

$$i^* = \arg\max_{j \in \text{lattice}} d_D(\boldsymbol{\mu}_j, \mathbf{t}_n), \tag{62}$$

- Then, at iteration t and learning rate $\alpha^{(t)} \in [0, 1]$, update the reference vector by moving it a distance $\rho = \alpha^{(t)} h_{i^* i}^{(t)}$ towards t:

$$\boldsymbol{\mu}_i^{\text{new}} = \boldsymbol{\mu}_i^{\text{old}} + \alpha^{(t)} h_{i^* i}^{(t)} (\mathbf{t}_n - \boldsymbol{\mu}_i^{\text{old}}) = (1 - \rho)\boldsymbol{\mu}_i^{\text{old}} + \rho \mathbf{t}_n. \tag{63}$$

Although KSOMs are useful in many applications, they have several drawbacks: there is no implicit criteria that they try to optimize, there are no rules to optimally select $\alpha^{(t)}$ and $h^{(t)}$, and there is no proof that they converge in general.

8.4.2 Density networks

Density networks [7] assume a probability distribution for the data given the parameters, as well as prior distributions for the parameters, then apply Bayesian learning techniques to model the data in terms of latent variables.

Generative topographic mapping (GTM) is a special density network based on constrained Gaussian mixtures that uses the expectation-maximization (EM) algorithm to estimate the parameters by maximizing the likelihood function. It was introduced in [3], and, unlike the KSOMs in Section 8.4.1, it provides a rigorous treatment of SOMs under certain assumptions.

8.5 Neural networks

Neural networks (NNs) model the set of output variables $\{y_j\}_{j=1}^{d}$ in terms of the input variables $\mathbf{x} = \{x_i\}_{i=1}^{p}$ as

$$y_j = y_j(\mathbf{x}, \mathbf{w}), \tag{64}$$

where the functions $y_j(\mathbf{x}, \mathbf{w})$ specify the network architecture, and the weights $\mathbf{w}$ are determined by training (learning) the NN using a set of known examples and an error function [2]. Many, traditional and more recent, linear and non-linear, dimension reduction techniques can be implemented using neural networks with different architectures and learning algorithms [2, 46, 40, 51, 7].

The simplest NN has three layers: the input layer, one hidden (bottleneck) layer, and the output layer. First, to obtain the data at node h of the hidden layer, the inputs x_i are combined through weights w_{ih} along with a threshold (bias) term α_h, then they are passed through the corresponding activation function ϕ_h. In the second step, the output is obtained in a similar way from the data on the hidden nodes, using the weights w_{hj}, the threshold α_j, and a possibly different output function ϕ_o:

$$y_j = \phi_o \left(\alpha_j + \sum_h w_{hj} \phi_h (\alpha_h + \sum_i w_{ih} x_i) \right). \tag{65}$$

The first part of the network reduces the input data into a lower-dimensional space, while the second decodes the reduced data into the original domain. Frequently used activation and output functions include the linear (identity) function, sigmoidal (S-shaped) functions such as the logistic function, and the Heaviside thresholding function [2, 53]. NNs with a single hidden layer networks and the threshold activation function are also called perceptrons. Networks that try to learn the identity mapping, i.e. the outputs y_j are identical to the inputs x_i, are called auto-associative (auto-encoders, bottlenecks, p-k-p networks). Hetero-associative neural nets have different number of input- and output layers, and are used, for example, in classification.

As summarized in [7], there are many types of NN architectures that can extract principal components. More complete details can be found in [9]. For example, a linear, one hidden layer auto-associative perceptron with p input units, $k < p$ hidden units, and p output units, can be trained with back-propagation to find a basis of the subspace spanned by the first k PCs, if the error metric used is the minimum squared sum of differences between the input and the output units. Other networks, based on Oja's rule and various de-correlating devices can also be used to find principal components.

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 principal components. Starting from the feed-forward neural network implementation of PCA [40, 7], [37] extended the idea to include non-linear activation functions in the hidden layers. In this framework, the non-linear PCA network can be thought of as an auto-associative neural network with five layers: input (1), hidden (2), bottleneck (3), hidden (4), and output (5). If $\lambda_{\mathbf{f}} : \mathcal{R}^p \rightarrow \mathcal{R}^k$ denotes the function modeled by layers (1), (2), and, (3), and $\mathbf{f} : \mathcal{R}^k \rightarrow \mathcal{R}^p$ denotes the function modeled by layers (3), (4), and (5), it can be shown [40] that the weights of the non-linear PCA network are determined such that

$$\min_{\mathbf{f}, s_{\mathbf{f}}} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{f}(\lambda_{\mathbf{f}}(\mathbf{x}_i))||^2, \tag{66}$$

where $\mathbf{x}_i$ denotes the ith instance of the p-dimensional vector $\mathbf{x}$. Note the close connection to principal surfaces (58) in Section 8.2. Both lead to PCA in case of linear $s_{\mathbf{f}}$ and $\mathbf{f}$.

