SILESIAN UNIVERSITY OF TECHNOLOGY

Department for Strength of Materials and Computational Mechanics
Department of Fundamentals of Machinery Design

POLISH ASSOCIATION FOR COMPUTATIONAL MECHANICS

RECENT DEVELOPMENTS IN ARTIFICIAL INTELLIGENCE METHODS

Editors:
T. Burczyński, W. Cholewa, W. Moczulski

AI-METH Series, Gliwice, 2004

SUGGESTED REFERENCING:

```
@INPROCEEDINGS{key,
  author  =      "",
  title =        "",
  pages =        "",
  booktitle =    "{R}ecent {D}evelopments in {A}rtificial {I}ntelligence {M}ethods",
  editor =       "Burczy\'nski, T. and Cholewa, W. and Moczulski, W.",
  publisher =    "AI-METH Series",
  year =         "2004",
  address =      "Gliwice",
  month =        "November"
}
```

ISBN 83–914632–9–X

TYPESETTING

Marek Wyleżoł
(materials submitted by authors)

COVER DESIGN

Mirosław Dziewoński

PUBLISHERS

AI-METH Series issued by:

Department for Strength of Materials and Computational Mechanics, Silesian University of Technology
Department of Fundamentals of Machinery Design, Silesian University of Technology

http://www.ai-meth.polsl.pl

Printed in
Drukarnia cyfrowa D&D sp. z o.o. ul. Moniuszki 6, 44-100 Gliwice

AI-METH Series on Artificial Intelligence Methods

Aims:

Artificial Intelligence Methods have grown in power and diversity in recent years. The aim of this series is to provide a clear account of computational intelligence methods in mechanical, material, civil, biomedical and other engineering, computer science, optimization, management, ecology, etc. The scope of the series covers theoretical foundations, feasibility studies, comparative studies and practical applications of AI methods, and includes but not it is limited to:

- Appropriate descriptions and representations of problems and problem space,
- Representation, acquisition, verification and validation of knowledge,
- Imprecise, uncertain and incomplete as well as quantitative and qualitative data, models and knowledge,
- Data and knowledge granularity,
- Knowledge-base inference and/or search (solving) strategies,
- Symbolic and numerical computation based on concepts underlying biological processes,
- AI supported pattern recognition.

Scientific Board:

Contents

Adamska K.: *Description of SGA population with the use of continuous H^l "brightness" distribution*……..... 11

Bartelmus W., Zimroz R.: *Application of self-organised network for supporting condition evaluation of gearboxes*……..…….…… 17

Bartkowiak A., Cebrat S., Mackiewicz P.: *Probabilistic PCA and neural networks in search of representative features for some yeast genome data*…....…..... 21

Bartyś M. Kościelny J. M., Rzepiejewski P.: *Fuzzy logic application for fault isolation of authors*…....…… 27

Bąchór G., Moczulski W.: *Simulator of an intelligent walking minirobot operating autonomously in an unknown environment*……..... 31

Bednarski M.: *Example of diagnostic model identification with the use of learning Bayesian networks* …….. 37

Behroozi R., Katebi D. S.: *Using adaptive logic network for classification and testing the success of ART*….. 41

Beluch W., Burczyński T., Kuś W.: *Shape optimization of the cracked mechanical structures using boundary element method and distributed evolutionary algorithm*……..... 47

Bhavani S. D., Pujari A. K.: *Identifying subnetworks of interval algebra network*..... 53

Bielińska E., Sosnowski K.: *Computer database system for speaker recognition*..... 59

Burczyński T., Długosz A., Kuś W.: *Shape optimization of heat radiators using parallel evolutionary algorithms*..... 65

Burczyński T., Orantek P.: *Application of artificail neural network in computational sensitivity analysis*……..…… 69

Burczyński T., Poteralski A., Kuś W., Orantek P.: *Two different types on interpolation functions in optimization of 3-D structures using distributed and sequential evolutionary algorithm*..... 73

Burczyński T., Skrobol A.: *Approximation of a boundary–value problem using artificial neural networks* 79

Burczyński T., Szczepanik M., Kuś W.: *Optimization of stiffeners locations in 2-D structures using distributed evolutionary algorithm*..... 85

Cholewa A.: *Representation of sequences of events for purposes of inference in technical diagnostics*…..... 91

Chrzanowski P.: *Example of a diagnostic model based on belief network*……..……..……..... 95

Ciupke K., Kuciński P.: *Virtual human body model for medical applications*…….……..…. 99

Czop P., Miękina L.: *Diagnostic of electrical motors based on acoustic measurement and with use of parametric modeling*……..…… 103

Frid W., Knochenhauer M.: *Development of a Bayesian belief metwork for a boiling water reactor during fault conditions*……..... 107

Galek M.: *Expert system to aiding identification of inverse models* 111

GhasemZadeh M., Klotz V., Meinel Ch.: *Representation and evaluation of QBFs in Prenex-NNF*……..... 115

Goldasz I.: *Inverse modeling of piston valve components. Evolutionary approach* 121

Górniak-Zimroz J., Malewski J.: *Application of the Kohonen neural network for classification of mining voids*……...…… 125

Grela W., Burczyński T.: *Evolutionary shape optimisation of a turbine blade shank with APDL language*….. 129

Jankowska A., Kornacki S.: *Practical aspects of neural models applications in industry*..... 139

Jarosz P., Burczyński T.: *Immune algorithm for multi-modal optimization - numerical tests in intelligent searching*……...…… 143

Kalita P.: *Artery wall modelling – a challenge for computer science and mathematics* …… 147

