VISUALIZATION OF DATA BY METHOD OF ELASTIC MAPS AND ITS APPLICATIONS IN GENOMICS, ECONOMICS AND SOCIOLOGY

A.N. GORBAN and A.Yu. ZINOVYEV

Institut des Hautes Études Scientifiques
35, route de Chartres
91440 – Bures-sur-Yvette (France)

Août 2001

IHES/M/01/36

VISUALIZATION OF DATA BY METHOD OF ELASTIC MAPS AND ITS APPLICATIONS IN GENOMICS, ECONOMICS AND SOCIOLOGY

Gorban A.N., Zinovyev A.Yu.

Institute of Computational Modeling,
Siberian Branch of Russian Academy of Science, Russia

Institute des Hautes Etudes Scientifiques, France

E-mail: gorban@icm.krasn.ru; auranic@online.ru;

Abstract

Technology of data visualization and data modeling is suggested. The basic of the technology is original idea of *elastic net* and methods of its construction and application. A short review of relevant methods has been made. The methods proposed are illustrated by applying them to the real biological, economical, sociological datasets and to some model data distributions.

1 Introduction

We live in the epoch of exponential growth of information. Internet, Intranet nets, local computers contain a huge amount of information databases and their number grows continually. Often such databases become a kind of “information tombs” because of the difficulties with their analysis.

The basic property of the information is its multidimensionality. Rather than 2-3 a typical object in database has hundreds and thousands features. Because of this information loses its clearness and one can’t represent the data in visual form by standard visualization means – graphs and diagrams.

In this paper a technology of visual representation of data structure is described in details. It turns out that many problems concerning data analysis could be solved, at least qualitatively, using visual two-dimensional (or three-dimensional) picture of data and laying on it additional relevant information. This data image should display cluster structures and different regularities in data.

The basic of the technology is original idea of *elastic net* – regular point approximation of some manifold that is put into the multidimensional space and has in a certain sense minimal energy. This manifold is an analogue of principal surface and serves as non-linear screen on what multidimensional data are projected.

Remarkable feature of the technology is its ability to work with and to fill gaps in data tables. Gaps are unknown or unreliable values of some features. It gives a possibility to predict plausibly values of unknown features by values of other ones. So it provides technology of constructing different prognosis systems and non-linear regressions.

The technology can be used by specialists in different fields. There are several examples of applying the method presented in the end of this paper.

ACKNOWLEDGEMENTS

The paper and developed software were discussed with Prof. Misha Gromov and Prof. Alessandra Carbone and we thank for several valuable remarks and good critisism.

Computer software realizing the methods was developed in tight collaboration with Dr. Alexander Pitenko. Because of his participation it became much more user-friendly and logically structurised.

We would like to thank Victor Uskin who helped us to improve English language in the paper.

2 Visualization of Multidimensional Data

As a rule, it is possible to present a database in the form of a big numerical table with “object-feature” structure. A row of such a table contains information about one object, and set of columns contains various numerical features of the object. Providing the set of the features is identical for all of the objects in the table, we may think of the table as a finite collection of points in R^M , where M is the number of features in the set.

Introducing Euclidean metric in the space, we get *geometrical metaphor* of the table of data. In this space close points correspond to the objects with similar properties.

Characteristics of such a cloud of multidimensional points correspond to the empirical laws that could be extracted from a dataset. There are a lot of methods to analyze the cloud. Most of them return as a result some numerical values.

Yet it would be very useful to have a possibility to “glance” over a collection of points to form a clear visual picture of data. Visual presentation of a new dataset helps choose an adequate instrument for further quantitative analysis, and promotes deeper insight in the results of investigation.

The problem is the human brain that can’t operate efficiently with objects with dimension greater than three. “Ancient” scientific instruments of data presentation – graphs and diagrams – may help show mutual dependence of two, maximum three variables.

Recently a great amount of efforts in practical statistics are given to the methods of data visualization with the help of *dimension reduction*. The most understandable image of data can be obtained from its presentation on two-dimensional picture.

Actually, the method of dimension reduction is a classical approach, but its concrete implementation varies in many ways. Linear methods are most popular in statistics because of their clearness. But in many practical cases these methods are unsatisfactory so developing non-linear methods is really actual problem.

The method of dimension reduction may be justified by several reasons. Most of the point collections taken from the real data tables form in multidimensional space a structure whose effective dimension is smaller (sometimes much smaller) than the dimension of the space. This effective dimension of the structure is determined by the number of significant factors that had influence on the system of objects observed. Even if the cloud of points could not be placed in a linear low-dimensional subspace, it may have low-dimensional non-linear structure (for example, it may be distributed in the neighborhood of curved manifold).

If the structure has higher dimension then three, then one must reconcile to distortions appearing from projecting to the space of lower dimension. On this venue there are several methods that allow to point out regions where mapping gives rise to severe distortions.

3 Methods of Data Visualization

At present several methodological approaches exist concerning the problem of data visualization. In this section we just point out to the three of them.

In the methods of *multidimensional scaling* mapping $P: R^M \to R^2$ is to be found that minimizes some functional calculated for the initial M-dimensional coordinates of the points and for the resulting 2-dimensional coordinates:

$$Q(x^{(1)}, x^{(2)} ... x^{(N)}, \hat{x}^{(1)}, \hat{x}^{(2)} ... \hat{x}^{(N)}) \rightarrow \min ,$$

where $x^{(i)}$ is the radius-vector of the i-th point in R^M, $\hat{x}^{(i)}$ is the radius-vector of the same point in the resulting 2-dimensional space, N is the number of points. If the mapping P is assumed to be arbitrary, then functional Q is minimized by varying values of $\hat{x}^{(i)}$, $i = 1..N$, using procedures of gradient optimization.

In the methods of *factor analysis* it is necessarily to find such combinations (usually linear) of the initial coordinates that using them as new coordinate system would make possible to “model” coordinates of given points with allowable accuracy, i.e.

$$x^{(i)} = \sum_{j=1}^{k} \alpha_j^{(i)} f^{(j)} + q ,$$

where $f^{(j)}$ is the factor, that actually is a vector in R^M, k is the number of factors, q is random variable that is not dependent on i.

Choosing type of the “description error” to be minimized one gets different variants of factor analysis. In some cases it is rational to project points of data onto the plane spanned by first two *principal components* – directions in the data space, where dispersion of the cloud of points is maximal.

A different ideology lies in the method of *Kohonen Self-Organized Maps* (SOM). In this method *a net of ordered nodes* is placed in the data space. It is common practice to use rectangular or hexagonal two-dimensional net. In the learning process positions of the nodes are adjusted. During one act of adjustment one of the points and the nearest node of the net are selected. Then this node make a small step toward the data point. Besides some of its neighboring nodes on the net are moved in the direction of the selected point. As a result we have a net of nodes that is placed in the multidimensional space and approximates the cloud of data points. In the regions where the data points concentrate the density of nodes is higher and vise versa. In addition the net tends to be “regular” and tries to reconstruct the form of the cloud.

After that the net is unfolded on a plane (it is possible because it was initially two-dimensional) and the data are visualized with the help of different methods (Sammon maps, Hinton diagrams etc., see Kohonen, 1996).

4 Constructing Elastic Net

Basic algorithm

Method of elastic maps, similar to SOM, for approximation of the cloud of data points uses an ordered system of nodes, that is placed in the multidimensional space.

Lets define *elastic net* as connected unordered graph $G(Y,\boldsymbol{E})$, where $Y = \{y^{(i)}, i=1..p\}$ denotes collection of graph nodes, and $\boldsymbol{E}=\{E^{(i)}, i=1..s\}$ is the collection of graph edges. Let's combine some of the adjacent edges in pairs $R^{(i)} = \{E^{(i)},E^{(k)}\}$ and denote by $\boldsymbol{R}=\{R^{(i)}, i=1..r\}$ the collection of *elementary ribs*.

Every edge $E^{(i)}$ has the beginning node $E^{(i)}(0)$ and the end node $E^{(i)}(1)$. Elementary rib is a pair of adjacent edges. It has beginning node $R^{(i)}(1)$, end node $R^{(i)}(2)$ and the central node $R^{(i)}(0)$ (see Fig. 1).

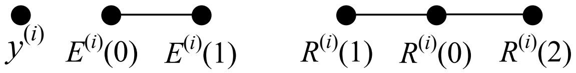


Fig 1: Node, edge and rib

Figure 2 illustrates some examples of the graphs practically used. The first is a simple polyline, the second is planar rectangular grid, third is planar hexagonal grid, forth – non-planar graph whose nodes are arranged on the sphere (spherical grid), then a non-planar cubical grid, torus and hemisphere. Elementary ribs at these graphs are adjacent edges, that subtend a blunt angle.

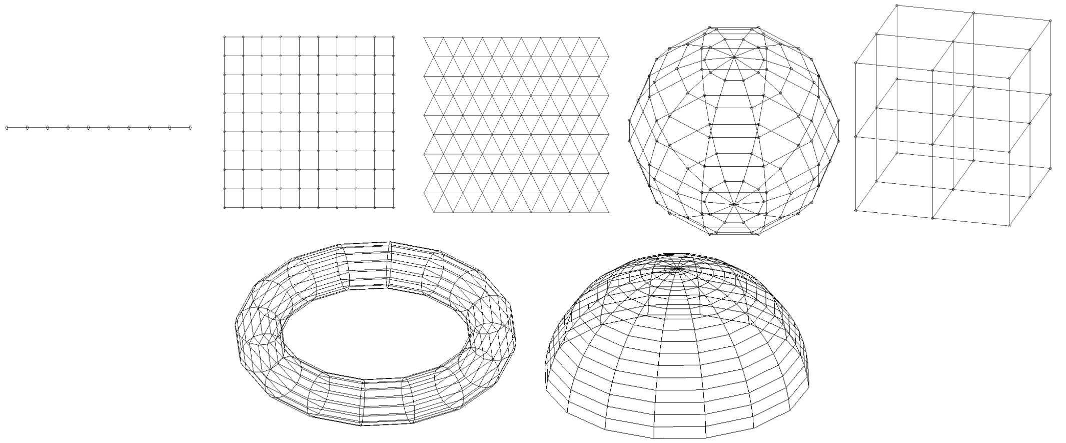


Fig 2: Elastic nets used in practice

Let's place nodes of the net in a multidimensional data space. This can be done in different ways, placing nodes randomly or placing nodes in a selected subspace. For example, it can be placed in the subspace spanned by first two or three principal components. In any case every node of the graph becomes a vector in R^M.

