{file nkesh.tex September 17, 2001}

An interesting grouping of Polish voievodeships obtained by self-organizing maps

Anna Bartkowiak
Inst. of Computer Science, Wrocław University,
Przesmyckiego 20, Wrocław 53–502, Pl

Running title: An interesting grouping by self-organizing maps

Abstract

We consider a data set containing 49 Polish administrative units, (voievodeships), each characterized by $p = 9$ variables. To discover similarities and dissimilarities among the units we use *self-organizing maps* (Kohonen 1995).

Applying this method we obtain a very interesting map reflecting both geography, history and development of the units: we obtain grouping due to cultural development (university cities) and due to history, where the influence of powerful neighbors occupying Polish territory for over 150 years in the past is still visible.

The method is compared with other methods of visualizing multivariate data: the grand tour and spin plots from principal coordinates.

KEYWORDS: visualization of multivariate data, self-organizing map, clustering of objects

1 Introduction

We consider a data set containing 49 Polish administrative units, each characterized by $p = 9$ variables describing socio–economic status of the unit. These variables are: X_1- usage of artificial fertilizers (in kg/ha), X_2- percentage of population under age of 18 years, X_3- divorce rate per 1000 of pop., X_4- natural increase per 1000 of pop., X_5- employment in industry (% per 1000 men), X_6- number of medical doctors per 10000 pop., X_7- % of married men, X_8- % of married women, X_9- % of persons with at least secondary education level. Speaking mathematically, each unit is characterized by a data vector $\mathbf{x} = (x_1, \ldots, x_p)$ and may be imagined as a data point located in R^p, a nine-dimensional variables space.

The analyzed data contain averaged values of the 9 variables recorded in the years 1989–1992. An augmented set of the data was considered by Bartkowiak and Kupść (1995) in the context of a regression analysis.

When establishing similarities among administrative units with given characteristics, it is very profitable to make a graphical visualization exhibiting somehow the discovered proximities or individuality of some units. The displays may be very different. Most of the displays encountered in publications take into account at one time only one or two variables; in some special cases one or two other variables are taken into account. This is decidedly not sufficient when establishing similarities among units which are truly multidimensional.

A very popular method for grouping multivariate units is cluster analysis. The results may then be displayed in the form of a dendrogram which exhibits the ordering of the considered units with respect to their similarity defined in one or the other way. However, by their nature, the similarities between multivariate units follow often a complicated pattern which can not be visualized by such simple tool as a dendrogram exploiting in fact only the concept of linear ordering.

In last years, with increase of computer speed and memory availability, also with better graphical facilities, there is an overall trend of developing new methods of data analysis which could use the nowadays available computer facilities. The problems are not easy and, in fact, an entirely new approach to data analysis should be promoted – see, e.g., for some ideas presented in the papers by Unwin (2000), Fayad (1999) and Wegman (1998). Generally, the emphasis is put on data visualization – the user should use his/her eyes and be able to *see* and *perceive* how near or how distant the analyzed units are when considered as multivariate points located in the variables space R^p.

In the following it is shown how to built and exploit an user–friendly graphical interface for data visualization. The data are considered as a cloud of p-variate data points, each representing an *individual* or an *unit* characterized by p traits called in the following also variables. The goal of the analysis is to discover some internal groupings within the data set. The basic methods used for the visualization are: Kohonen's self-organizing maps and grand–tour projections. These two methods – when combined together in an easy, interactive mode – may complement each other and throw light on interesting facts not noticed other ways. The methods and their results are shortly described in Section 2 and Section 3. For the analyzed data some very interesting grouping was discovered.

Sometimes it is possible to obtain a reduction of dimensionality of the data, resulting in 2 or 3 principal coordinates or latent variables, being sufficient for a good approximation to the original data. In such a case the data set may be visualized directly by a scatterplot or spin plot constructed from the representative principal coordinates. This is shown in Section 4.

In Section 5 some additional methods useful in multivariate data are mentioned.

2 Visualization of data points by self-organizing maps

Self-organizing maps, like Kohonen neural networks (Kohonen 1995; Vesanto 1999) aim at obtaining a planar (2D) representation of points from the multivariate space R^p ($p > 2$). The planar mapping is supposed to preserve and reflect the topology of the true mutual position of the data points in the original p-dimensional space.