Kosiński W., Cudny W., Burzyński M.: *Cellular automata and their selected applications in technology and nature modelling* ……...…… 151

Krok A., Waszczyszyn Z.: *Kalman filtering for nueral prediction of response spectra from mining tremors*... 157

Kuś W., Burczyński T.: *Parallel artificial immune system in optimization of mechanical structures*…… 163

Lin P., Zheng Ch-X., Yang Y., Gu J-W.: *Statistical model based level set method for image segmentation*……...……... 167

Ławrynowicz A.: *A genetic algorithm for job shop scheduling*…….……..... 173

Masłowska I.: *Web search results clustering - new requirements for clustering techniques*…….…..... 177

Mazur D.: *Clustering based on genetics algorithm*……...……....…… 183

Niederliński A.: *A modification of the Stanford Certainty Factor Algebra for uncertain expert systems*……... 189

Ogonowski Z., Plaza K.: *Mechanical vibrations damping improvement using higher level AI algorithms for a magnetic levitation system*…….……...……....…… 195

Oleksiak J., Ligęza A.: *Hierarchical diagnosis of technical systems on the basis of model and expert knowledge*…….……...……....…… 199

Pokojski J.: *Intelligent personal assistant and multi-criteria optimization*…….…….…..... 203

Przybyło A., Achiche S., Balazinski M., Baron L.: *Enhancing fuzzy learning with data mining techniques*……...…… 207

Psiuk K.: *Identification of bayesian network as a relations model of state changes propagation*……...…… 211

Raad A., Sidahmed M., Antoni J.: *Indicators of cyclostationarity: theory and application to gear fault diagnostic*……...…… 215

Rogala T.: *General concept of virtual sources identification of diagnostic signals*……...…….…… 219

Rutkowski J.: *Dictionary approach to fault diagnosis in analog circuits*……..……...…… 223

Skarka W.: *Capturing knowledge through web services based on scenarios during product development*……..…. 229

Skarka W.: *Object-oriented approach to modeling ontology of knowledge base*……..……. 233

Skarka W., Urbanek G.: *Web service for technical manuals*……..…. 239

Skołud B., Zientek A.: *Constraints identification in multi-project scheduling*…….....……. 243

Skupnik D., Ciupke K.: *An application of ant algorithm for diagnosis of technical object*.....……. 247

Sławik D.: *Sensitivity evaluation and sensitive feature selection* ……..…. 251

Słoński M.: *Prediction of concrete fatigue durability using Bayesian neural networks*…….……..…. 255

Sokołowski A., Czyszpak T.: *Mamdani versus Tagaki-Sugeno fuzzy reasoning for machine diagnostics*…..... 259

Stefanowski J., Kaczmarek M.: *Integrating attribute selection and dynamic voting of sub-classifiers to improve accuracy of bagging classifiers*……...…… 263

Studziński M.: *Supporting data mining technology by using case based reasoning*……..…. 269

Szulim R., Moczulski M.: *A method of mining knowledge to aid control of complex industrial processes*…….……..…. 273

Tadeusiewicz R., Ogiela M. R.: *Automatic understanding of the images*……..…. 277

Timofiejczuk A.: *Context-based approach in technical diagnostics*……..…. 281

Urbanek G.: *Rough simulator in the inverse models identification*……..……..…. 285

Wachla D.: *The general concept of a method for discovering the quantitative dynamics*……...…… 289

Witczak M., Prętki P.: *An experimental design strategy for neural networks and its application for fault detection of non-linear systems*…….....…….....…… 293

Wojtusik J.: *Distance measures and trajectories clustering*…….……..…. 297

Xiong X., Moraga C.: *Parametric feedforward neural network with fuzzy inputs configured by genetic algorithm*……...…….... 301

Yang Y., Zheng Ch., Lin P.: *Image tresholding based on spatially weighted fuzzy C-Means clustering algorithm* ……..……...…. 307

Zieniuk E., Kużelewski A.: *Modelling of potential boundary problems described by Bézier curves using the fuzzy Parametric Integral Equations System*……..…….... 313

Probabilistic PCA and neural networks in search of representative features for some yeast genome data

Anna Bartkowiak
University of Wrocław, Inst. of Computer Science, ul. Przesmyckiego 20, 51-151 Wrocław, Poland
e-mail: aba@ii.uni.wroc.pl

Stanisław Cebrat and Paweł Mackiewicz
University of Wrocław, Inst. of Genomics and Microbiology, Przybyszewskiego 63/77, 51-148 Wrocław, PL
e-mail: {cebrat,pamac}@microb.uni.wroc.pl

Abstract

We considered a data matrix with $N = 3300$ rows (objects, genes) and $d = 13$ columns representing variables (traits) measured for each object (identified gene). The 13 characteristics were obtained from so called 'spider-plots' constructed for each gene. Our goal was to find a latent structure in the data and possibly reduce the dimensionality of the data. To achieve this goal we used the methods of ordinary principal components (PC), probabilistic principal components (PPCA) and feed forward neural networks (multi-layer perceptrons). We got some evidence, that H=6 latent variables explain the essential features of the data. Our results are the following: a) First six principal components explain 0.8844 of total variance, however have no interesting interpretation; b) First 6 probabilistic principal components with rotation varimax explain 78.53 % of total variance of the data and have a very interesting interpretation: the set of the primary 12 variables is reduced to 3 double factors, each factor expressed by 2 latent variables; thus we found a meaningful latent structure with a parsimonious representation. c) The multi-layer perceptron with architecture 13–6–13 explains about 88.40 % of total variance, moreover, the matrix of weights, after permuting the columns, yields the same interesting interpretation as the PPCA. Thus a neural network (perceptron) is able to reduce the dimensionality and yield a parsimonious representation of the original variables, similar to that yielded by the PPCA. This – to our knowledge – was not noticed before.