Then we define on the graph G energy function U that summarize energies of every node, edge and rib:

$$U = U^{(Y)} + U^{(E)} + U^{(R)}.$$

Let's divide the whole collection of data points into subcollections (called *taxons*) $K^{(i)}$, $i = 1\ldots p$. Each of them contains data points for which node $y^{(i)}$ is the closest one:

$$K_i = \{x^{(j)} : \left\| x^{(j)} - y^{(i)} \right\| \underset{Y}{\to} \min\}.$$

Let's define

$$U^{(Y)} = \frac{1}{N}\sum_{i=1}^{p} \sum_{x^{(j)} \in K^{(i)}} \left\| x^{(j)} - y^{(i)} \right\|^2,$$

$$U^{(E)} = \sum_{i=1}^{s} \lambda_i \left\| E^{(i)}(1) - E^{(i)}(0) \right\|^2,$$

$$U^{(R)} = \sum_{i=1}^{r} \mu_i \left\| R^{(i)}(1) + R^{(i)}(2) - 2R^{(i)}(0) \right\|^2 .$$

Actually $U^{(Y)}$ is the average square of distance between $y^{(i)}$ and data points in $K^{(i)}$, $U^{(E)}$ is the analogue of summary energy of elastic stretching and $U^{(R)}$ is the analogue of summary energy of elastic deformation of the net. We can imagine that every node is connected by elastic bonds to the closest data points and simultaneously to the adjacent nodes (see Fig. 3).

Values λ_i and μ_j are coefficient of stretching elasticity of every edge $E^{(i)}$ and coefficient of bending elasticity of every rib $R^{(j)}$. In a simple case we have

$$\lambda_1 = \lambda_2 = \ldots = \lambda_s = \lambda(s)\,,\ \mu_1 = \mu_2 = \ldots = \mu_r = \mu(r)\,.$$

To obtain $\lambda(s)$ and $\mu(r)$ dependences we simplify the task and consider the case of evenly stretched and evenly bend net. We introduce natural parameter u and vector function $F(u)$, then

$$U^{(E)} = \lambda(s)s\frac{\left\| E(1) - E(0) \right\|^2}{\Delta u^2}\Delta u^2 \approx \lambda(s)s\left\| E_u^{'} \right\|^2 \Delta u^2 \approx \lambda(s)\frac{s}{s^{2/d}}\left\| E_u^{'} \right\|^2 = \frac{\lambda(s)}{s^{\frac{d-2}{d}}}\left\| E_u^{'} \right\|^2$$

$$U^{(R)} = \mu(r)r\frac{\left\| R(1) + R(2) - R(0) \right\|^2}{\Delta u^4}\Delta u^4 \approx \mu(r)\frac{r}{r^{4/d}}\left\| F_{uu}^{"} \right\|^2 = \frac{\mu(r)}{r^{\frac{d-4}{d}}}\left\| F_{uu}^{"} \right\|^2$$

This simplified consideration shows that, if we require that elastic energy of the net remains unchanged in case of finer net, then

$$\lambda = \lambda_0 s^{\frac{2-d}{d}}\,,\ \mu = \mu_0 r^{\frac{4-d}{d}} \qquad (*)$$

where d is the "dimension" of the net ($d = 1$ in the case of polyline, $d = 2$ in case of hexagonal, rectangular and spherical grids, $d = 3$ in case of cubical grid and so on).

Now we will find such positions of the nodes of graph that it will have minimal energy. It is the elastic net to be constructed that approximates the cloud of data points and has some regular properties. Minimization of term $U^{(Y)}$ gives approximation, using $U^{(E)}$ provides more or less evenness of the net and $U^{(R)}$ makes the net "smooth", preventing it from strong folding and "twisting".

To start let's consider the situation when we separated collection of data points to taxons $K^{(i)}$, $i = 1\ldots p$.

Let's denote

$$\Delta(x,y) = \begin{cases} 1, x = y \\ 0, x \neq y, \end{cases}$$

$$\Delta E^{ij} \equiv \Delta(E^{(i)}(1), y^{(j)}) - \Delta(E^{(i)}(2), y^{(j)}),$$

$$\Delta R^{ij} \equiv \Delta(R^{(i)}(3), y^{(j)}) + \Delta(R^{(i)}(2), y^{(j)}) - 2\Delta(R^{(i)}(1), y^{(j)}).$$

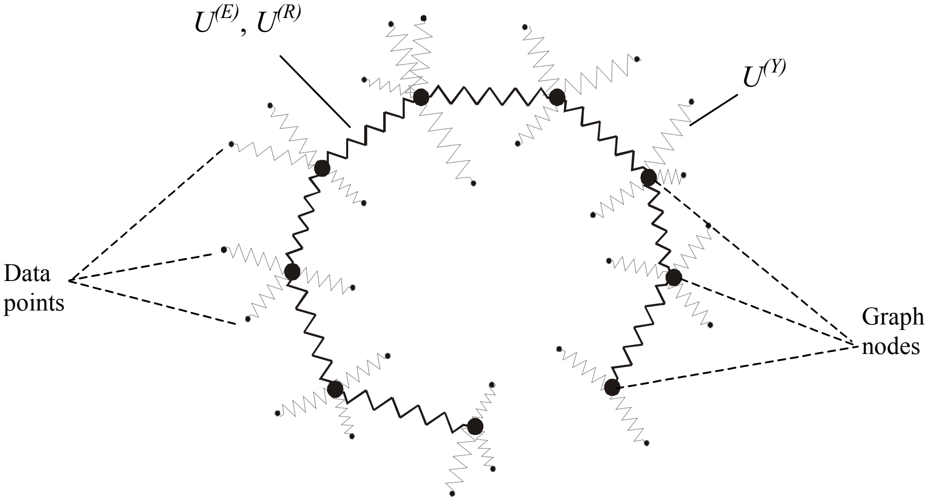


Fig. 3: Energy of elastic net

Then differentiation gives

$$\frac{1}{2}\frac{\partial U^{(Y)}}{\partial y^{(j)}} = n_j y^{(j)} - \sum_{x^{(i)} \in K^{(j)}} x^{(i)}$$

$$\frac{1}{2}\frac{\partial U^{(E)}}{\partial y^{(j)}} = \sum_{k=1}^{p} y^{(k)} \sum_{i=1}^{s} \lambda_i \Delta E^{ij} \Delta E^{ik} = \sum_{k=1}^{p} y^{(k)} e_{jk},$$

$$\frac{1}{2}\frac{\partial U^{(R)}}{\partial y^{(j)}} = \sum_{k=1}^{p} y^{(k)} \sum_{i=1}^{r} \mu_i \Delta R^{ij} \Delta R^{ik} = \sum_{k=1}^{p} y^{(k)} r_{jk},$$

where n_j is the number of points in $K^{(i)}$, $e_{jk} = \sum_{i=1}^{s} \lambda_i \Delta E^{ij} \Delta E^{ik}$, $r_{jk} = \sum_{i=1}^{r} \mu_i \Delta R^{ij} \Delta R^{ik}$. As a result we obtain

$$\frac{1}{2}\frac{\partial D}{\partial y^{(j)}} = \sum_{k=1}^{p} y^{(k)} \left(\frac{n_j \delta_{jk}}{N} + e_{jk} + r_{jk} \right) - \frac{1}{N} \sum_{x^{(i)} \in K_j} x^{(i)} = 0, \ j = 1 \dots p,$$

and the system of p *linear* equations to find new positions of nodes in multidimensional space $\{y^i, i=1...p\}$:

$$\sum_{k=1}^{p} a_{jk} y^{(k)} = \frac{1}{N} \sum_{x^{(i)} \in K_j} x^{(i)}, \text{ where}$$

$$a_{jk} = \frac{n_j \delta_{jk}}{N} + e_{jk} + r_{jk}, \ j = 1 \dots p, \qquad (**)$$

$$\delta_{jk} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

The values of e_{jk} and r_{jk} depend only on the structure of the net. If the structure does not change then they are constant, n_j depends on the separation of collection of data points to taxons.

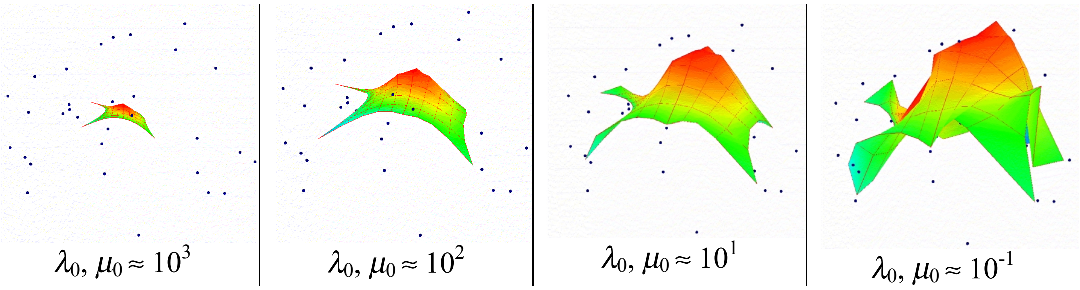
Therefore to minimize the energy of graph U, the following algorithm is effective:

1. Place the net of nodes in multidimensional space.
2. Given nodes placement, separate collection of data points to subcollections $K^{(i)}$, $i = 1 \dots p$.
3. Given this separation, minimize graph energy U and calculate new positions of nodes.
4. Go to step 2.

It is evident that this algorithm converges to the final placement of nodes of the elastic net (energy U is a non-decreasing value, and the number of divisions of data points into taxons is finite). Moreover, theoretically the number of iterations of the algorithm before converging is finite. In practice this number may be unacceptable, therefore we interrupt the process of minimization if changes of values of U become less than a small number ε.

To choose constants λ_0 and μ_0 in (*) is a non-trivial task. If we make λ_0 and μ_0 very large then we get a "squeezed" net, its nodes will concentrate in the vicinity of the geometrical center of the cloud of data points. Making λ_0 and μ_0 too small leads to a very irregular net, it will be strongly twisted and its nodes will distribute in the space very unevenly.

Practice showed that it is useful to make λ_0 and μ_0 variable, using the procedure of "annealing". We trained nets in several epochs, starting from large values of λ_0, μ_0 (approximately 10^3) and finishing with small values ($\sim 10^{-1}$) (see Fig. 4). As a result we have a net, approximating the cloud of points, with rather evenly distributed nodes, arranged along a rather smooth *d*-dimensional surface. The process of "annealing" promises that the resulting net will realize the global minimum of energy *U* or rather close configuration.



$\lambda_0, \mu_0 \approx 10^3$ $\lambda_0, \mu_0 \approx 10^2$ $\lambda_0, \mu_0 \approx 10^1$ $\lambda_0, \mu_0 \approx 10^{-1}$

Fig.4: Training elastic net in several epochs

Adaptive elastic nets

We can change properties of the net to be constructed by varying coefficients λ_i, μ_i, number of nodes in the graph and by changing structure of the net. On this venue it is possible to construct adaptive nets whose structure suits the given point distribution.