Kohonen's self organizing maps constitute specific artificial neural network, where neurons are located usually in a hexagonal or rectangular grid, called *map*. The neurons located in the map are called also nodes (of the map). The neurons have assigned weight vectors, called codebook vectors, which are subjected to an unsupervised learning process (competitive learning).

On the basis of the analyzed data vectors (presented to the network in various combinations in subsequent 'epochs'[1]) the network (in many epochs) adapts the codebook vectors with the aim (i) to made them reflect the position and density of data points in R^p, (ii) with the additional constraint that the adapted codebook preserve the neighborhood relations among the individual data points.

Practically this means that – after finishing the training – data points located nearby each other in R^p, the multivariate space, are located also nearby in the planar map. Using colors or shades of gray we may also mark in the map the distances between the units.

The learning algorithm proceeds in two stages. The first stage allows for big displacements of the codebook vectors. In the second stage the displacements of the codebook vectors are restricted and finally become negligible.

Of course, to use efficiently this method, we should declare some tuning parameters. Among them are: size of the grid, shape of the neighborhood, shrinkage rate of the neighborhood radius, number of iterations (epochs) in

[1]in neural networks an epoch means one iteration of the training algorithm

the first and second stage of learning. Because the learning is iterative, we need also an initial configuration of the codebook vectors, which is usually done starting from an at random assignement.

In Figure 1 we show maps obtained when using 6×6 and 10×10 grids of nodes (neurons) corresponding to the $6 \times 6 = 36$ and $10 \times 10 = 100$ codebook vectors (the nodes of the maps are more clearly seen in Figure 2). We have assumed a hexagonal neighborhood of neurons and also a hexagonal grid of the nodes for both maps. The starting positions of the codebook vectors were generated at random, independently for each map.

The obtained maps are calibrated in shades of gray using the *umat* technique (Kohonen 1995) in such a way that they exhibit the median of distances of each codebook vector (located truly in R^p, but corresponding to the node of the map) to its neighboring codebook vectors.

According to the employed algorithm, after the considered 49 administrative units (speking more precisely: their data points $\mathbf{x}_i$ $i = 1, \ldots, 49$) are assigned to the nearest codebook vector each. Considering our data, it happened, that some codebook vectors have attracted several data points, while some others – none of them. This fact is depicted in the maps shown in Figure 1 (remind, that each node in the map is connected with one codebook vector in R^p). Integers (from the range 1–49) written on the top of a node indicate which data points were nearest to the codebook vector corresponding to that node.

Looking at the maps presented in Figure 1 one may notice that the data points are distributed over the maps highly unevenly. Many nodes – depicted by filled circles – are empty, while some of them have attracted up to 4 data points.

In case of the first map characterized by a 6×6 grid, thus which has fewer nodes as the second map, the configuration of points is more tight. One may notice that quite a lot of nodes from this map have atracted 2, 3 or 4 surrounding data points.

In the second map, with a 10×10 grid, the space tessellation was more loose and we notice that only 7 nodes of the map represent two data points, while 58 nodes (out of 100) are empty, which means that their codebook vectors are designating in the R^p space Voronoi regions without any data point from our analyzed data set.

What concerns the configuration of the data points $\mathbf{x_i}$ $i = 1, \ldots, 49$ as seen in the maps one may notice, that geometrically (i.e. accounting for their coordinates in the maps), is is different; however *topologically* it is equivalent – we are going to consider this matter in more detail.