Keywords: reduction of dimensionality, yeast genome, latent structure, probabilistic PCA, multi-layer perceptron

1. Introduction, the data and the problem

We consider data characterizing $N = 3300$ yeast genes, each described by $d = 13$ variables (traits). The data will be in the following referred to as the yeast genome data. A more detailed description of the data may be found in [4, 5, 1] or [12].

The gathered variables have a quite clear meaning and some of them are fairly dependent. Attempt to simply omit some of the variables is not working: the eventually omitted variables (by use of the *idep* procedure) can not be explained in a satisfactory manner by the retained variables. At least some of the recorded variables are fairly interdependent, which may be seen when looking at the correlation map shown in Figure 1.

Our problem is: Could the observed variables be transformed to a reduced set, containing $H < d$ new, derived features – without loosing not too much of total inertia (variance) of the entire set.

Our goal may be formulated in two points:

1. to find a latent structure in the data,
2. possibly reduce the dimensionality, i.e. the number of variables.

We will work with 3 methods:

PCA, traditional principal component analysis,

PPCA, probabilistic principal component analysis,

ANN, artificial neural networks, multi-layer perceptrons.

Special emphasis will be put on comparison of results provided by the PPCA and ANN methods.

In the following we explain briefly the methods and show some results obtained when using the chosen methods.

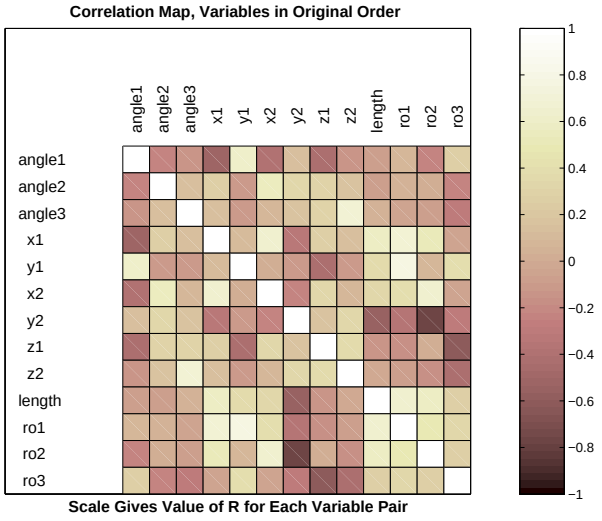

Fig. 1. Correlation map for 13 variables characterizing yeast genes. The strength of correlation – positive or negative – is expressed by color shade.

2. Traditional PCA and Probabilistic PCA

2.1. Traditional PCA

PCA is a well known technique of data analysis (see, e.g., the book by Jolliffe [7]). It works in terms of approximation theory. No underlying generative model of the data is provided. The results are heuristic, depending on the gathered data. The method reproduces in a purely mathematical way the entire data set (or, its covariance matrix), by rank one

matrices.

To perform the PCA, one needs to calculate firstly the eigenvalues $\lambda_1, \ldots, \lambda_d$ and the corresponding eigenvectors $\mathbf{a}_1, \ldots, \mathbf{a}_d$ of the covariance or correlation matrix $\mathbf{S}$ of the data. The derived eigenvalues and eigenvectors serve for construction of the principal components. They allow to reproduce the matrix $\mathbf{S}$ by lower rank matrices and estimate the quality of the reproduction.

Let $\mathbf{t} = (t_1, \ldots, t_d)^T$ denote the observed variables.

For computational convenience we assume, throughout this paper – *that the variables are 0–centered and have unit variances.* In such a case the covariance matrix is equal to the correlation matrix.

Then, with the d eigenvectors of the covariance (correlation) matrix of these variables, we may construct d new variables (features), defining them as:

$$z_j = \mathbf{a}_j^T \mathbf{t}, \quad j = 1, \ldots, d. \tag{1}$$

Definition 1. Principal components.

The variables z_j, $j = 1, \ldots, d$ constructed using the formula (1) above are called *principal components.* They are independent linear combinations of the observed variables $\mathbf{t}$ and have variances $var(z_j) = \lambda_j$, $j = 1, \ldots, d$.

For a given z_j (alias: the jth PC) the coefficients of the corresponding eigenvector $\mathbf{a}_j$ are called sometimes *l*oadings of that variable.

In the following we will need the following two properties:

Property 1. Reproduction of the covariance matrix.

$$\mathbf{S} = \sum_{j=1}^{d} \lambda_j \mathbf{a}_j \mathbf{a}_j^T. \tag{2}$$

Property 1 says that the matrix $\mathbf{S}$ can be reproduced by rank 1 matrices composed from subsequent eigenvalues and eigenvectors.

Property 2. Reproduction of the sum of variances.

$$trace(\mathbf{S}) = \sum_{j=1}^{d} s_j^2 = \sum_{j=1}^{d} trace(\lambda_j \mathbf{a}_j \mathbf{a}_j^T) = \sum_{j=1}^{d} \lambda_j. \tag{3}$$

This property describes the quality of the reproduction; we use that property to say, how much of total variance of the original variables is reproduced by subsequent principal components.

Property 2 says that the sum of variances of the original variables is equal to the sum of all eigenvalues of the covariance matrix $\mathbf{S}$. The sum of all variances ($trace(\mathbf{S})$) is called the total variance or the total inertia.