Let's consider the case of a rectangular grid.

First suggest the algorithm of a growing adaptive net. After the net reached its energy minimum we can calculate the values of "tension" of vertical and horizontal rows in the rectangular grid:

$$\varepsilon_k = \sum_{E^{(i)} \in G_k} \lambda_i \left\| E^{(i)}(1) - E^{(i)}(0) \right\|^2 ,$$

where G_k are subcollections of edges, that form horizontal or vertical rows in the grid. In the row where ε_k is maximal we divide every edge in two (see Fig. 5,a). After that the procedure of energy minimization should be applied and so on. The process is stopped when value of $U^{(Y)}$ become smaller than the accuracy level.

The second algorithm constructs adaptive structure of the net. After the net reached its energy minimum we get values of n_j (number of points in the *j*-th taxon) and find the square of the net, whose total number of n_j in the taxons of its corners are maximal. Then we divide this square into smaller squares. Coefficients λ_i of new edges (whose length is twice smaller than the side of the initial square) are doubled (see Fig. 5,b). After that the procedure of energy minimization should be applied and so on. The process is stopped when the value of $U^{(Y)}$ becomes smaller than the accuracy level.

As a result those edges whose nodes lie in that regions of space where concentration of points is low become longer, and dispersion of values of n_j become smaller.

The third algorithm adapts values of λ_i so that the density of nodes in multidimensional space corresponds to the density of point distribution (the net become more uneven):

a. After the net reached its energy minimum we get values of n_j (number of points in the j-th taxon).
b. For every edge $E^{(i)}$ the total number of points in the taxons of $E^{(i)}(0)$ and $E^{(i)}(1)$ is calculated. Let's denote these values by $n_i^{(E)}$, $i = 1\ldots s$.
c. New values of λ_i are calculated:

$$(\lambda_i)' = \frac{s \times n_i^{(E)}}{\sum_{i=1}^{s} n_i^{(E)}} \lambda_i ,$$

where $(\lambda_i)'$ are the new values of the coefficients of elasticity.

d. The process continues until we have some equilibrium configuration.

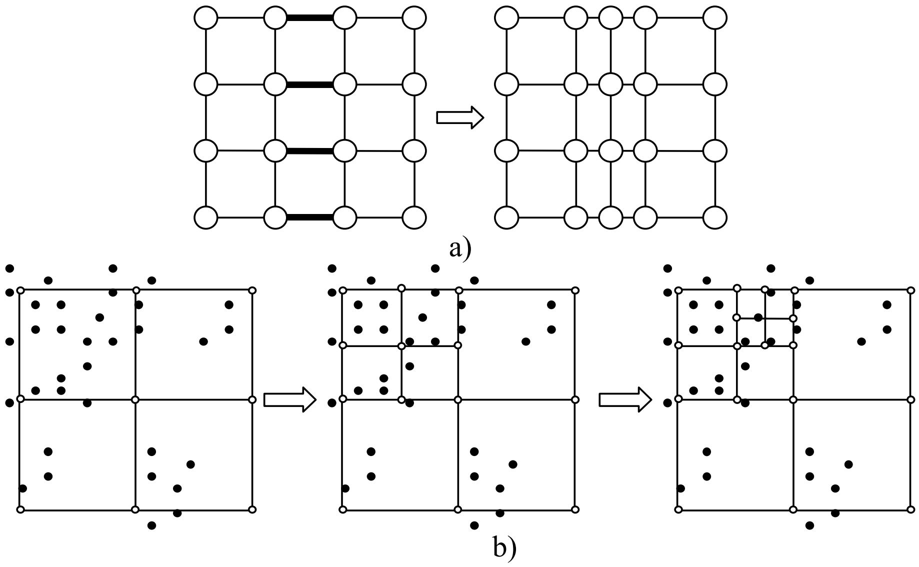


Fig. 5: Constructing adaptive nets

Teaching the net "online"

After the net is built on the basic collection of data, it can be required to allow to take into account new data coming in real time mode.

We could notice from (**) that in the matrix of the system only the first term $\frac{n_j\delta_{jk}}{N}$ depends on data. Actually for the net reconstruction we need to know only 'frequencies dictionary' – a set of sample vectors $\hat{y}^{(i)} = \frac{1}{n_i}\sum_{x^{(j)}\in K^{(i)}} x^{(j)}$ and frequencies $f_i = \frac{n_i}{N}$, $i = 1\ldots s$.

Now consider the case when a new data point *x'* (new row in the data table) appears. Lets think first that we have no full information about data distribution, we just have frequencies dictionary. Then new point slightly corrects the dictionary – increases the value of n_j of that sample $\hat{y}^{(j)}$, that is the closest one to *x'*, and shifts this sample at a small step:

$$\Delta\hat{y}^{(j)} = \frac{x' - \hat{y}^{(j)}}{n_j + 1}.$$

As a result, we have the matrix of the system (**) changing. We get new system $(A + \Delta A)(y + \Delta y) = (b + \Delta b)$, and equation for correction of node positions:

$$(A + \Delta A)\Delta y = \Delta b - \Delta A y\,.$$

Matrix ΔA has one non-zero element *1/N* in the *j*-th row and *j*-th column. Δb is vector with one non-zero *j*-th element $\frac{\Delta\hat{y}^{(j)}(n_j + 1) + \hat{y}^{j}}{N}$. So we have a system to calculate corrections of node positions:

$$\sum_{i=1}^{p}(a_{ij} + \frac{1}{N}\delta_{ij}\delta_{jk})\Delta y^{(i)} = \frac{\Delta\hat{y}^{(k)}(n_k + 1) + \hat{y}^{(k)} - y^{(k)}}{N}\delta_{jk}\,,\ j = 1..p,$$

We can solve the system for every component of the vectors and get new node positions $\left(y^{(i)}\right)' = y^{(i)} + \Delta y^{(i)}$. Although we can assume that just one node, the closest to *x'* is shifted, others stay at rest. In this case we have the following scheme of recalculation of nodes:

a. Find node $y^{(k)}$ closest to *x'*.
b. Correct the dictionary:

$$\left(\hat{y}^{(k)}\right)' = \hat{y}^{(k)} + \Delta\hat{y}^{(k)} = \frac{x' + \hat{y}^{(k)} n_k}{n_k + 1},\ \left(n_k\right)' = n_k + 1 .$$

c. Calculate the new position of $y^{(k)}$:

$$\left(y^{(k)}\right)' = y^{(k)} + \Delta y^{(k)} = \frac{y^{(k)} a_{kk} N + x'}{a_{kk} N + 1} .$$

d. Recalculate matrix: $\left(a_{kk}\right)' = a_{kk} + {}^{1}\!/\!_{N}$.

e. Magnify N: $N = N + 1$.

If we have full information about the data points then we can take into account the data points jumping from one taxon to another:

f. Enumerate all points in K_k and check for every point if the $y^{(k)}$ is still the closest node. If not then the point jumps to another taxon.

SOM and elastic net

What are the features of elastic nets compared to the methods of SOM?

First, compared to the original SOM algorithm distribution of nodes in the space is more regular and more controllable during the teaching process. This fact can be used to apply the elastic net as a point approximation of some manifold (see next section). Mapping data points onto this manifold is more isometric than in the case of SOM.

Second, the form of resulting manifold does not depend strongly on the number of nodes m, unlikely to SOM. Magnifying m, we just make the point approximation more accurate.

Third, given values λ_i, μ_i, the result of net constructing is optimal because of the minimum of energy of the graph, unlike SOM, where the net is just a result of a stochastic process without optimality criterion.

Fourth, the speed of training of elastic net promises to be higher as compared to SOM. In addition the training process may be easily made highly parallel.

Resume

In this section we developed a method of constructing an elastic net. An elastic net is a net of nodes that approximates multidimensional data points in some regular way. Nodes of the net lie on a kind of a smooth surface with minimal energy. This surface is like a film stretched in the data space.

The nets used in practice have different dimensions and even different topologies. In some cases it is useful to construct nets with spherical or other non-trivial topology.

The method allows to construct nets that have adaptive structure. The structure of the net may depend on different factors such as the density of points in the space and so on.

The net may can be constructed on the basic set of points and then be corrected taking into account new real-time information.

5 Building Elastic Map

Piecewise linear map

Elastic map is a manifold constructed in a multidimensional space using an elastic net as point approximation.

Formally the task is as follows (consider two-dimensional case). We need to reconstruct vector-function $r = r(u,v)$ using its values in a definite set of points $\{y_i = r_i(u_i, v_i), i = 1\ldots p\}$. A pair of values u_i, v_i are internal coordinates of the nodes in R^2.

There are many possible ways to construct elastic map. One of the simplest one is to construct piecewise linear manifold. Let's show how to do it.

Let's introduce a *triangulation* of the graph. We define a collection of elementary simplexes $\{T^{(k)}\}$, $k = 1\ldots t$. In case of polyline ($d = 1$) they are simple linear segments. In case of $d = 2$ they are triangles. And in case of $d = 3$ they are tetrahedrons. For clarity we limit ourselves with the case of $d = 2$.

Let's define internal coordinates of the point of the map (u and v in case of $d = 2$). Then using the triangulation we can find the simplex, for which the selected point belongs (it is the closest one). Let's denote numbers of nodes that are corners of this triangle as i_1, i_2, i_3. We can calculate coordinates α, β of the selected point relatively to these nodes:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_{i1} \\ v_{i1} \end{pmatrix} + \alpha \begin{pmatrix} u_{i2} - u_{i1} \\ v_{i2} - v_{i1} \end{pmatrix} + \beta \begin{pmatrix} u_{i3} - u_{i1} \\ v_{i3} - v_{i1} \end{pmatrix}.$$

Then we have equations for finding α and β:

$$\alpha \begin{pmatrix} u_{i2} - u_{i1} \\ v_{i2} - v_{i1} \end{pmatrix} + \beta \begin{pmatrix} u_{i3} - u_{i1} \\ v_{i3} - v_{i1} \end{pmatrix} = \begin{pmatrix} u - u_{i1} \\ v - v_{i1} \end{pmatrix}.$$

Solving it we find the value of the vector-function:

$$r(u,v) = y^{(i1)} + \alpha\left(y^{(i2)} - y^{(i1)}\right) + \beta\left(y^{(i3)} - y^{(i1)}\right).$$

As a result we get faceted surface, an example is shown in Fig. 6.