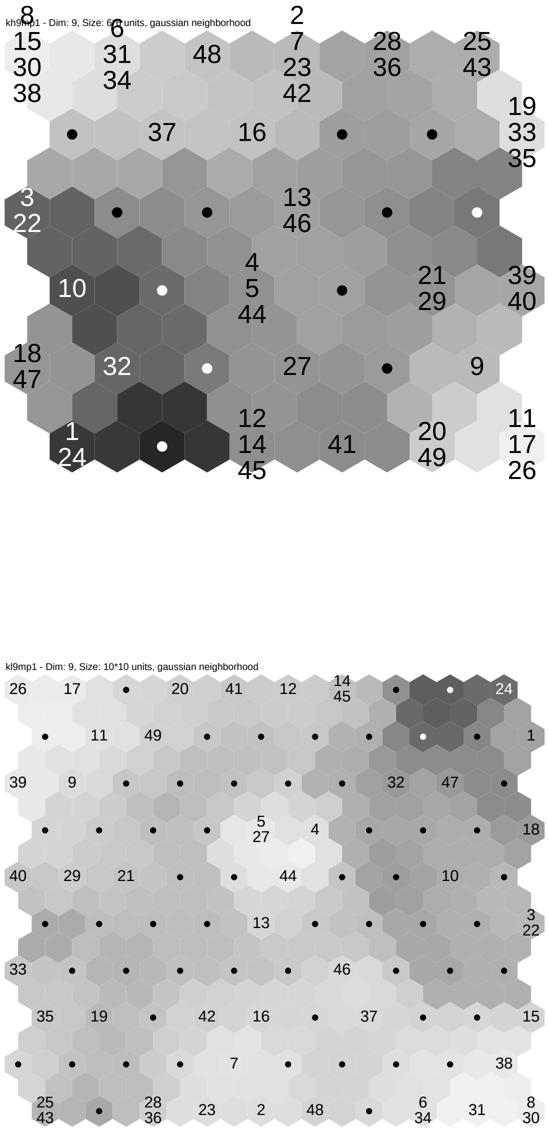

Figure 1. Planar Kohonen maps illustrating similarities between administrative units numbered $1, \ldots, 49$. *Dark color means that the corresponding codebook vectors are distant. Top: Map constructed on a hexagonal 6×6 grid. Bottom: Map constructed using a hexagonal 10×10 grid. Filled black circles denote empty nodes. A node may be representative for several units*

Our more detailed analysis of the configuration (of the data points) encountered in the maps starts from the fact that:

Data points assigned to the same nodes may be considered as located very close each other. Data points assigned to neighboring nodes located in areas with a similar shade of gray may be again considered to some extent as 'close' – thus establishing a homogeneous subset.

To establish some homogeneous subsets in the maps shown in Figure 1 we found close data points starting from the corners of the map. After a careful analysis we have established the following four subgroups of data points:

Selection 1: `U` – *Universities,* contains the units: 1, 24, 18, 47, 32, 3, 22.

Selection 2: `WN` – *West North,* contains the units: 20, 49, 41, 11, 17, 26, 9.

Selection 3: `EC` – *East Central,* contains the units: 6, 2, 23, 36, 28, 25, 43, 19, 42, 33, 35.

Selection 4: `C` – *Central,* contains the units: 8, 15, 30, 38, 31, 37.

It happens, that these subgroups contain administrative units which have a specific geographical position and had in the past a different economic development. Moreover, in one corner of the map, separated clearly by dark hexagons from other units are two points, corresponding to Warsaw and Łódź, really with a specific structure, notified by an independent statistical analysis as outliers.

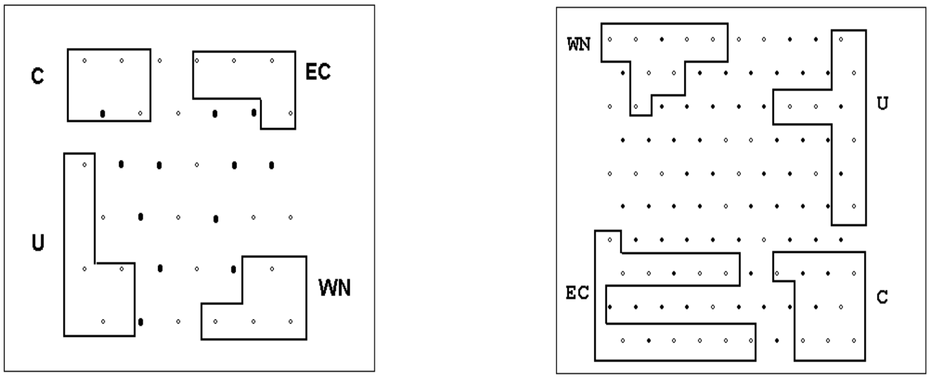

Figure 2. Schemes presenting similarity subgroups C, U, EC, WN discovered in maps exhibited in Figure 1. Filled circles denote empty nodes. Left: Scheme corresponding to the upper map in Figure 1. Right: Scheme corresponding to the bottom map in Figure 1. Comparing both figures one may notice, that the location of the subgroups in the correspondent upper and bottom plot is different however the content is the same