If the first eigenvalues are big (constitute a big percentage) in relation to the remaining ones, then the corresponding principal components explain a big percentage of total variance, and the remaining ones may be neglected.

2.2. *Probabilistic principal components based on the concept of latent variables*

A more general approach to modelling of the data is to introduce a generative model. This is done by building a model based on latent variables and superposition of an additional noise.

One such model, called *probabilistic principal components*, was elaborated by Tipping and Bishop (see, e.g., [13, 3]). The following basic model is assumed:

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{4}$$

or, alternatively, writing explicitly the dimensions:

$$\mathbf{t}_{d\times 1} = \mathbf{W}_{d\times H}\mathbf{x}_{H\times 1} + \boldsymbol{\mu}_{d\times 1} + \boldsymbol{\epsilon}_{d\times 1}.$$

Vector $\mathbf{t}$ denotes the observational and vector $\mathbf{x}$ – the latent variables. Conventionally it is assumed that $\mathbf{x}$ is distributed $N_H(\mathbf{0}, \mathbf{I})$, with the H latent variables offering a more parsimonious explanation of the dependencies between the observations.

The variable $\boldsymbol{\epsilon}$ in the model above denotes additional Gaussian noise, independent of $\mathbf{x}$, distributed $N_d(\mathbf{0}, \sigma^2\mathbf{I})$.

The observed values of $\mathbf{t}$ are supposed to be generated by $H < d$ hidden (latent) variables $\mathbf{x}$ distributed normally with isotropic variance.

Under the assumed model (4) the marginal distribution of the observed vector $\mathbf{t}$ is again normal (we follow here [13]):

$$\mathbf{t} \sim N_d(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}). \tag{5}$$

Hence, for a given sample of N observed vectors $\{\mathbf{t}_n\}$, $n = 1, \ldots, N$, the corresponding log-likelihood function $\mathcal{L}^*$ can be easily constructed. It takes the form:

$$\mathcal{L}^* = -\frac{N}{2}[d \cdot ln(2\pi) + ln|\mathbf{C}| + tr(\mathbf{C}^{-1}\mathbf{S})],$$

with $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$, and $\mathbf{S}$ denoting the sample covariance matrix:

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{t}_n - \boldsymbol{\mu})(\mathbf{t}_n - \boldsymbol{\mu})^T.$$

The unknown parameters of the model (5) are: $\mathbf{W}_{d\times H}$ and σ^2. They may be estimated directly from the log-likelihood $\mathcal{L}^*$. Tipping and Bishop [13] found that the log-likelihood is maximized when taking the following estimates:

$$\mathbf{W}_{ML} = \mathbf{U}_H(\boldsymbol{\Lambda}_H - \sigma^2\mathbf{I})^{1/2}\mathbf{R}, \tag{6}$$

$$\sigma^2_{ML} = \frac{1}{d-H}\sum_{j=H+1}^{d}\lambda_j, \tag{7}$$

with

- $\boldsymbol{\Lambda}_H = diag(\lambda_1, \ldots, \lambda_H)$ containing the largest H eigenvalues of $\mathbf{S}$, where $\lambda_1 \geq \ldots \geq \lambda_H$,
- $\mathbf{U}_H = [\mathbf{u}_1, \ldots, \mathbf{u}_H]$ containing the corresponding eigenvectors of $\mathbf{S}$,
- $\mathbf{R}$ being an arbitrary $H \times H$ orthogonal rotation matrix.

Tipping and Bishop [13] refer to σ^2_{ML} as to the variance 'lost' in the projection, averaged over the lost dimensions.

The presented PPCA model resembles the model of factor analysis (known in the statistical methodology), however, as pointed out in [13], there are some important distinctions resulting from the use of the isotropic noise covariance in generating the $\mathbf{x}$'es. One major distinction is that PPCA extracts the principal axes incrementally.

Comparing the PCA and PPCA we state, that both methods use the eigenvalues and eigenvectors of $\mathbf{S}$ as key elements for computing. However the derived principal directions are scaled in a different way.

The found statistics $\mathbf{W}_{ML}$ and σ^2_{ML} provide a maximum likelihood estimate for $\mathbf{C}$, the covariance matrix of the observed variables $\mathbf{t}$ generated – according to the model (4) – by H latent variables:

$$\hat{\mathbf{C}} = \mathbf{W}_{ML}\mathbf{W}^T_{ML} + \sigma^2_{ML}\mathbf{I}$$

Obviously, all the derived estimates depend from the number H of latent variables.

Thus, the formula above might be rewritten in a more precise way for $H = 1, \ldots, d-1$ as

$$\hat{\mathbf{C}}_{(H)} = \mathbf{W}_{ML(H)}\mathbf{W}^T_{ML(H)} + \sigma^2_{ML(H)}\mathbf{I}. \tag{8}$$

To get an analogy with Property 1 of principal components, let $\tilde{\mathbf{w}}_{j(H)}$ denote the jth column of $\mathbf{W}_{ML(H)}$. Then $\mathbf{W}_{ML(H)}$ may be rewritten as

$$\mathbf{W}_{ML(H)} = [\tilde{\mathbf{w}}_{1(H)}, \ldots, \tilde{\mathbf{w}}_{H(H)}].$$

Taking this into account, we may rewrite the formula for $\hat{\mathbf{C}}_{(H)}$ as

$$\hat{\mathbf{C}}_{(H)} = \sum_{j=1}^{H} \tilde{\mathbf{w}}_{j(H)}\tilde{\mathbf{w}}^T_{j(H)} + \sigma^2_{ML(H)}\mathbf{I}, \quad 1 \le H < d\,. \tag{9}$$

2.3. How to find the right dimension?

The fundamental question is: How large should be H, denoting the number of retained PCs (when performing PCA), and the number of hidden (latent) variables (when performing PPCA)?