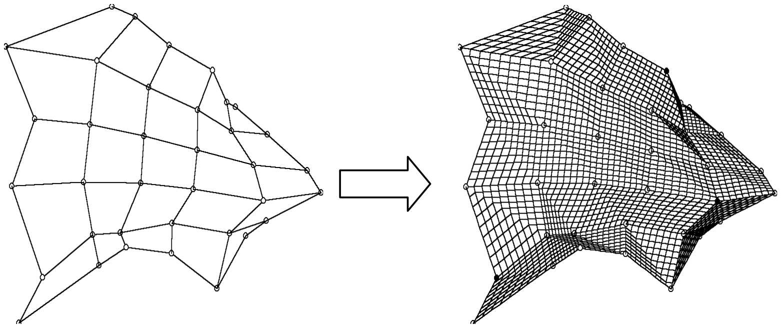


Fig. 6: Piecewise linear manifold

Constructing this piecewise linear manifold is the least labor-intensive method. Although we may use more intricate approaches (like multidimensional Karleman formula). Then as a result we get a continuous smooth surface and all nodes lie on it.

Projecting Data Points onto Map

In order to have a possibility to analyze data points on a plane, we need to project them onto the constructed two-dimensional manifold. Method of SOM makes use of piecewise constant projecting. It means that every point is transferred to the nearest node.

Because of the regularity of elastic net it is reasonable to apply piecewise linear projecting. We will project a point in the closest point of the *map* (unlikely to SOM where projecting is performed in the closest node of the net).

Lets introduce a *distance from the point to the segment of line*. We will calculate it in this way:

1. Project orthogonally on the line to which the segment belongs. If the projection is on the segment, then the result is the distance to the projection.
2. Otherwise the result is the distance to the closest corner of the segment.

Distance from the point to the triangle is calculated analogously:

1. Project orthogonally on the plane to which the triangle belongs. If the projection is on the triangle, then the result is the distance to the projection.
2. Otherwise the result is the distance to the closest side of the triangle (every of them is a segment).

Distance from the point to the tetrahedron:

1. Project orthogonally in the three-dimensional subspace to which the *tetrahedron* belongs. If the projection is in the tetrahedron, then the result is the distance to the projection.
2. Otherwise the result is the distance to the closest side of the tetrahedron (every of them is a triangle).

Analogously we can find the distance from the point to any *k*-dimensional simplex.

One-dimensional map consists of line segments, so the closest point of the map is the closest point of the closest segment. Analogously the closest point of the two-dimensional map is the closest point of the closest triangle and so on.

Note here that in the proposed way of projecting, there exist whole regions of space, from that every point is projected in a node. It may be considered as a principal drawback of such projecting. In case of one-dimensional map other ways of projecting are known, they are free from this defect (see Gorban, Rossiev, 1999).

Piecewise linear projecting gives a more detailed picture of data than in the case of using Hinton diagrams (see Fig. 7).

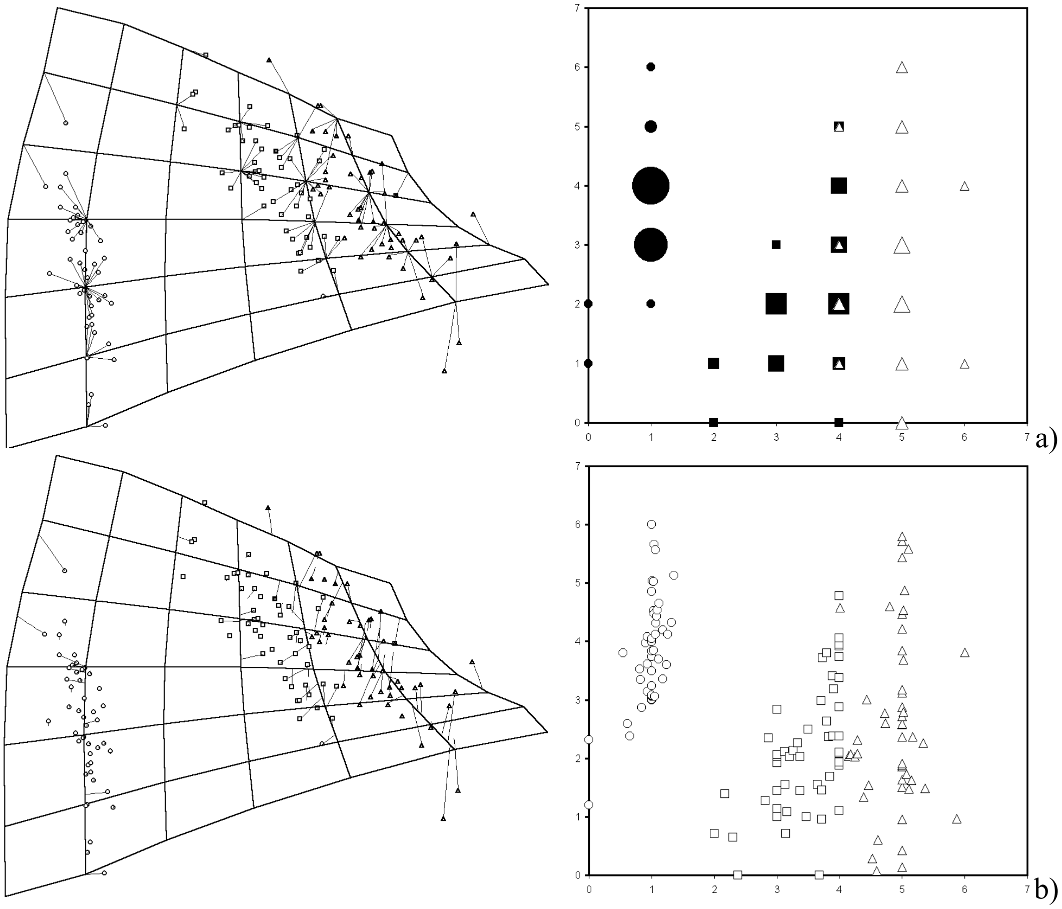


Fig. 7: Comparing piecewise constant (a) and piecewise linear (b) projecting.

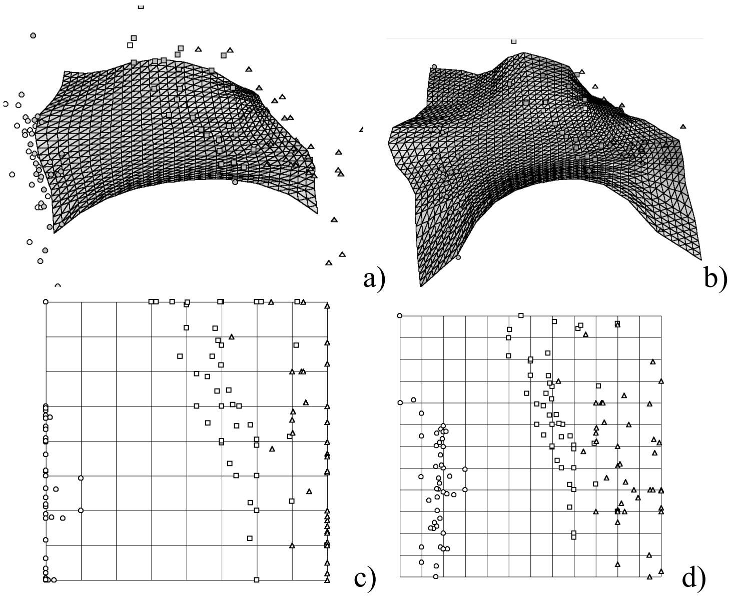


Fig. 8: Extrapolation of the map;
a – initial map, b – extrapolated map,
c – initial picture of data, d – the picture after extrapolation

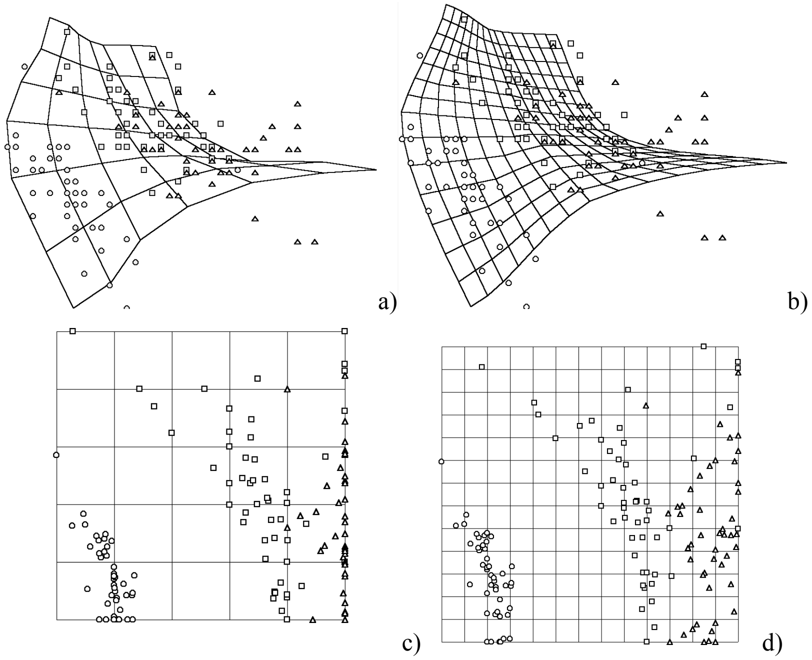


Fig. 9: Interpolation of the map;
a – initial map, b – interpolated map,
c – initial picture of data, d – the picture after interpolation

Extrapolation and Interpolation

The constructed manifold is principally bound. It lies in a cloud of points and there are relatively many points such that the closest point of the map is on the border of the manifold. Such “boundary effects” distort the picture of data points considerably. Thus we need to know how to extrapolate the map on its neighborhood. It is desirable for the manifold to slightly extend beyond the cloud of points.

The simplest case is the extrapolation of a piecewise linear map. The manifold is extended in linear fashion by gluing additional simplexes to its borders. The example of such a procedure with resulting picture of data is shown in fig. 8.

Analogously if we wish to make the grid finer then we can apply the procedure of interpolation of the map. There is an example of interpolation in fig. 9.

Mapping distortions

While mapping from the multidimensional space onto a curved low-dimensional manifold some inadequacy is unavoidable. There are two basic types of distortion: projecting distant points in multidimensional space in the close ones on the map (bad resolution of the mapping) and projecting close points in multidimensional space in the distant ones on the map (bad topology compliance), see Fig. 10,a.

In the work of Kivimoto, 1997 *topographical error* was introduced. Let’s take a data point, calculate the closest node of the graph and denote it as BMU (Best Matching Unit). Then determine the second-close node and denote it as BMU2. If BMU and BMU2 are not adjacent on the graph, then the data point is *unstable*. The value of topographical error is the ratio of the number of unstable points to the overall number of points.