In the following we describe the established subgroups, called *Selections.*

Selection 1, subgroup `U`, located in one corner of the maps, contains administrative units (voivodeships) with a strong *Academia* life. These are: no. 1 – Warsaw; no. 24 – Łódź; no. 18 – Kraków; no. 47 – Wrocław; no. 32 – Poznań; no. 3 – Białystok; no. 22 – Lublin.

Very near to this subgroup, however separated in the lower plot of Fig. 1. by a hexagon of empty neurons, is: no. 10 – Gdynia-Gdańsk, also with strong Academia life, however specific in other characteristics.

Geographically these units are scattered all over Poland – but in the maps they appeared grouped together.

The units no. 1, Warsaw, and no. 24, Łódź, were formerly notified as outliers (Bartkowiak and Kupść, 1995).

The remaining three selections reflect geographical location of the units and their historical inheritance.

Selection 2, subgroup `WN`, located in another corner of the maps, contains administrative units (voivodeships) located in the western and northern area of Poland. The specificity of these areas lies in the fact that they were for several hundreds of years under the German authority and they came back to Poland only in 1945 after the Jałta treaty.

It is to some extent surprising that the 9 considered traits could sort out this specificity.

Selection 3, subgroup `EC`, located in another corner of the maps, contains administrative units (voivodeships) in the eastern and partially central part of Poland. Again, we have here a historical heritage. Poland as a country has lost completely its independence in 1795. Its territory was then annexed by 3 big neighbors: Prussia, Russia and Austria. Poland has recovered its independence after the Polish uprising in 1918.

Curiously enough, the former Russian influence got its manifestation in the subgroup of the administrative units isolated in *selection 3*.

Selection 4, subgroup `C`, located in another corner of the maps, contains administrative units (voivodeships) located geographically in the central part of Poland. The shades of gray are here similar as those of selection 3, one may also notice, that both areas (i.e. those of selection 3 and selection 4) are adjacent in the maps shown in Figure 1.

The essential question is: Are the discovered regularities in the maps accidental – or do they correspond really to different location in the variables space?

This question will be answered in next section.

3 Confirmation by projections from the grand tour

The grand tour is like a movie which reports different views of the multivariate data cloud consisting of the analyzed data points.

