TALLINNA POLÜTEHNILISE INSTITUUDI TOIMETISED
PROCEEDINGS OF TALLINN TECHNICHAL UNIVERSITY
(Data Processing, Compiler Writing, Programming)
ТРУДЫ ТАЛЛИНСКОГО ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА

ISSN 0136-3549, 0320-3409 No. 614 1986
pp. 15 – 23 UDC 681.3.06

L. Vyhandu

# Fast Methods for Data Analysis and Processing

Abstract

Some effective methods to open the structure of multidimensional data are described. The notion of standardized data base schema is introduced to provide automatic generation of application programs

This paper presents some ideas of theory and practice, which have been useful for author and his colleagues at the Department of Data Processing of Tallinn Technical University. Our concern was to build an effective data analysis and processing package, having powerful graphical data representing methods for creative analysis as well as different report generators suited to users' standard administrative needs.

To archive both goals, our methods can be described as using the main part of inner degrees of freedom of data, to quickly reach a crude solution. This solution is brought (transformed) to the required accuracy (quality).

The main tools to do it are

– fast orthogonal transformations
– theory of monotonic systems developed by our group
– standardized data base schemas with automatic program generators.

## 1. Data analysis using fast orthogonal transformations

Let us have a $N \times M$-matrix $A$, representing data for $N$ objects with $M$ variables. With computer use we need $O(N \cdot M)$ operations just to have a look at the unknown data. Therefore any method of analysis has to be slower than this lower bound.

If data have been measured at least on interval scale, different least squares methods can be used to open the structure of $M$-dimensional objects. There are such well-known methods as Principal Component Analysis (PCA), Factor Analysis (FA), Clustering (C), Pattern Recognition (PR), Multidimensional Scaling (MDS). Those techniques are nowadays standard and any data analysis package has them.

But aren't there methods to give a general view of data in mach shorter time? We want to get results, which can compete with PCA, FA, C, PR and MDS. Yes, there are! They use well-known orthogonal transformations [1] with some additional handling [2].

To reach a maximum speed for $M$-dimensional data structure representation, first we use fast Haar transformation and then rotations of Jacoby type. Fast Haar transformation take only $O(N \cdot M)$ operations. It is recursively defined as follows:

$$D_1 = \| 1 \|,\ D_2 = \begin{Vmatrix} 1 & 1 \\ 1 & -1 \end{Vmatrix},\ D_{2^{k+1}} = \begin{Vmatrix} 2^{k/2} \cdot D_{2^k} & I_{2^k} \\ -2^{k/2} \cdot D_{2^k} & I_{2^k} \end{Vmatrix},$$

where $I_{2^k}$ is a Haar matrix of order $2^k$.

Through Haar transformation $B = A \cdot D$ the columns of $B$ are mach more orthogonal than those of $A$. To make them still more orthogonal we use additional Jacoby type rotations of columns $B$. Taking two columns $p$ and $r$ of $B$ such that $b_p > b_r$ ($b_p$ and $b_r$ are sums of squares of elements of columns $p$ and $r$), we can make two columns orthogonal using rotation angle

$$tan\, 2\varphi = 2 \cdot \sum b_{i\,p} \cdot b_{i\,r} \Big/ \left( \sum b_{i\,p}^2 - \sum b_{i\,r}^2 \right).$$

But we do not rotate the vectors $p$ and $r$ automatically. Namely, the sum of squares for vector $p$ will grow by the amount

$$\sum \left(b_{i\,p}^2 - b_{i\,r}^2\right) \cdot \sin^2 \varphi \Big/ \left(1 - 2 \cdot \sin^2 \varphi\right).$$

If the change is too small, we do not rotate at all. Moreover, in view of the precision of graphical representation little work is needed to obtain the first eigenvectors with adequate accuracy.

Another way is to use Thurstone's diagonal method [3] directly on $B$-matrix (not on correlation matrix) to represent the objects in a low-dimensional space.

We have also developed a very efficient method for multidimensional scaling without using gradient methods. Taking quadratic splines and optimizing coordinate-wise, we have achieved excellent results [4].

## 2. Niminal data ordination with orthogonal transformations

Let us now have a $N \times M$ data table $A$ with nominal data. We define a frequency transformation for $A$ as follows. For every variable we take its histogram and change every value $a_{i\,j} = h$ to its frequency $f_{h\,j}$ in the histogram. The row sums describe the conformity of objects in the data system.