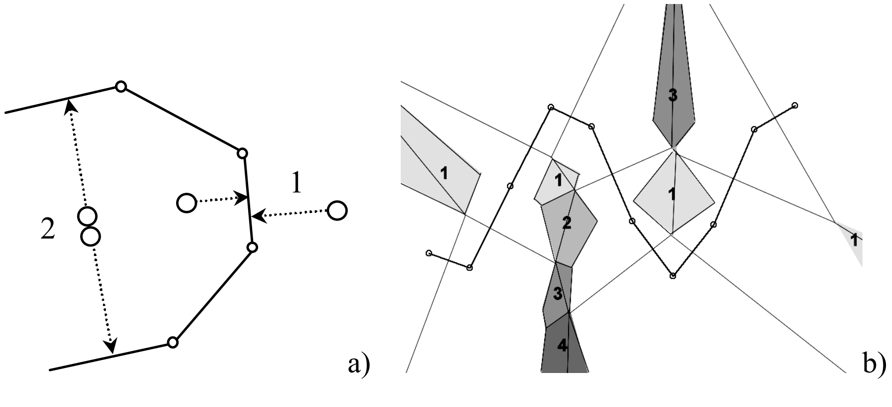


Fig. 10: a – distortions while mapping, 1 – bad resolution, 2 – bad topology accordance; b – areas of projection unstability.

In Fig. 10,b you can see where the areas of unstability are located in case of a one-dimensional manifold on the plane. The number inside the gray polygons denotes the degree of unstability – how many nodes separate BMU from BMU2.

Actually these areas themselves do not involve severe problems while projecting onto the map. Thin lines on the figure are the borders of Voronoi cells. If we have in the space a continuous trajectory, then projection of the trajectory has a break when it traverses the border that is situated in the area of unstability.

It is clear that in case of a "flat" map the topographical error is equal to zero (there are no unstabilty areas in this case).

Visualizing data points with gaps

In the given methods it is easy to manage with data points that have gaps – missing or unreliable values of some features. Actually real data tables almost always contain gaps. Distribution of gaps is sometimes crucial for using standard methods of statistics.

We can consider a data point that has value of the *i*-th feature unknown as a line parallel to the *i*-th coordinate axis. Then we can calculate for the line the closest point of the map just as we did it in a common situation (see Fig. 11).

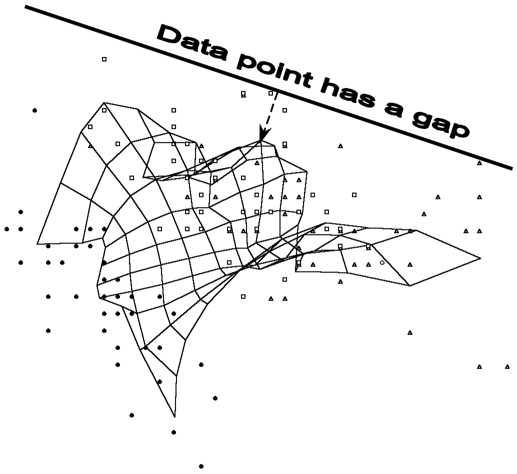


Fig. 11: Projecting data points having gaps

It is easy to show that formally we should calculate all distances and scalar products for this point in the subspace where all coordinates are known.

Resume

In this section we considered simple ways of constructing elastic map. It is a manifold that use elastic net as point approximation. The most simple way is to construct piecewise linear manifold. In this case we define on the net some triangulation.

Then the map serves as a non-linear screen on which we project data points. After that we can investigate data points in the space of internal coordinates of the low-dimensional manifold.

To avoid "boundary effects" extrapolation of the map is made. Procedure of interpolation helps to get finer and smoother map.

We can project onto the map data points having gaps. Formally we operate in this case in the subspace where all coordinates of the point are known.

6 Using Elastic Map

Colorings

Let's emphasize here that unlike the SOM technology and methods of multidimensional scaling we get as a result a low-dimensional bounded manifold put in multidimensional feature space. Every point of the manifold has from one hand M coordinates as a point in feature space, from other hand d "internal" coordinates (d – dimension of the manifold).

Thus we can show on the manifold values of any functional in R^M in the points of the manifold. What are the functionals that may be useful for coloring the map?

First, the simplest case is the use of values of coordinates. It gives M variants of the map colorings. Not all of these colorings are really useful. Some coordinates can be more significant than others. Using linear and non-linear methods of evaluating the level of feature significance we can choose most informative coordinate colorings.

If some coordinate colorings are similar (at least partially) it means that the corresponding features are correlated.

Fig. 12 shows how to visually estimate the accuracy of correspondence of coordinate coloring to the real values of the coordinate of the points on the map. In this picture areas of high values of the coordinate are shown as mountains, low values are shown as seas. The altitude of every point corresponds to the value of the coordinate of this point.

Second, on the map we can show values of some linear functional. Practice showed that it is very useful to illustrate on the map results of applying such traditional methods as linear discriminant and regression analysis. This gives an instrument for visual evaluation of the quality of applying these methods.

Third, on the map we can show values of more intricate functionals as an non-parametric estimate of density of data points distribution or density of some subcollection of points. It really helps to visually solve the task of cluster analysis or compare the results of applying numerical procedures of cluster analysis with real data structure (see fig. 13).

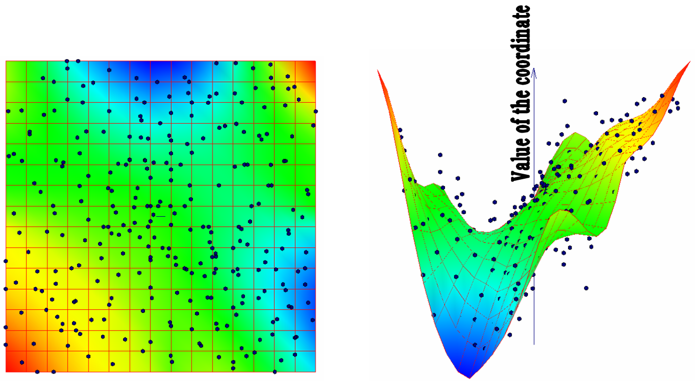


Fig.12: Making "3D-coloring"

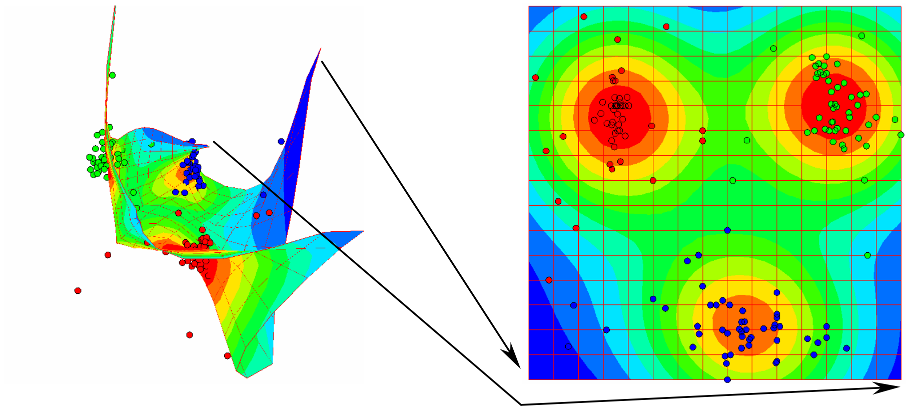


Fig.13: Visual estimating the procedure of cluster analysis

Non-linear regression (plausible gaps filling)

The map itself together with the procedure of projecting onto the map gives a possibility to construct regression of some coordinates of data space from the others.

Consider the two-dimensional case. In R^M we have map M and its vector-function $r = r(u,v)$: $R^2 \to R^M$, where u, v are internal coordinates, given in the finite connected domain in R^2. Besides we have mapping $P(x)$: $R^M \to R^2$ that assigns two coordinates u_x, v_x in the domain to point x from R^M. We could see that the mapping also applicable to the point x with some unknown coordinates.

If we consider the dependent coordinate of point x to be unknown then we can "recover" it by projecting x onto the map, getting internal coordinates u_x and v_x , and read the value in vector $r(u_x,v_x)$.

Iterative error mapping

It is evident that using two-dimensional manifold as a model of highly multidimensional data distribution leads to considerable error of approximation.

There are several ways to diminish the error.

First is to increase the number of nodes. *Second* is to make the map "softer" making values of elasticity coefficients lower. In both ways the map with better approximation becomes more "twisted" in the data space. It is a rather common situation in the methods of data approximation and it could be called "accuracy-regularity dilemma".

Other way is to construct rather regular ("rigid") map and to build a new space named "space of errors" with a collection of points, each of them representing the vector of error of mapping by the first constructed map. So every point in the space has a vector equal to

$$x_{err}^{(i)} = x^{(i)} - r(u_{x^{(i)}}, v_{x^{(i)}}),$$

where vector $r(u_x,v_x)$ is the projection $P(x)$ of data point x onto the map.

In the new space we can construct another regular map, getting regular model of errors of first map. If the accuracy of this model is not satisfactory then we can make new iteration and to continue to model errors until the quality becomes allowable. In such a manner we can model data points distribution with the error as small as desired.

So we have *the first* map that models the data distribution itself, *the second* map that models errors of the first model, *the third* map models errors of the second model and so on.

Let's denote mapping onto the first map in the data space by $P(x)$, mapping in the space of errors onto the second map by $P_2(x)$ and so on. So every point x in the initial data space is modeled by the vector

$$\tilde{x} = P(x) + P_2(x - P(x)) + P_3(x - P(x) - P_2(x - P(x))) + ...$$

If x is a vector with some coordinates "unknown", then $\tilde{x}$ is the vector with no gaps in coordinates.

This technology allows to make program systems with the ability to prediction just like they do with neural networks, but unlike the latter there is the possibility to deal with the data tables that have unknown cells (gaps), and the intrinsic possibility to make data visualization.

Resume

In this section we considered basic aspects of using elastic maps as a mean for visualization, analysis and modeling of data points.

Using colorings makes data map as informative as a geographical one. On the data map we can see cloud of data points with its structure, and different information layers may be shown as background. Using colorings we can represent on the map values of any multidimensional functional. For example, we can see results of any method of applied statistics – estimates of density distribution, functions of linear regressions, discriminant functions and so on.

Using the procedure of iterative error mapping we can approximate the data themselves, errors of the initial modeling and so on. Thus we can approximate data distribution with arbitrary accuracy. This procedure is especially helpful in case when there are gaps in the data – unknown or unreliable values of some coordinates.