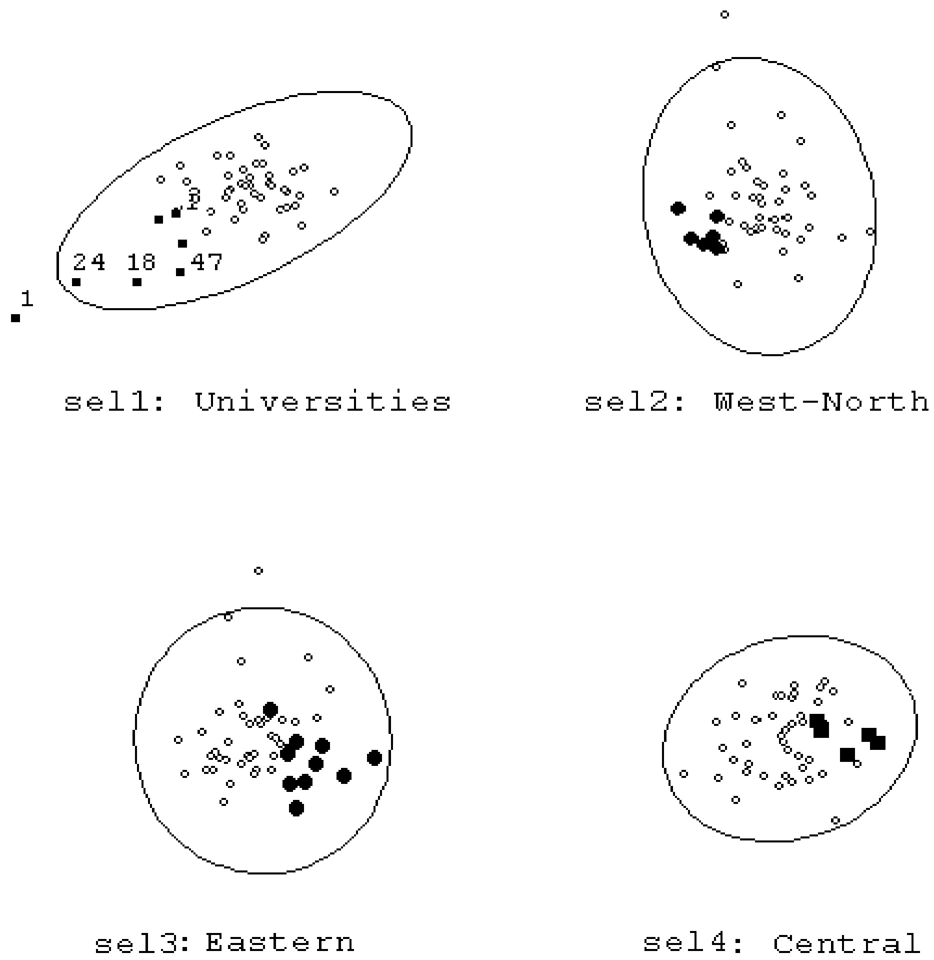

Figure 3. Snapshots from the Grand Tour –illustrating the identified subsets described as Selection 1, `U`*, Selection 2,* `WN`*, Selection 3,* `EC`*, Selection 4,* `C`

The grand tour has been gaining in last years more and more attention (see eg. Wegman, Poston and Solka, 1998; Huh and Kiyeol, 1999; Bartkowiak and Szustalewicz, 2000; Bartkowiak, 2001) for more detailed explanation and further references). Briefly speaking, this is a method which permits to "see" and perceive clusters of data points located in truly multivariate space. The method works by making quasi-continuous projections from multivariate space onto two- or three-dimensional subspaces. The method has been, a.o., successfully applied to outlier detection (Bartkowiak and Szustalewicz, 2000) and for process tracking (Huh and Kiyeol, 1999).

In Fig. 3. we show some snapshots from the grand tour projections with the aim to indicate the position of the selected subgroups, i.e. of the *selections* `U, WN, C` and `EC` performed in Section 2.

The plots exhibited in that figure are showing also a 99% ellipse of concentration superimposed on the scatterplot of points–projections. The ellipse is aimed at detecting outliers. In fact, two outliers were detected: no. 1, Warsaw, and no. 2, Łódź.

Generally, the plots in Fig. 3. confirm, that our selections were right, and that the data points belonging to the selected subgroups are really near in R^p.

4 Confirmation in spin plots constructed from principal coordinates

The method of principal components (called sometimes the *Karhunen-Levy transformation*) permits to reconstruct the data matrix from a reduced number of constructed variables, called Principal Coordinates. The same method permits also to reconstruct the covariance matrix $\mathbf{S}$ of a given data set from rank one matrices built from eigenvectors and eigenvalues of the covariance matrix. Usually one works with normalized data and correlation matrices. The presentation in this section is based also on normalized data.

The traditional principal component analysis yields the following decomposition ($\mathbf{S}$ may denote a covariance or a correlation matrix):

$$\mathbf{S} = \sum_{i=1}^{p} \lambda_i \mathbf{a}_i \mathbf{a}_i^T = \sum_{i=1}^{p} \mathbf{A}_i,$$

with $\lambda_1, \ldots, \lambda_p$ and $\mathbf{a}_1, \ldots, \mathbf{a}_p$ denoting the eigenvalues and eigenvectors of $\mathbf{S}$ appropriately.

The diagonal of $\mathbf{S}$ contains the *variances* of the considered variables. Thus, the above formula tells us, how – and to what extent – the variances of the considered variables can be reconstructed from subsequent principal components.