Using PCA, the question may be answered by inspecting so called *scree graph*, constructed from the eigenvalues derived from the analyzed data (see, e.g., Jolliffe [7]). The scree graph constructed for the gene data is shown in Figure 2.

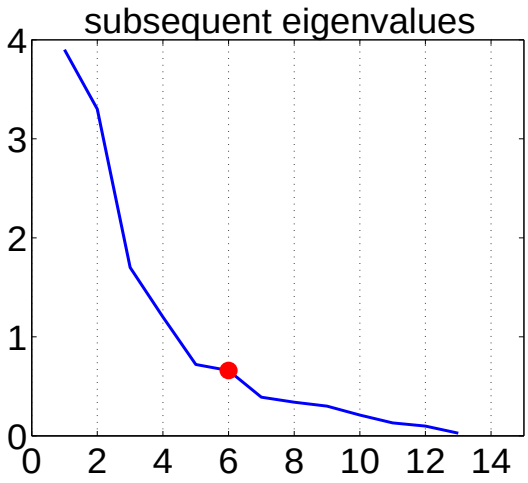

Fig. 2. PPC. Scree graph from correlation matrix calculated from $N = 3300$ genes. The graph exhibits the decay of subsequent eigenvalues 1, 2, ... 13. One may notice that beginning from the seventh eigenvalue the decay exhibits linear pattern.

Albeit inference based on a scree graph belongs to heuristic methods, the scree graph proved to be very useful in practice.

The scree graph is simply a plot of (j, λ_j), $\quad j = 1, \ldots, d$. We look at the decay of subsequent eigenvalues. When the decay starts to be linear, then it is deduced that the entire interdependence structure between the observed variables is already explained and included in the previous PC's.

Inspecting the graph shown in Figure 2 one may see that starting from the 7th eigenvalue the decay is linear. Thus the intrinsic dimension of the data is accepted as $H = 6$.

Similarly, when working with PPCA, *the number of latent variables* may be estimated from the scree graph exhibiting the decay of residual variances obtained from eq. (7). The respective graph is shown Figure 3.

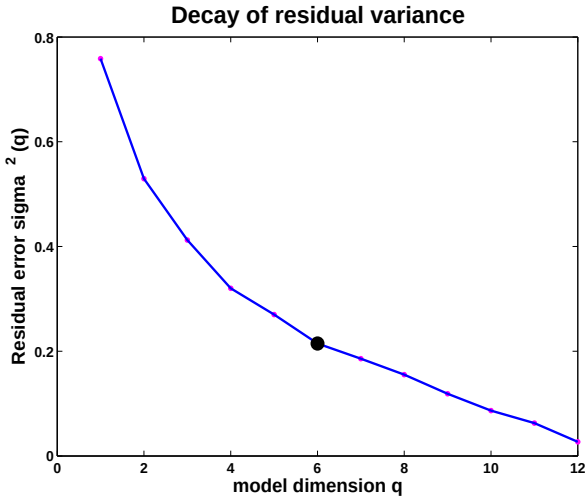

Fig. 3. PPCA. Scree graph obtained on the basis of correlation matrix calculated from $N = 3300$ genes. Residual variances σ^2_{ML} are put against q, the number of latent variables included into the model. For model dimension q, the residual variance is calculated as $(\sum_{j>q} \lambda_j)/(d-q)$.

Generally, both statistics (i.e. the eigenvalues and the residual errors) decrease, when augmenting the number of latent variables included into the model. Starting from a number q, $1 \le q < d$ the decay exhibits a linear pattern. Appearance of a linear pattern means that the decay is proportional to the number $d-q$ of remaining (not yet accounted for) latent variables. This means also that there is no more common structure to extract; and that the remaining variables are independent and have an individual random variance σ^2, which may viewed as Gaussian noise.

Looking at the plots shown in Figure 3 one may state, that this linear decay pattern starts to the right of $q = 6$. To the right of the 6th residual variance – marked by a big filled circle – the decay exhibits a linear pattern, which means that no more common factors can be extracted. Thus it seems that $q = 6$ is the right number of latent variables.

Therefore we decided to seek for $H = 6$ hidden factors.

3. Multi-layer perceptron

Artificial neural networks (ANNs) provide a methodology which may be seen in terms of predictions. The ANN, learning from a provided training sample, is expected to build a

model, which – on the basis of given explanatory variables – permits to predict the sought target.

Nabney [9] writes: "The goal of training a network is to model the underlying generator of the data in order to make the best possible predictions when new input data is presented. The most general information about the target vector $\mathbf{t}$ for inputs $\mathbf{x}$ is given by the conditional density $p(\mathbf{t}|\,\mathbf{x})$".

Generally, artificial neural networks are considered as semi-parametric or non-parametric models for data analysis, see e.g., Gaudart et al. [6], and the references therein.

Neural networks have developed a special type of learning (Hebbian learning) to capture the essential characteristics (main directions) of the data. Quite a lot of research was needed to find out, what really the Hebbian learning is yielding.

Realization of the method of principal components in the framework of Hebbian learning was the subject of many investigations, (see, e.g., the papers by Oja, Sanger et others). Recently, a critical discussion of the approaches has been published by Nicole [11]. Our opinion is that the algebraical method, as presented e.g., in Jolliffe [7], is many times faster and yields univocal results (see also G. Bazan [2]).