7 Illustrations of the method applications

Simple model data distributions

Points on 3D sphere, points on pretzel

Fig.14,15 show simple point distributions with constructed elastic maps. The first is a set of points arranged on a three dimensional sphere, the second is a set of points forming a three dimensional pretzel. In both cases two variants of the grid were applied: two-dimensional rectangular (Fig.14b, 15a) and spherical (Fig.14c,15b). The figures show the resulting grids and projections.

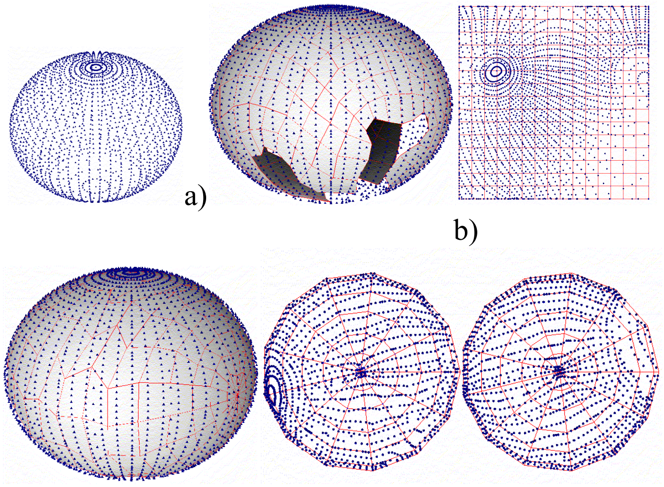
When linear mapping mistakes

Fig. 16a shows five-cluster data distribution. There are three massive and two small clusters in this simplified picture. Fig. 16b shows a linear mapping onto the plane of first two principal components. It is apparent that two small clusters are indiscriminable on the plane. The constructed elastic map and resulting mapping onto the map are shown in figures 16c and 16d. It is clearly now that there are five clusters in the point distribution rather then four.

Visualization of economical indicators

Let's illustrate the proposed methods by applying them to the visualization of the table of economical indicators of the biggest Russian companies. The table was taken from the russian magazine "Expert". The files of data were retrieved from the official site of the magazine: http://www.expert.ru. The table contains information of some economical characteristics of two hundred biggest companies in Russia, sorted in the order of decreasing of their gross production output. The following fields (only part of them are independent, others are calculated by explicit formula) were in the table:

1) Name of the company
2) Geographic region
3) Branch of industry the company belongs to
4) Gross production output in 1998
5) Gross production output in 1997
6) Rate of the growth
7) Gross production output in 1998, recalculated in dollar equivalent
8) Balance profit
9) Profit after taxation
10) Profitability
11) Number of workers
12) Efficiency of production

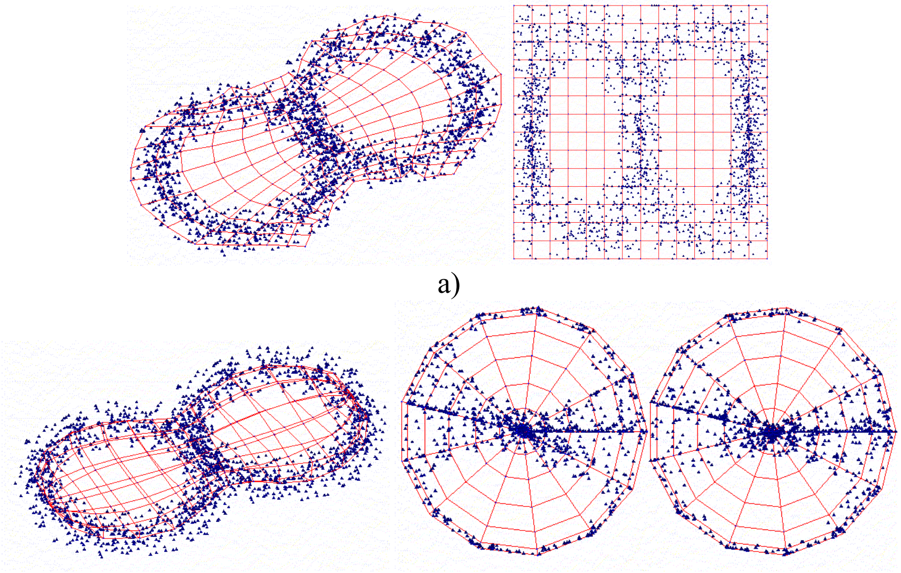


a)

b)

c)

Fig. 14: Distribution of points on the sphere



a)

b)

Fig. 15: Pretzel-like distribution

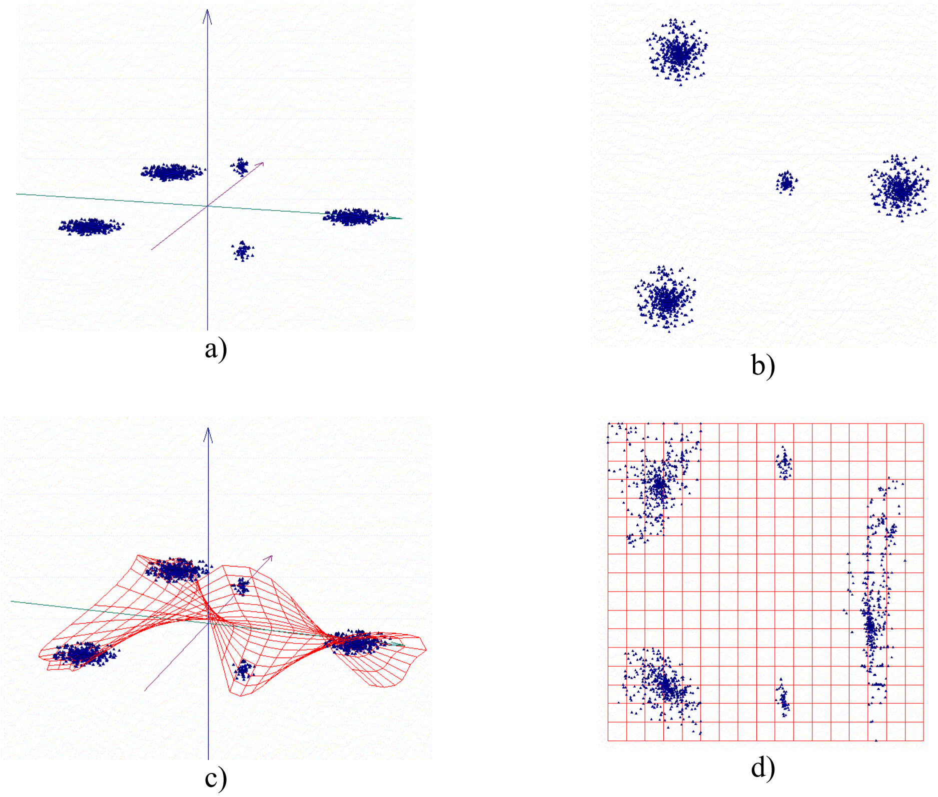


Fig. 16: Comparing linear and non-linear mapping in case of five-cluster distribution

a) five-cluster point distribution
b) projection onto the principal plane
c) constructed elastic map
d) projection on the map

In the work of Shumsky, 1998 traditional Kohonen's self-organizing maps and Hintons diagrams were used to visualize the same table but containig information for 1997 year. For data space coordinates it was suggested to use the relations of some independent features.

We enlarged the dimension of data space up to five and as a result obtained the following set of independent indicators:

N	Indicator	Description
1	LG_VO1998	Logarithm of gross production output in 1998
2	RATE	Gross production output in 1998 / Gross production output in 1997
3	PROFIT_BAL	Balance profit / Gross production output in 1998
4	PROFIT_TAX	Company profit after taxation / Gross production output in 1998
5	PRODUCTIV	Profit after taxation / Number of workers

The resulting dataset contains two hundred records and five fields. Parts of the records contain incomplete information (there are gaps in some cells).

The data is normalized with the formula

$$\tilde{x}_i = th\left(\frac{x_i - M}{\sqrt{D}}\right),$$

where $\tilde{x}_i, x_i$, M, D are the new and old value of the coordinate, mean value and dispersion respectively.

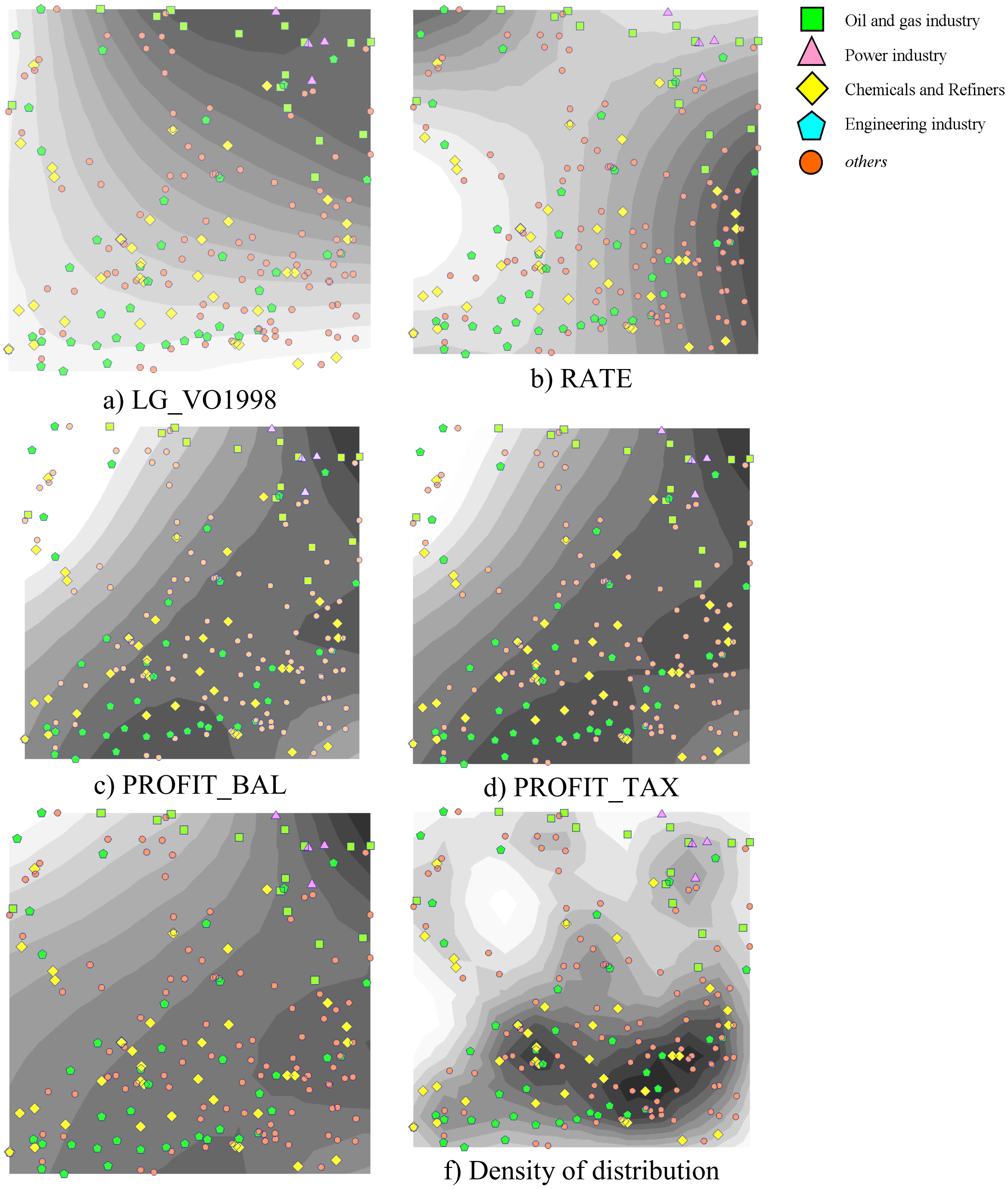
The map visualizing the data was constructed according to the algorithm of elastic map described above. The method of "annealing" was used to find the local minimum of the map functional. Parameters μ_0 and λ_0 were changing slowly (so after each change the map could reach the closest local minimum) from $\mu_0 = 5000$, $\lambda_0 = 5000$ to $\mu_0 = 0.01$, $\lambda_0 = 0.01$.