The new matrix $Z$ is called the frequency matrix of a data matrix. If the number of categories for variables differs, we have to multiply frequencies by the number of categories $a_{i\,j} \to l_j \cdot z_{i\,j}$. We get an equalization of frequencies for all columns of $Z$ (In practice we keep the original data naturally unchanged and use histograms of all variables directly in computations).

Using either Hadamard [1] or Haar transformations and the strategy of section 1, we get one-, two- or more- dimensional ordinations for nominal data. The importance of every coordinate is measured analogously to the importance of principal components.

Another way to get interesting results is to use the scale of influence.

We define a measure of variation for every object as a sum

$$S_i = \sum_{j=1}^{m} z_{ij}^2 .$$

The larger the sum $S_i$ the more conform to group behavior is object i. Further we will define a measure of variation for the whole system as

$$S = \sum_{i=1}^{N} S_i .$$

Through a change in the sum of squares $S$, when the object $i$ is eliminated from the system, we define the influence of object $i$ on the set of objects. It is easy to find that the influence of object $i$ can be calculated as $\pi_i = \sum_{j=1}^{M} \pi_{ij}$, where

$$\pi_{ij} = 2 \cdot z_{ij}^2 - 3 \cdot z_{ij} + 1.$$

The set of numbers $\pi(i)$, $i = 1,2,...,N$ is called a scale of influence.

It is easy to see that taking a series of transformations

$$A \to Z \to \pi \to \pi H$$

we can use orthogonal transformations to open the structure of multidimensional nominal data.

---

[1] Hadamard matrix of order $m$ is a $(m \times m)$ matrix $H$ with $+1$ and $-1$ elements, such that $HH^T = mI_m$, $I_m$ – a unit matrix of order $m$. This equality is equivalent to the statement, that each pair of rows in $H$ are orthogonal. The framework of Hadamard matrices construction may by found in Hall, M, JR., Combinatorial Theory, *Blaisdell Publishing Compnay, Waltham,* Toronto, London, 1967. (notice added by J. Mullat – JM).

### 3. Monotonic systems in data clustering

Classical clustering methods are fairly slow and some difficulties occur in interpretation of clustering results. For the last twelve years our team has successfully used monotonic systems theory [2] for multidimensional data structuring [5]. Here are some general ideas of this method.

Let us suppose that there is a system $W$ with a finite number of elements. Each element has a numerical measure of its weights (influence) in the system. Further let us suppose that for every element $\alpha \in W$ there is a feasible discrete operation, which changes as well the weight of $\alpha$ and the weights of any other element $\beta$ of the system. If the elements in $W$ are independent, then it is natural to suppose that a change in the weights of $\alpha$ does not change the value of another element $\beta$.

To use the method of monotonic system we have to meet three conditions.

1. There has to be a function $\pi$, which gives a measure (weight) $\pi(w)$ of influence for every element $w$ of the monotonic system $W$.
2. There have to be rules $f$ to re-compute the influences of the elements of the system in case there is a change in the weight of one element. [3]
3. The rules for influence re-computing have to be commutative. [4]

---

[2] The first publication on monotonic systems – MS theory appeared in Tallinn Technical University Transactions, 1971, see Mullat, J.E., "On the principle for some set functions", Seria A, nr. 313, pp.37-44 (in Russian), a version of this article, translated to English, is available for download from http://www.datalaundering.com/download/modular.pdf. Yet another, an extended publication, appeared in Automat. & Telemekh. (in Russian), 1975,1976. Here the MS theory goes beyond its primary idea in finding some saturated subsets of vertices (or edges) in graphs (saturated more that any other subset does) to a general theory of weights but on subsystems subject to specific monotonic property, whereby it inherits its name. A downloadable version, translated by Plenum Publishing Corporation, J.E. Mullat, "Extremal Subsystems of Monotonic Systems, I,II,III," *Automation and Remote Control, 1976, 37*, 758-766, *37*, 1286-1294; 1977, *38*. 89-96, is available at http://www.datalaundering.com/mono/extremal.htm (notice - JM).

[3] A seminal paper reg. influences as a constructive method for Monotonic Systems design may be found in http://www.datalaundering.com/download/herring.pdf , Estonian Contributions to the International Biological Program, VI, Tartu, 1974, pp.42-47. The idea of influences turns out once again in the connection of average hits recalculation along the Makov chain, Tallinn Technical University Transactions, 1979, nr. 464, pp. 71-84, http://www.datalaundering.com/download/markov.pdf , (notice - JM).