For our data – when taking into account 3 principal components only, i.e. PC1, PC2, PC3 – we obtain quite a high reconstruction of the total variance: the reconstruction amounts 50.10% when using only PC1; 72.11% when using both PC1 and PC2; and 83.72% when using all three of them, i.e. PC1, PC2 and PC3.

The reconstruction of the individual variances of the variables 1, …, 9 is respectively (values given as fractions of Total=1.00):
by PC1 and PC2 : 0.54, 0.82 ,0.64 ,0.73 ,0.64, 0.86, 0.76, 0.75, 0.75,
by PC1, PC2 and PC3: 0.81, 0.95 ,0.72, 0.97, 0.64, 0.87, 0.88, 0.79, 0.90.

Looking at these results one may state that the representation of the data points in 3 dimensions is already satisfactory, except for the variables no. 3 and no. 5.

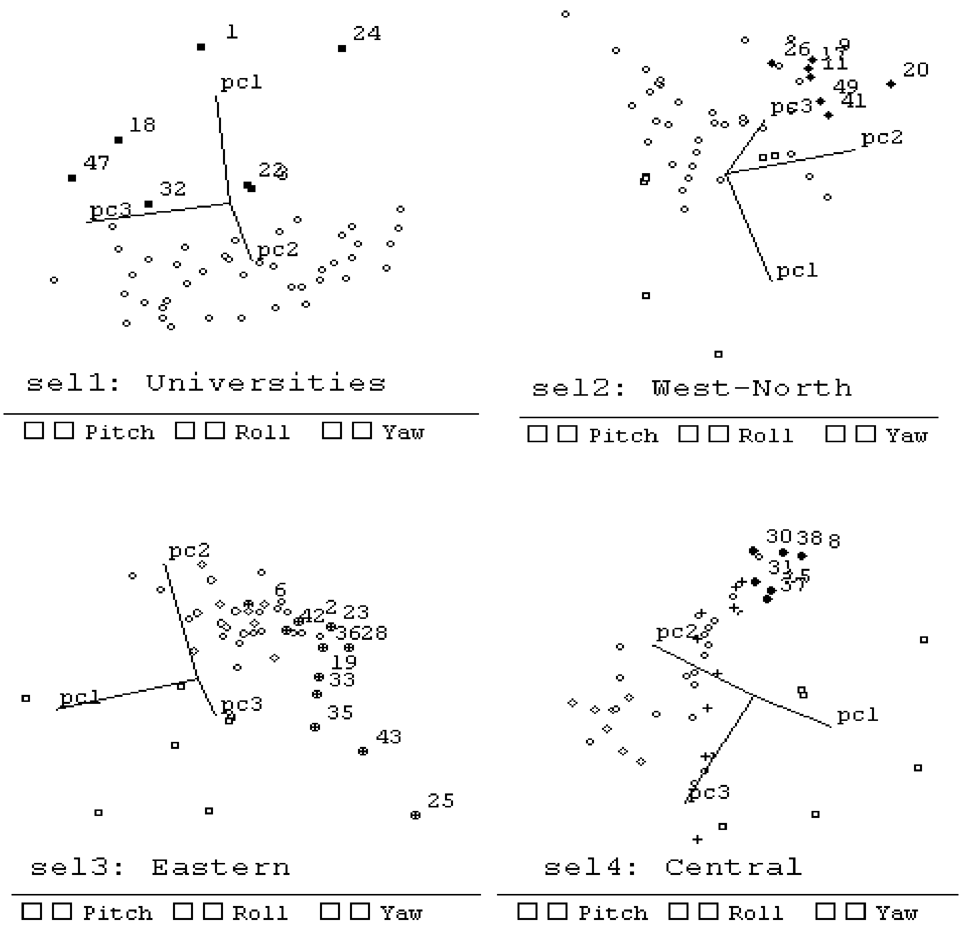

Figure 4. Spin plots from principal components PC1, PC2, PC3 – illustrating the identified selections `sel1, sel2, sel3, sel4`*. The three PC's explain together 83.72% of the total variance. Considered individually for the variables the three used principal components explain respectively 81, 95, 72, 97, 64, 87, 88, 79, 90 percentages of total variance of subsequent variables, therefore the representation in the spin plots – except perhaps for the variables no. 5 and no. 3 – is quite good and trustworthy*

In this situation we may hope that the display in a spin plot will reveal all the most prominent features of the similarities among the administrative units – except their characteristics given by variable no. 3 (divorce rate) and 5 (employment in industry).