Instead of the traditional Hebbian approach we formulated the task in terms of approximation of the data. A simple feed-forward neural network, the multi-layer perceptron, was used. The network had as target just the data presented at the input. The number of neurons in the hidden layer was put equal to H, the number of the desired latent variables (in our case, this was $H = 6$).

The applied perceptron had 3 layers: the input layer, the hidden layer and the output layer. The layout of the network was: 13 – 6 – 13. This means, there were

- $d = 13$ neurons at the input,
- $H = 6$ neurons in the hidden layer,
- $K = d = 13$ neurons at the output.

The hidden layer produced 6 derived variables $z_1, \ldots, z_6$. They acted as input to the third (output) layer who's task was to reproduce from the z's the target, which was again the input vector.

Let us mention that the implementation in Netlab [9], which was used for our calculations, puts obligatorily in the second layer of the perceptron the 'tanh' activation function, which makes that all z's are contained in the interval (-1,1).

We have declared for the output layer the 'linear' activation function.

Let $\dot{\mathbf{W}}_{H\times d}$ and $\ddot{\mathbf{W}}_{d\times H}$ denote the weights of the hidden and the output layer appropriately.

The neurons of the hidden layer perform a transformation that maps the input data into a feature space $I\!R^H$. For a given input vector $\mathbf{t} = [t_1, \ldots, t_d]^T \in I\!R^d$ we obtain a vector of derived variables/features $\mathbf{z} = [z_1, \ldots, z_H]^T \in I\!R^H$ calculated as

$$z_h = tanh(\sum_{j=1}^{d} \dot{w}_{hj} t_j + \dot{b}_h), \quad h = 1, \ldots, H. \tag{10}$$

The derived variables $\mathbf{z} = [z_1, \ldots, z_H]^T$ serve as base to construct d another variables $y_1, \ldots, y_H$ approximating the original variables $t_1, \ldots, t_H$. This is done using the formula

$$y_k = \sum_{h=1}^{H} \ddot{w}_{kh} z_h + \ddot{b}_k, \quad k = 1, \ldots, d.$$

The formula above, by analogy with model [4], may be written also in a vector-matrix notation ($\mathbf{y} = [y_1, \ldots, y_d]^T$)

$$\mathbf{y}_{d\times 1} = \ddot{\mathbf{W}}_{d\times H} \mathbf{z}_{H\times 1} + \ddot{\mathbf{b}}_{d\times 1}. \tag{11}$$

4. Results

The analysis was carried out using standardized data. This means that the covariance matrix was equal to the correlation matrix of the data. In such a case the trace of $\mathbf{S}$ equals to the number of analyzed variables:

$$trace(\mathbf{S}) = \sum_{j=1}^{d} s_j^2 = \sum_{j=1}^{d} 1 = d.$$

Our data contained $d = 13$ variables. Their names may be read in Figure 1 and in Tables 2–5. The variables characterize three legs notified in the spider-plot describing an ORF (Open Reading Frame in a chromosome) [5, 1, 12] . Each leg is characterized by 4 variables. An additional variable, length of the ORF, is introduced as variable no. 10. The data contained a total $N = 3300$ ORFs.

After an analysis of the scree graphs shown in Figure 2 and 3 we decided to seek for $H = 6$ hidden factors, alias latent variables, alias PPCs.

4.1. Traditional PCA

To do the PCA analysis, we calculated firstly the eigenvalues $\lambda_1, \ldots, ..., \lambda_d$ and the corresponding eigenvectors $\mathbf{a}_1, \ldots, \mathbf{a}_d$ of the correlation matrix $\mathbf{S}$ of our data.

By use of *Property 2* we were then able to say, how much of total variance of the original variables is reproduced by subsequent PC's. This is shown in Table 1.

Table 1. Reproduction of total variance by 2, 3, 5, 6, 7 and 10 principal components.

PC no. j	2	3	5	6	7	10
eigenvalue	3.3	1.7	0.72	0.66	0.39	0.21
fraction sum	0.55	0.68	0.83	**0.88**	0.91	0.98

One may see that 6 principal components reproduce 88% of total variance of the data.

One might ask also, for each variable separately, how much of individual variance (for each variable) is explained by subsequent PC's. We may calculate this using *Property 1* and considering only the diagonals of the respective matrices. The details of individual reproduction, when using models with j = 5, 6, and 7 PC's, are shown in Table 3.

On the basis of the derived eigenvectors $\{\mathbf{a}_j\}$ we calculated the PC's by use of formula (1).

The meaning (interpretation) of the obtained PC's may be obtained by looking at their correlations with the standardized original variables $t_1^*, \ldots, t_d^*$. In the case, when the eigenvectors were obtained from the correlation matrix, the respective correlations may be found by a simple re-scaling of the obtained eigenvectors ([7], p. 25).

In Table 2 we show the correlations of the first 6 PC's with the standardized observed variables t^* (i.e. 0–centered and with unit variances). Each column $\mathbf{r}_j$ was obtained simply as

$$\mathbf{r}_j = \sqrt{\lambda_j}\,\mathbf{a}_j, \quad j = 1, \ldots, 6.$$

Table 2. Correlation coefficients $\{r_{ij}\}$of the first 6 PC's with the observed standardized variables $t_1^*, \ldots, t_{13}^*$.