After constructing the elastic map the data points were projected from the multidimensional data space onto the map with the use of algorithm of piecewise linear projecting in the closest point of the map.

As an illustration of analysis of economical data coordinate below we give colorings of the map. In addition density of data points distribution is showed. The data points have different forms corresponding to the industry the company belongs. It helps to get insight in economical situation in Russia.

It is evident from the figures that companies with high gross output are not the same as companies with high growth rate. The largest companies belong to oil-gas and power industry. The companies of oil-gas industry are separated for two subclasses with considerably different profitability.

Coordinate colorings of PROFIT_BAL, PROFIT_TAX and PRODUCTIV are similar, this points out to the correlation of the last three indicators. At the same time distinctions in colorings allow to distinguish the companies which drop out of the correlation dependence.



a) LG_VO1998

b) RATE

c) PROFIT_BAL

d) PROFIT_TAX

e) PRODUCTIV

f) Density of distribution

Fig. 17: Coordinate and density colorings of the map of economical indicators

Political forecast

In this section we visualize the table of situations preceding moment of president election in USA. Features of every situation are binary coded answers for 12 simple questions:

1. Has the presidential party (P-party) been in power for more than one term? (MORE1)
2. Did the P-party receive more than 50% of the popular vote in the last election? (MORE50)
3. Was there significant activity of a third party during the election year? (THIRD)
4. Was there serious competition in the P-party primaries? (CONC)
5. Was the P-party candidate the president at the time of the election? (PREZ)
6. Was there a depression or recession in the election year? (DEPR)
7. Was there an average annual growth in the gross national product of more than 2.1% in the last term? (VAL2.1)
8. Did the P-party president make any substantial political changes during his term? (CHANG)
9. Did significant social tension exist during the term of the P-party? (WAVE)
10. Was the P-party administration guilty of any serious mistakes or scandals? (MIST)
11. Was the P-party candidate a national hero? (R_HERO)
12. Was the O-party candidate a national hero? (O_HERO)

Resulting table contains situations before 33 elections:

YEAR	MORE1	MORE50	THIRD	CONC	PREZ	DEPR	VAL2_1	CHANG	WAVE	MIST	RHERO	OHERO	RES
1860	1	0	1	1	0	0	1	0	1	0	0	0	2
1864	0	0	0	0	1	0	0	1	1	0	0	0	1
1868	1	1	1	0	0	0	1	1	1	0	1	0	1
1872	1	1	0	0	1	0	1	0	0	0	1	0	1
1876	1	1	0	1	0	1	0	0	0	1	0	0	2
1880	1	0	0	1	0	0	1	1	0	0	0	0	1
1884	1	0	0	1	0	0	1	0	1	0	1	0	2
1888	0	0	0	0	1	0	0	0	0	0	0	0	1
1892	0	0	1	0	1	0	0	1	1	0	0	1	2
1896	0	0	0	1	0	1	0	1	1	0	1	0	2
1900	0	1	0	0	1	0	1	0	0	0	0	1	1
1904	1	1	0	0	1	0	0	0	0	0	1	0	1
1908	1	1	0	0	0	0	0	1	0	0	0	1	1
1912	1	1	1	1	1	0	1	0	0	0	0	0	2
1916	0	0	0	0	1	0	0	1	0	0	0	0	1
1920	1	0	0	1	0	0	0	1	1	0	0	0	2
1924	0	1	1	0	1	0	1	1	0	1	0	0	1
1928	1	1	0	0	0	0	1	0	0	0	0	0	1
1932	1	1	0	0	1	1	0	0	1	0	0	1	2
1936	0	1	0	0	1	1	1	1	0	0	1	0	1
1940	1	1	0	0	1	1	1	1	0	0	1	0	1
1944	1	1	0	0	1	0	1	1	0	0	1	0	1
1948	1	1	1	0	1	0	0	1	0	0	0	0	1
1952	1	0	0	1	0	0	1	0	0	1	0	1	2
1956	0	1	0	0	1	0	0	0	0	0	1	0	1

YEAR	MORE1	MORE50	THIRD	CONC	PREZ	DEPR	VAL2_1	CHANG	WAVE	MIST	RHERO	OHERO	RES
1960	1	1	0	0	0	1	0	0	0	0	0	1	2
1964	0	0	0	0	1	0	1	0	0	0	0	0	1
1968	1	1	1	1	0	0	1	1	1	0	0	0	2
1972	0	0	0	0	1	0	1	1	1	0	0	0	1
1976	1	1	0	1	1	0	0	0	0	1	0	0	2
1980	0	0	1	1	1	1	0	0	0	1	0	1	2
1992	0	1	1	1	1	1	0	1	1	1	0	0	2
2000	1	1	1	0	0	0	1	1	0	1	0	0	2

The last field RES contains information about which party has won (1 – ruling party, 2 – opposition party) and gives classification of the elections for 2 classes.

All data points from the collection representing the table actually lie on the vertexes of the unit hypercube. It turns out that using sphere elastic net is a method more accurate for approximating such point distribution as compared to the rectangular grid. Practice showed that with equal number of nodes and elasticity coefficients value of residual sum of squares of distance to the map is considerably smaller in this case. It is quite evident thing (it is impossible to cover cube by rectangular piece of plane without folds, but sphere easily pass through every its vertex without deformation).

So for the approximation a spherical two-dimensional grid was used. To present the sphere on the plane we applied stereographic projection.

In fig.18,a the resulting grid is shown in the three-dimensional subspace spanned by the first three principal components. It is apparent that the grid approximates the points rather tightly. On the fig.18,b stereographic projection is shown. It turned out that almost all winners of the ruling party "live" on one hemisphere, the opposition winners "live" on other.

In fig.18,c-f several relevant for classification coordinate colorings are shown. There are black and white colorings because the features are binary. It is apparent that features CONC and PREZ are useful for making forecast.

We apply the method of linear regression to find the explicit form of dependence of field RES on all other fields. Values of resulting linear function are shown in fig.18,g. Black and white color corresponds to the areas of reliable prognosis, gray tint correspond to the areas of uncertainty.

Finally we applied methods of visualization to the "transposed" table. In this table features become objects and vice versa. So close in a new space objects correspond to directly correlated features, distant objects correspond to inversely correlated ones. RES feature is also included in this space. Here one can see that CONC feature is mostly correlated with RES, PREZ feature is inversely correlated, THIRD, MIST, O_HERO, DEPR, CHANGES, WAVE features group together.

Visualization of Triplet Distribution in a Window of DNA

In this section we'll show how the methods of visualization can be applied in the problem of gene identification in DNA. An overview of application of statistical methods in this area are made in Fickett, 1996.

DNA is a long word that consist of four letters. Some subwords of the DNA codes biological information, necessary for maintaining cell life cycle, and they are called *genes*. Areas between genes is *junk*. The problem is to distinguish genes and junk in the whole DNA word.

In the overview made by J.-M. Claverie (1997) three general problems of current computational gene identification methods were underlined: A) the most of the methods detect only protein coding exons; B) most of the methods work with a piece of sequence containing only one gene; C) most of the programs use methods of pattern recognition with learning with teacher - they need a training set for tuning their parameters.

In the companion paper (Gorban,Zinovyev,Popova 2001) method for identification of distinguished coding phase was proposed that uses principles of unsupervised learning. The main idea of the method is following.

The information that defines the order of aminoacids in protein are coded in DNA by codons - triplets of nucleotides. If we take an arbitrary window of coding sequence (without introns) and divide it into successive non-overlapping triplets, starting from the first base pair in window, then this decomposition and arrangement of the real codons may not be *in phase*. We can divide the window into triplets in three ways, shifting every time on one base pair from the beginning. So we have three triplet distributions and one of them coincides with the real codons distribution. So the coding region are characterized by the presence of distinguished phase.

Junk evidently has no such feature because inserting and deleting the base pair in junk do not change properties of DNA considerably, thus this kind of mutations is allowed in the process of evolution. But every such mutation breaks the phase, so we can expect than distributions of triplets in junk will be similar for all three phases.