Using an interactive spin plot we may investigate, whether the selections obtained on the basis of Kohonen maps correspond really to neighboring points in the spin plots.

Momentary views of the spin plots constructed from the first three principal components are shown in Figure 4. The plots exhibited in that figure confirm that our selections were right. The points belonging to one selection stick together – what can be seen when rotating appropriately the 3-dimensional spin plot.

5 Interesting alternatives of visualization

Interesting alternatives to principal coordinates are Sammon's mapping and approximations based on latent factors.

Sammon mapping seeks for coordinates $\mathbf{y} = (y_1, \ldots, y_h)$, $h = 2$ or $h = 3$ such that distances between mapped points $\mathbf{y}_i, \mathbf{y}_j$ i.e. the distances $d_{ij} = d(\mathbf{y}_i, \mathbf{y}_j)$ reflect the original distances $d^*_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ between the points $\mathbf{x}_i, \mathbf{x}_j$ located in R^p. The error function - to be minimized - is:

$$E = \frac{1}{c} \sum_{i<j}^{n} \left| d^*_{ij} - d_{ij} \right|^2 / d^*_{ij}, \tag{1}$$

with

$$c = \sum_{i<j}^{n} d^*_{ij}. \tag{2}$$

Usually the distance is defined as Euclidean distance:

$$d_{ij} = \sqrt{\sum_{k=1}^{M} |z_{ik} - z_{jk}|^2}, \tag{3}$$

with M denoting the dimension of the space, in which the points $\mathbf{z}_i, \mathbf{z}_j$ are located.

In this layout the mapped points $\mathbf{y}_i, \mathbf{y}_j$ are obtained as the result of an iterative minimization procedure; Sammon has used for this purpose a simplified algorithm based on Newton's method.

Latent factor analysis assumes that the observed values $\mathbf{x} = (x_1, \ldots, x_p)$ are results of a realization in them of some directly non-observable (hidden) factors $\mathbf{u}_1, \ldots, \mathbf{u}_h$, $h < p$.

Assuming, that the hidden factors are normalized ($\mathbf{u} \sim N(\mathbf{0}, \mathbf{I})$) we have the model equation:

$$\mathbf{x} = \mathbf{u}\mathbf{W} + \boldsymbol{\mu} + \boldsymbol{\epsilon},$$

in which $\mathbf{u}$ and resulting from the model observed vector $\mathbf{x}$ are specific for the given individual (unit), $\mathbf{W}$ and $\boldsymbol{\mu}$ are parameters of the model to be estimated from the data, and $\boldsymbol{\epsilon}$ represents random error (noise). Usually one assumes, that the error is distributed normally: $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Psi})$ with $\boldsymbol{\Psi}$ diagonal.

Tipping and Bishop have recently published a paper (Tipping and Bishop, 1999), in which they show how the principal axes of a set of observed data vectors may be determined through maximum likelihood estimation of parameters in a latent variable model which is closely related to factor analysis. They have also elaborated an EM algorithm for estimating the principal subspace iteratively.

We do not present here the results of applying that algorithm.

Generally, it follows from our experience, that computing using deterministic models (like PCA) is much faster, none-the-less, because of assuming, that the observed data are noise–free, the models become overfitted and have less generality, when applied to other data.

The latent factor or probabilistic principal components take into account also the variability of the data and are therefore more appropriate for most analyses. Very often, the derived latent factors, have a realistic and usable interpretation.

Apart from Kohonen's maps, recently Anouar *et al.*(Anouar, Badran and Thiria, 1997) have make the remark, that Kohonen maps are a special case of a more general method developed earlier by Diday (Diday, 1971; Diday et collaborateurs, 1980) and called **les nuées dynamiques** (dynamic kernels). May be this path is worth of pursuit.