Variable	PC No.					
$t_i \downarrow$	1	2	3	4	5	6
1. ang1	-0.10	0.71	-0.46	0.08	0.09	-0.44
2. ang2	-0.00	-0.55	-0.17	0.69	-0.26	-0.08
3. ang3	-0.16	-0.49	-0.50	-0.46	-0.37	-0.08
4. x1	0.69	-0.56	-0.09	0.02	0.16	0.33
5. y1	0.48	0.49	-0.65	0.15	0.18	-0.07
6. x2	0.56	-0.61	0.03	0.36	-0.14	-0.24
7. y2	-0.72	-0.11	-0.42	0.39	0.00	0.24
8. x3	-0.20	-0.77	0.05	-0.00	0.38	-0.19
9. y3	-0.26	-0.54	-0.55	-0.38	-0.14	-0.02
10. leng	0.80	-0.04	-0.14	-0.28	0.05	0.09
11. rho1	0.84	0.05	-0.43	0.11	0.17	0.16
12. rho2	0.82	-0.19	0.30	-0.10	-0.12	-0.34
13. rho3	0.45	0.61	0.07	0.09	-0.47	0.19

On the basis of the correlation coefficients shown in Table 2, we were not able to attach any interesting meaning to the derived PC's.

4.2. Probabilistic principal components

After analysis of residual variances shown in Fig. 3, we have fixed $H = 6$ latent variables. The corresponding probabilistic model was estimated using formulae (5), (6) and (7).

One might ask again, how much of total variance is reproduced by the constructed latent variables. The answer may be read from Table 3. exhibiting the reductions obtained when using $H = 5$, 6 and 7 principal components (PC's) or latent variables (PPC's).

The values in the columns of Table 3 were obtained as

$$\sum_{j=1}^{H} diag(\lambda_j\,\mathbf{a}_j\,\mathbf{a}_j^T), \qquad \text{for PCA,}$$

$$\sum_{j=1}^{H} diag(\tilde{\mathbf{w}}_{j(H)}\,\tilde{\mathbf{w}}_{j(H)}^T), \quad \text{for PPCA.}$$

One may see in Table 3 that the probabilistic principal components extract about 10% less of common structure as this is done by ordinary principal components. This is no surprise.

The estimated matrix $\mathbf{W}_{ML(6)}$ – obtained from formula [6] was subjected to rotation *varimax*. The rotated matrix is shown in Table 4.

Looking at the matrix $\mathbf{W}$ in a coarse way (i.e. looking only at the loadings $|w_{ij}| > 0.30$) one may state that the matrix exhibits a particular structure: The derived variables may be split into three pairs headed in Table 4 as: '1.leg', '3.leg', and '2.leg'. Each pair is mainly composed by 4 observed variables describing one leg of the spider-plot (see [5, 12, 1] for description of the observed data and the construction of the spider-plots).

Table 3. Reduction of variance of individual variables. Part of total variance reproduced by 5, 6, 7 ordinary principal components and 5, 6 and 7 PPC's. One may notice that PPCA reproduces a smaller part of total inertia then the ordinary PCA.

Original	PC No.			PPC No.		
variable	5	6	7	5	6	7
1 ang1	0.74	0.94	0.94	0.66	0.81	0.83
2 ang2	0.88	0.89	0.93	0.72	0.76	0.80
3 ang3	0.86	0.87	0.88	0.71	0.74	0.77
4 x1	0.83	0.93	0.94	0.76	0.84	0.86
5 y1	0.94	0.95	0.95	0.82	0.85	0.87
6 x2	0.83	0.89	0.89	0.75	0.80	0.82
7 y2	0.85	0.90	0.91	0.75	0.81	0.82
8 x3	0.77	0.81	0.91	0.67	0.71	0.78
9 y3	0.82	0.82	0.83	0.71	0.73	0.74
10 leng	0.74	0.75	0.85	0.68	0.70	0.76
11 rho1	0.94	0.96	0.97	0.84	0.88	0.90
12 rho2	0.82	0.93	0.94	0.75	0.84	0.85
13 rho3	0.81	0.84	0.94	0.68	0.73	0.79
sum/13	0.83	**0.88**	0.91	0.73	**0.78**	0.81

Thus, e.g., the first two columns are spanned mainly by the (original) variables x1, y1, ang1 and rho1, which are just the variables describing the first leg of the spider-plot. It appears that all these four variables are necessary and bear important information; they have the main shares in the derived PPC's. However, also 'length' (variable no. 10) has a big share in the first PPC.

The split into the 3 double factors is not ideal: we have looked only at the loadings $|w_{ij}| > 0.30$). Concerning the first PPC ('1.leg'), also x2, y2 and rho2 have some minor shares in it. This might mean that in fact the derived PPC's, albeit formally orthogonal, share among them some additional intrinsic information.

4.3. Neural networks

As described in Section 3, we have applied a multilayer perceptron in the 13–6–13 layout. It had a 'tanh' activation function in the hidden layer and a 'linear' activation function in the output layer. We have carried out the calculations using Netlab [10].

The neural network needed about 3000 epochs (presentations of the data matrix) to get stabilized parameters. The derived matrix $\dot{\mathbf{W}}$ is shown in Table 5. All weights $\{\dot{w}_{ij}\}$ were multiplied by 10.

It was a big surprise to us to obtain, by such a standard and simple tool as the perceptron, results very similar to those, obtained by a sophisticated method – as the PPCA with rotation varimax is.

To obtain a comparable index exhibiting how much of original variance can be reduced when using the neural network model with 6 neurons in the hidden layer, we have defined such index as the squared ratio of the Frobenius norm of $\mathbf{T}$, the observed data matrix, in the numerator, and the Frobe-

Table 4. PPCA. Matrix $\mathbf{W}$ expressing 6 latent variables for the yeast genome data. The presented matrix was obtained from rotated matrix $\mathbf{U_H}\sqrt{(\mathbf{\Lambda}_H - \sigma^2_{ML}\mathbf{I})}$, $H = 6$.