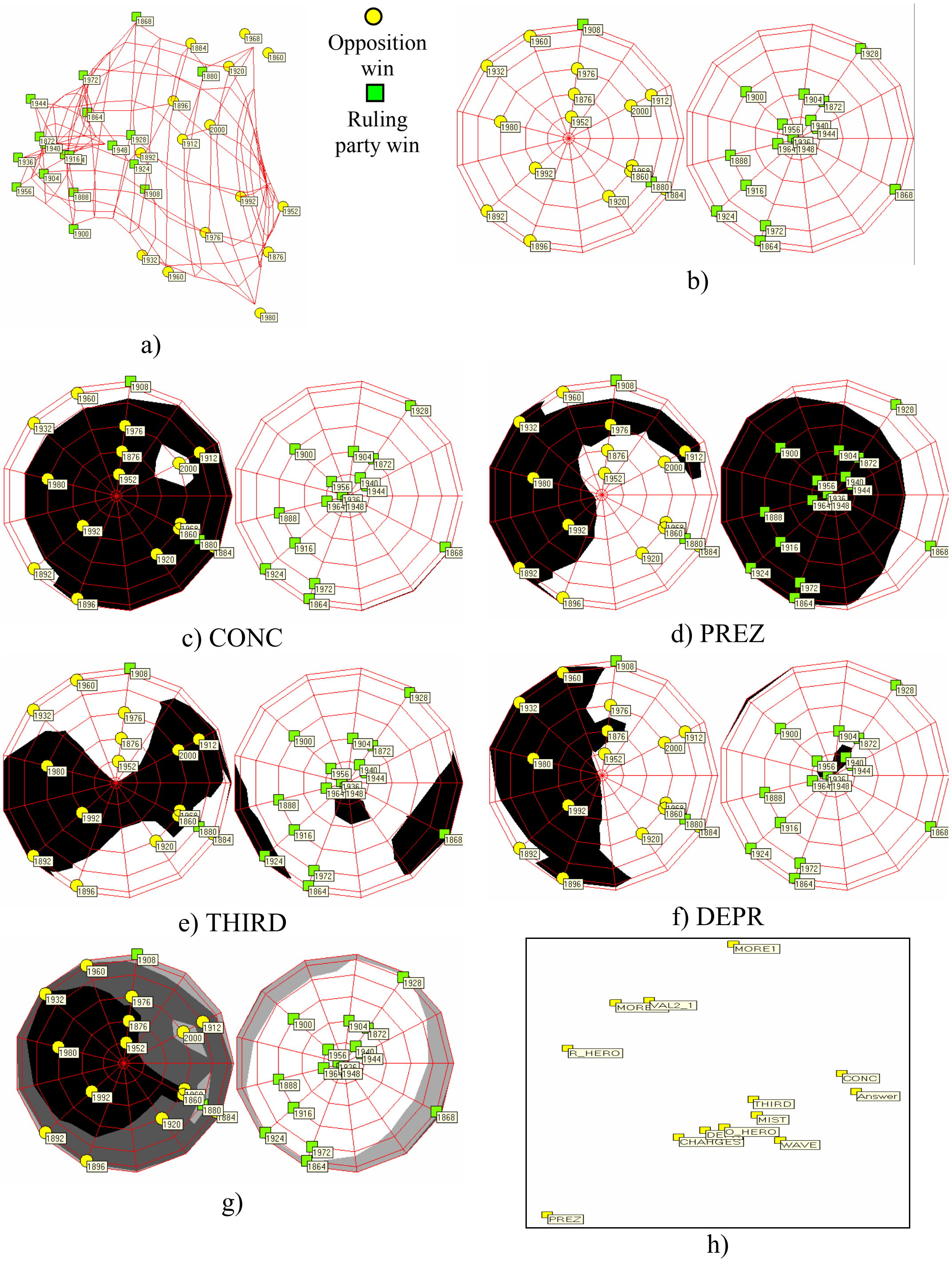
In this section we investigate distribution of frequencies of using triplets in a window sliding along the whole sequence. Visualizing of the distribution of frequencies in multidimensional space allows to formulate the procedure of gene identification without knowing anything about a new sequence.

We analyzed DNA as a single strand (without distinguishing W- and C-strand separately). The sliding window was divided into successive non-overlapping triplets, starting from the first base pair in the window (triplets in 0-phase) and frequencies of all triplets were calculated. So, every base pair is characterized by a 64-dimensional vector of frequencies. For our experiments we took every 12-th base pair of short mitochondrial DNA (Prototheca wickerhamii genome) and window length of 120 bp.

As a result we have a set of multidimensional data points x_i, i = 1… N in the space of frequencies. The data were centered and normalized on the unit standard deviation of every coordinate:

$$\tilde{x}_i^k = \frac{x_i^k - \bar{x}_i^k}{\sigma_k}$$

where x_i^k is value of the *k*-th frequency characterizing the *i*-th base pair, $\tilde{x}_i^k$ is normalized value, σ_k and $\bar{x}_k$ are standard deviation and mean value of the *k*-th coordinate.



a)

b)

c) CONC

d) PREZ

e) THIRD

f) DEPR

g)

h)

Fig. 18: Political forecast

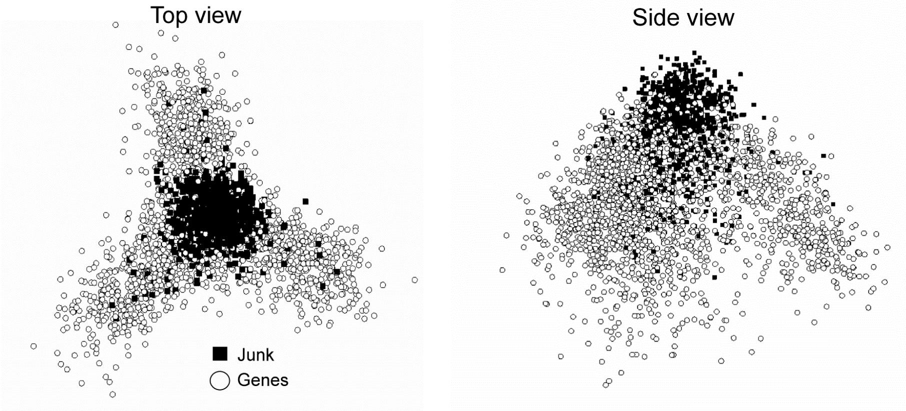
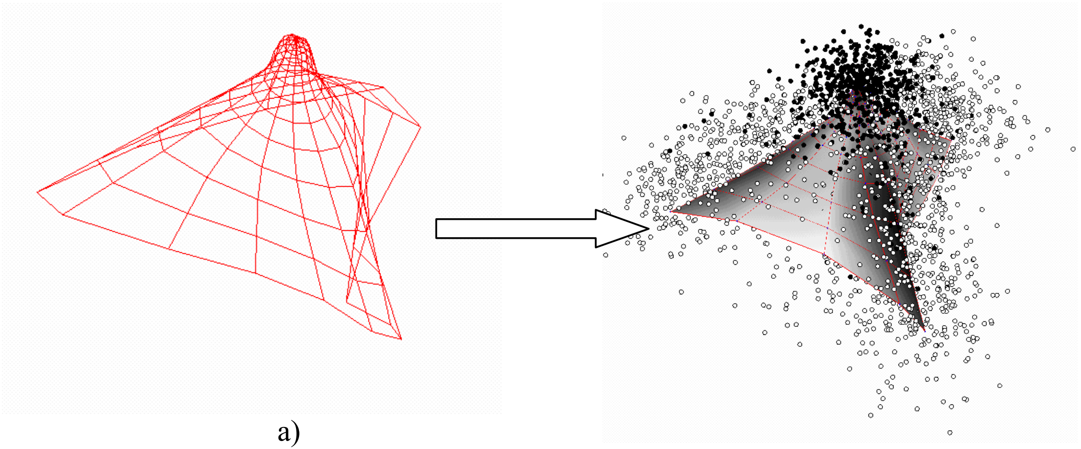


Fig. 19: Visualization of triplet frequencies in the space of the first three principal vectors



b)

Fig. 20: Elastic net (a) and elastic map (b) for visualization of triplet frequencies

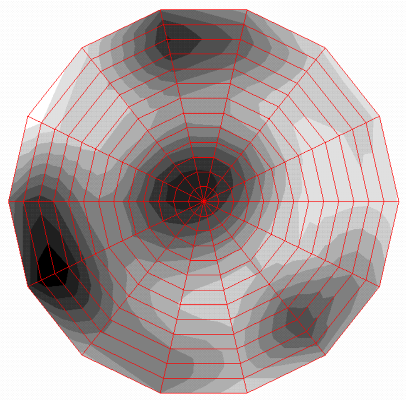


Fig. 21: Visualization of density of data points

To begin with we visualized the set of data points in the subspace of three principal vectors. The resulting pictures of data (plan view and side view) are shown in figure 19.

The pictures are quite understandable. Since the junk has no distinguished phase, it is represented by almost normal distribution situated in the center of the data point cloud. Genes with different phases forms three wings on the sides of the junk kernel. The data structure has interesting bullet-like form.

We tried to construct a map with the structure suitable for the point distribution. It has been initialized as a hemisphere with the pole in the center of the bullet. Then the elastic net with minimized energy was calculated and the final configuration of the map is shown in fig. 20.

The density of the distribution of data points is shown in fig. 21 in internal coordinates of the map. It is apparent that there are four clusters in the distribution (the central one corresponds to junk areas, and other three are genes in three different phases).

Conclusion

Methods of data visualization as a part of primary data analysis become standard in practical statistics. We think that some of them will be included in popular computer programs for statistical analysis. It is remarkable that all supposed methods require a number of computational operations proportional to the number of points analyzed.

The methods are partially realized in freeware computer program ViDaExpert 1.0, working under Windows'95'98'2000. All interested in the program may feel free to contact the authors by e-mail.

References

1. *Bernaola-Galvan P., Grosse I., Carpena P. and others.* Finding Borders between Coding and Noncoding DNA Regions by an Entropic Segmentation Method. Phys.Rev.Letters (2000), V.85, N.6.
2. *Claverie J.-M.* Computational methods for the identication of genes in vertebrate genomic sequences. Human Molec. Genetics 6. pp. 1735-1744 (1997).
3. *Fickett J.W.* The Gene Identification Problem: An Overview For Developers. Computers Chem., 1996. Vol.20, No.1, pp.103-118.
4. *Gorban A.N., Rossiev A.A.* Method of principal curves for data with gaps. Proc. of 12th international conference "Problems of neurocybernetics". Rostov-Na-Donu, Russia, 1999. pp. 198-201.
5. *Gorban A.N., Zinovyev A.Yu., Pitenko A.A.* Visualization of data using method of elastic maps. Informatsionnie technologii, 'Mashinostrornie' Publ., Moscow, 2000. N6, pp.26-35.
6. *Gorban A.N., Zinovyev A.Yu., Popova T.G.* Statistical approaches to automated gene identification without teacher. IHES preprint. IHES/M/01/34, 2001.

7. *K.Kivimoto.* Topology Preservation in SOM. Proc. of International Conference on Neural NetWorks. Washington, DC, 1996. Vol.1. PP. 294-300.

8. *Kohonen T.* Self-Organizing Maps. Springer: Berlin – Heidelberg, 1997.

9. *Shumsky S.A., Kochkin A.N.* Self-Organizing maps of financial indicators of the 200 biggest Russian companies. Proc. of All-Russia science conference "NeuroInformatics-99". Moscow, 1999. Part 3. P. 122-127.

10. *Zinovyev A.Yu.* Visualization of Multidimensional Data. Krasnoyasrk State Technical University Press, 2000. 168 p.