6 Conclusions and Final Remarks

Visualization by Kohonen's self-organizing maps appeared to be the most fundamental method in our study. The method works quickly also for large data sets. However it does not yield unique results and it is safer to confirm the results by some other methods. It is a machine method which needs further human interaction. In the paper we have shown, how further analysis, carried out by man, may supplement the results obtained from a formal machine analysis.

The interactive interface to the results yielded by self-organizing maps should contain easy-to-use and quick access to every information on the data points, without compiling anew the source code of the macros producing the results.

This can be done using such interactive systems like XLispStat, which works like an interpreter, and many graphical functions may be accessed and executed simply by a mouse-click.

Acknowledgements

The research was partially sponsored by the grant *no.* 8 T11 C031 16 obtained from KBN, the Polish State Research Committee.

The paper has used data gathered in the Institute of Cardiology, Division of Biometry, Warsaw–Międzylesie. The data were used in a former joint publication (Bartkowiak and Kupść, 1955).

Figure 1 was obtained by use of SOM_PAK, a software programmed by T. Kohonen and his team (Kohonen, Hynninen and al. 1996).

The remaining graphics was programmed by the author using XLispStat (Tierney, 1990) .

References

[1] Anouar F., Badran F., and Thiria S. (1997). Cartes topologiques et nuées dynamiques. In: S. Thiria et al. (Eds), *Statistiques et méthodes neuronales.* Dunod, Paris, 190–206.

[2] Bartkowiak A. (2001). A semi-stochastic grand tour for identifying outliers and finding a clean subset. *Biometrical Letters* **38**, No 1, in print.

[3] Bartkowiak A. and Szustalewicz A. (2000). Outliers – finding and classifying which genuine and which spurious. *Computational Statistics*, **15**, no 1, 3–12.

[4] Bartkowiak A. and Kupść W. (1995). Regression Modeling of the Polish Mortality Data 1989–1991. Mortality of men. *Biometrical Letters*, **32**, Nb 2, 61-80.

[5] Diday E., et Collaborateurs (1980). Optimisation en Classification Automatique, tome 1–2, INRIA, Le Chesnay.

[6] Diday E. (1971). La mèthode des nuées dynamiques. *Rev. Stat. Appliquée*, **19**, No 2, 19–34.

[7] Fayad U.M. and Smyth P. (1999). Cataloging and Mining Massive Datasets for Science Data Analysis. *Journal of Computational and Graphical Statistics,* **8**, Nb. 3, 589–610.

[8] Huh M.Y. and Kiyeol K. (1999) Visualization of multivariate data using modifications of Grand Tour. *ISI'99, Invited Paper,* Helsinki.

[9] Kohonen T. (1997). *Self-Organizing Maps.* Springer, Berlin (1995), 1st Edition, (1997) 2nd Edition.

[10] Kohonen T., Hynninen J., Kangas J., and Laaksonen J. (1996). *SOM_PAK, The Self-Organizing Map Program Package*, Report A31. Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland. Also available from the Internet at the address `http://www.cis.hut.fi/nnrc/nnrc-programs.html`

[11] Tierney, L. (1990). *LISP–STAT, an Object–Oriented Environment for Statistical Computing and Dynamic Graphics.* Wiley, New York (1990). `http://www.stat.umn.edu/~luke/xls/xlsinfo/`

[12] Tipping M.E., Bishop Ch.M. (1999) Probabilistic principal component analysis. *J. R. Stat. Soc. B*, **61**, Part 3, 611–622.

[13] Unwin A. (2000). Using your eyes – making statistics more visible with computers. *Computational Statistics and Data Analysis* **32**, Nb. 3–4, 303–312.

[14] Vesanto J. (1999): SOM–based data visualization methods. Intelligent Data Analysis, **3**, Nb. 2, 111–137.

[15] Wegman E.J. (2000). On the eve of the 21st century: statistical science at a crossroads. *Computational Statistics and Data Analysis* **32**, Nb. 3–4, 239–243.

[16] Wegman E.J., Poston W.L., Solka J.L. (1998). Image grand tour. *Technical Report no. 150*, George Mason University, The Center for Computational Statistics, April 1998.