	1.leg	1.leg	3.leg	2.leg	3.leg	2.leg	%
ang1	-.08	**.84**	.07	-.18	-.21	.14	.81
ang2	.03	-.10	-.08	**.83**	.13	.17	.76
ang3	.00	-.06	**-.85**	.06	.11	.02	.74
x1	**.72**	**-.37**	-.09	.28	.15	-.28	.84
y1	**.58**	**.67**	.06	-.02	-.27	.02	.85
x2	.30	-.17	-.04	**.69**	.16	**-.42**	.80
y2	-.21	.08	-.17	.20	.14	**.82**	.81
x3	-.04	-.27	-.20	.24	**.74**	-.01	.71
y3	.05	-.04	**-.79**	.05	.27	.17	.73
leng	.65	-.01	-.08	-.06	-.14	-.50	.70
rho1	**.85**	.21	.04	.14	-.16	-.24	.88
rho2	.29	-.10	.09	.21	-.05	**-.83**	.84
rho3	.15	.12	.24	-.05	**-.77**	-.19	.73

Table 5. ANN, results from training a 3-layer perceptron. Weights $\dot{\mathbf{W}}$ linking the hidden layer with the input layer are shown. All weights were multiplied by 10. To be comparable with results from Table 3, some columns should be permuted.

	3.leg	2.leg	2.leg	1.leg	1.leg	3.leg
ang1	-.10	.03	-.22	**-.76**	-.26	-.43
ang2	.41	-.32	**-.64**	.21	-.34	.06
ang3	**.96**	.04	.37	**-.40**	-.15	**.61**
x1	.17	-.11	.06	**.43**	**.51**	-.20
y1	.06	.02	-.31	**-.37**	.31	-.54
x2	.27	**-.55**	-.24	.07	-.29	-.02
y2	.27	**.40**	**-.44**	.26	.05	-.20
x3	-.08	-.16	.32	.20	-.25	**-.75**
y3	**.82**	.13	.36	-.26	-.05	.19
leng	.15	-.13	.24	-.09	.39	.01
rho1	.12	-.05	-.18	-.00	**.52**	-.40
rho2	-.05	**-.55**	.20	-.23	-.25	.18
rho3	.12	-.03	**-.39**	-.11	.22	**.92**

nius norm of the predicted data matrix $\mathbf{Y}$ obtained by formula [11] – in the denominator (the original data matrix $\mathbf{T}$ was standardized to mean 0 and unit variances). Surprisingly enough, this squared ratio appeared to be equal to 0.8838, a value very similar to that obtained by classical PCA.

5. Discussion and closing remarks

We got results interesting for several reasons:

1. It was confirmed, that principal components (PC's) extract too much of variability of the data set (which means, that the PC's account some random effects as systematic effects).

2. The new features (latent variables, PPC's), derived from the observed variables, have a very clear and interesting interpretation. The original 13 variables may be represented by 6 derived variables, called latent variables or factors. These factors appear grouped in pairs. Each pair is spanned by 4 original variables, having an interesting interpretation.

3. It was stated that multi-layer perceptrons may be used for modelling of the data, reduction of dimensionality and finding latent factors. This ability was put recently in doubt by Nicole [11].

   In particular, it was interesting to state, that the weights of the perceptron provided a kind of parsimonious loadings of the observed variables considered as a function of the hidden variables. This, to our knowledge, was not noticed before.

References

[1] A. Bartkowiak, A. Szustalewicz, S. Cebrat, P. Mackiewicz, *Kohonen's self-organizing maps as applied to graphical visualization of some yeast DNA data*, Biometrical Letters, **40**, 2003, No. 2, 37–56.

[2] G. Bazan, *Calculations of principal components using the Oja and Sanger methods.* In Polish. Master Diploma Dissertation, Institute of Computer Science, University of Wrocław, 2001.

[3] Ch.M. Bishop, M.E. Tipping, *Latent variable models and data visualization.* In: J.W. Kay and D.M. Titterington (Eds), Statistics and Neural networks, Advances at the Interface. Oxford University Press, 1999, 147–164.

[4] S. Cebrat, M.R. Dudek, *The effect of DNA phase structure on DNA walks.* The European Physical Journal B., **3** (1998), 271–276.

[5] S. Cebrat, P. Mackiewicz, M.R. Dudek, *The role of the genetic code in generating new coding sequences inside existing genes.* Biosystems, **45** (2) (1988), 165–176.

[6] J. Gaudart, B. Giusiano, L. Huiart, *Comparison of the performance of multi-layer perceptron and linear regression for epidemiological data.* Computational Statistics & Data Analysis, **44** (2004), 547–570.

[7] I.T. Jolliffe, *Principal Components Analysis,* 2nd Edition. Springer, New York, Berlin, 2002.

[8] *Matlab, The Language of Technical Computing, Version 6.5.* The Mathworks Inc., Natick MA USA, 2002.

[9] I. Nabney, *Netlab: Algorithms for Pattern Recognition.* Springer, 2002.

[10] Netlab by Ian Nabney: Netlab neural network software, Neural Computing Research Group, Division of Electric Engineering and Computer Science, Aston University, Birmingham UK, http://www.ncrg.aston.ac.uk/

[11] S. Nicole, *Feedforward neural networks for principal components extraction.* Computational Statistics & Data Analysis, **33** (2000), 425–437.

[12] Smorfland: http://smorfland.microb.uni.wroc.pl/

[13] M.E. Tipping, C.M. Bishop, *Probabilistic principal component analysis.* J. Roy. Stat. Society, B, **61** (1999), 611-622.