The thesis of [51] compares principle component analysis, vector quantization, and five layer neural networks, for reducing the dimension of images. It also provides a C software package called NeuralCam implementing those methods.

8.6 Vector quantization

As explained in [51], [29] introduced a hybrid non-linear dimension reduction technique based on combining vector quantization for first clustering the data, then applying local PCA on the resulting Voronoi cell clusters. On the image data set used in [51], both non-linear techniques (vector quantization, VQ, and non-linear PCA using five layer neural network implementation, NLPCA) outperformed the linear PCA. Among the non-linear techniques, VQ achieved better results than NLPCA.

8.7 Genetic and evolutionary algorithms

Genetic and evolutionary algorithms (GEAs) are optimization techniques based on Darwinian theory of evolution that use natural selection and genetics to find the best solution among members of a competing population [16]. There are many references describing how GEAs can be used in dimension reduction. In essence, given a set of candidate solutions, an objective function to evaluate the fitness of candidates, and the values for the parameters of the chosen algorithm, GEAs search the candidate space for the member with the optimal fitness. For example, [45] use GAs in combination with a k-nearest neighbor (knn) classifier to reduce the dimension of a feature set: starting with a population of random transformation matrices $\{\mathbf{W}_{k\times p}\}^{(i)}$, they use GAs to find the transformation $\mathbf{W}_{k\times p}$ such that the knn classifier using the new features $\mathbf{S}_{k\times n} = \mathbf{W}_{k\times p}\mathbf{X}_{p\times n}$ classifies the training data most accurately.

8.8 Regression

Regression methods can be used for dimension reduction when the goal is to model a response variable $\mathbf{y}$ in terms of a set of $\mathbf{x}_i$ variables. The regression function can be linear, or non-linear. Traditionally, the $\mathbf{x}_i$ variables have been called the independent, or explanatory variables in statistics, while $\mathbf{y}$ was the response, or the dependent variable. In this regression context, it is generally assumed that the $\mathbf{x}_i$s were carefully selected, uncorrelated, and relevant to explaining the variation in $\mathbf{y}$. In current data mining applications, however, those assumptions rarely hold. Variable selection, or dimension reduction, is therefore needed for such cases.

A well-known statistical variable selection method is step-wise regression, where different models are fit using different subsets of the explanatory variables. The results are then compared by calculating various goodness-of-fit measures, and the subset with the best measure is chosen as the explanatory variables with the reduced dimension. A similar approach, selecting the most relevant features by evaluating random subsets of the original features, is called the wrapper method in the machine learning community [34].

Dimension reduction methods related to regression include projection pursuit regression [20, 7], generalized linear [42, 10] and additive [19] models, neural network models, and sliced inverse regression and principal hessian directions [39].

9 Summary

In this paper, we described several dimension reduction methods.

Acknowledgments

UCRL-ID-148494. This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

[1] Hyvärinen A., J. Karhunen, and E. Oja. *Independent Component Analysis*. Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. Wiley, 2001.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] C.M. Bishop, M. Svensen, and C.K.I. Williams. EM optimization of latent-variable density models. In Touretzky D.S., M.C. Mozer, and M.E. Hasselmo, editors, *Advances in Neural Information Processing Sytems*, volume 8. MIT Press, Cambridge, MA, 1996.

[4] L. Breiman. Random forests. Technical report, Department of Statistics, University of California, 2001.

[5] L.J. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, California, 1984.

[6] J.-F. Cardoso. ICA website of J.-F. Cardoso. http://www.tsi.enst.fr/∼cardoso/icacentral/.

[7] M.A. Carreira-Perpinan. A review of dimension reduction techniques. Technical report CS-96-09, Department of Computer Science, University of Sheffield, 1997.

[8] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman and Hall, second edition, 2001.

[9] K.I. Diamantaras and S.-Y. Kung. *Principal Component Neural Networks. Theory and Applicatins.* John Wiley & Sons, New York, Londin, Sydney, 1996.

[10] A.J. Dobson. *An introduction to generalized linear models*. Chapman and Hall, 1990.

[11] D.L. Donoho. 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, August 6-11. http://www-stat.stanford.edu/∼donoho/Lectures/AMS2000/AMS2000.html, 2000.

[12] C. Faloutsos and K.-I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In Michael J. Carey and Donovan A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174, San Jose, California, 1995.

[13] J. Friedman, T. Hastie, and R. Tibshirani. *Elements of Statistical Learning: Prediction, Inference and Data Mining*. Springer, 2001.

[14] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. San Diego: Academic Press, 2nd edition edition, 1990.

[15] M. Girolami, editor. *Advances in Independent Component Analysis*. Perspectives in Neural Computing. Springer, 2000.

[16] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading, MA, 1989.

[17] D.J. Hand. *Discrimination and Classification*. New York: John Wiley, 1981.

[18] T. Hastie and W. Stuetzle. Principal curves. *J. Am. Stat. Assoc.*, 84:502–516, 1989.

[19] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, 1990.

[20] P.J. Huber. Projection pursuit. *Ann. Stat.*, 13(2):435–475, 1985.

[21] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634,, 1999.

[22] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999. citeseer.nj.nec.com/hyv99survey.html.

[23] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[24] A. Hyvvärinen et al. ICA website at the Helsinki University of Technology. http://www.cis.hut.fi/∼aapo/.

[25] J.E. Jackson. *A User's Guide to Principal Components*. New York: John Wiley and Sons, 1991.

[26] I.T. Jolliffe. Discarding variables in principal component analysis I: artificial data. *Appl. Statist.*, 21:160–173, 1972.

[27] I.T. Jolliffe. Discarding variables in principal component analysis II: real data. *Appl. Statist.*, 22:21–31, 1973.

[28] I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[29] N. Kambhaltla and T. K. Leen. Fast non-linear dimension reduction. In *Advances in Neural Information Processing Systems*, pages 152–159. Morgan Kaufmann Publishers, Inc., 1994.

[30] J. Karhunen. Nonlinear independent component analysis. In S. Roberts and R. Everson, editors, *Independent Component Analaysis: Principles and Practice*. Cambridge University Press, Cambridge, UK, 2000.

[31] J. Karhunen, P. Pajunen, and E. Oja. The nonlinear pca criterion in blind source separation: Relations with other approaches. *Neurocomputing*, 22:5–20, 1998.

[32] S. Kaski. *Data exploration using self-organizing maps*. PhD thesis, Helsinki University of Technology, Finland, 1997.

[33] S. Kaski. Dimensionality reduction by random mapping: fast similarity computation for clustering. *Proc. IEEE International Joint Conference on Neural Networks*, 1:413–418, 1998.

[34] R. Kohavi and G. John. The wrapper approach. In H. Liu and H. Motoda, editors, *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Springer Verlag, 1998.

[35] T. Kohonen et al. Self organization of massive document collection. *IEEE Transactions on Neural Networks*, 11(3):574–585, May 2000.

[36] T.K. Kohonen. The self-organizing map. *Proc. IEEE*, 78:1484–1480, 1990.

[37] M.A. Kramer. Non-linear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243, 1991.

[38] T.-W. Lee. *Independent Component Analysis: Theory and Applications*. Kluwer Acedemic Publishers, 2001.

[39] K.-C. Li. High dimensional data analysis via the SIR/PHD approach. http://www.stat.ucla.edu/∼kcli/, April 2000. Lecture notes in progress.

[40] E. Malthouse. Some theoretical results on nonlinear principal component analysis. citeseer.nj.nec.com/malthouse96some.html, 1996.

[41] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Probability and Mathematical Statistics. Academic Press, 1995.

[42] P. McCullagh and J.A. Nelder. *Generalized linear models*. Chapman and Hall, 1989.

[43] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[44] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.

[45] M.L. Raymer et al. Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(2):164–171, July 2000.

[46] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[47] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.

[48] S. Roberts and R. Everson, editors. *Independent Component Analaysis: Principles and Practice*. Cambridge University Press, Cambridge, UK, 2000.

[49] T. Sejnowski et al. ICA website at The Salk Institute. http://www.cnl.salk.edu/∼tewon/ica_cnl.html.

[50] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

[51] J.A. Spierenburg. Dimension reduction of images using neural networks. Master's thesis, Leiden University, 1997.

[52] R. Tibshirani. Principal curves revisited. *Statistics and Computing*, 2:183–190, 1992.

[53] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S-Plus*. Springer, 1996.