[4] In other words – path independent, (notice – JM )

These conditions leave a lot of freedom to the researcher to choose the influence functions and rules of influence change in the system. The only constraints we have to keep in mind is that the functions $f$ and $\pi$ have to be compatible in the sense that after eliminating all elements $w$ of the system $W$ the final weights of $w \in W$ must be equal to zero.

We study all $2^{|W|}$ subsets of the set $W$. Let $\alpha \in H \subseteq W$ and $\pi^{+}H(\alpha)$ or $\pi^{-}H(\alpha)$ be the value of function $\pi$ on the element $\alpha$. We define a kernel $H^{\ominus}$ (or $H^{\oplus}$) of a system $W$ as a subset of $W$ on which there is global maximum of function $F$ of subsets $H$

$$F_{-}(H) = \min_{\alpha \in H} \pi^{-}H(\alpha)$$

or global minimum of function $F_{+}(H) = \max_{\alpha \in H} \pi^{+}H(\alpha)$.

The main theorem guaranties finding of so-called determining sequence [5], which defines exactly the extremal subset of $W$.

We will demonstrate how to use this theory on data matrices.

Let us have a $N \times M$ nominal data matrix $A$. If we take for the influence function for a data element $a_{ij}$ as $\pi_{ij} = 2 \cdot z_{ij}^2 + 3 \cdot z_{ij} + 1$, then we can define different monotonic systems on our data matrix: [6]

– objects (rows of the data matrix)

---

[5] Following monotonic systems theory so-called *defining sequence* of $W$ elements "hooked" into $p$-intervals and only one such sequence $\langle \alpha_1, \alpha_2, ..., \alpha_{|W|} \rangle$ of all $W$ elements, due to MS specific monotonic property, guaranties that the last interval $|\alpha_p, \alpha_{p+1}, ..., \alpha_{|W|}\rangle$ to the right is the best subset $H^{\ominus}$ (or $H^{\oplus}$) representing the kernel (notice – JM),

[6] One can see that the $\pi_{ij}$ values comply with the MS specific monotonic property that the recomputed values $\pi'_{ij} \leq \pi_{ij}$ after eliminating a row, column or elements from our data matrix (notice – JM).

– objects and variables (rows and columns of the data matrix)

– elements of the data matrix.

Changes in the algorithm dependent on different monotonic systems are trivial.

For simplicity we describe here very briefly but without any programming shortcuts only the first case (object clustering) using plus-influence. [7]

A1. Find the sum $P(i) = \sum_{j=1}^{M} \pi_{ij}$ .

A2. Find $R = \max_{i} P(i)$ with index $k$.

A3. Copy object $k$ as a new object into the system.

A4. Label object $k$ as taken and calculate new influence $P(i)$.

A5. Find $R' = \max_{i(i \neq k)} P(i)$ with index $k'$.

A6. If $R' \geq R$ then go to A3.

A7. All the objects from step A3 belong to the kernel.

A8. If there are more objects eliminate the first kernel and go to A2.

Our practice has shown that for interpretation it is best to use both objects and variables as elements of the monotonic system.

If the data are real numbers, we shall use an influence function for data element

$$g(a_{ij}) = a_{ij} + R_i + C_j ,$$

where $R_i$ is the sum of $i$-th row and $C_j$ – of the $j$-th column.

[7] Plus-influence in the vocabulary of monotonic system theory is a $\oplus$-action; here it is an object coping into the data table (notice – JM).

For $i$-th row we have an influence function $G(i)=\sum_{j=1}^{M} g(a_{ij})$ and for $j$-th column $G(j)=\sum_{i=1}^{N} g(a_{ij})$. For a multiplicative case one can take as an influence function

$$g'(a_{ij}) = a_{ij} \cdot (R_i - a_{ij})) \cdot (C_j - a_{ij}).$$ [8]

## 4. Effective data processing system building

To use data base systems directly is not enough. The software build-up for a given client must be evolutionary. In practice a typical database will stabilize after initial booting in 2-4 years.

To speed up the design and to shorten the tuning-in process, we have developed special technologies. They are called principles of "lazy programming" and "view of the innocent bystander".

Using the first principle we practically never solve a problem in a way our client sees it. We generalize it into some class of tasks and try to use powerful report generators [6], list processors, fast logic queries [6], compiler writing system ELMA [7] as a grammatical formalism and tool for programming [8]. The so-called standardized data base schemas have proved especially useful. We have found that 3 tunable schema classes help to bring a client directly into the data processing system creation. The first schema is very simple (Fig. 1a) and has only one main record type.

The records are broken into subsets by upper structure to speed up the processing. Below there are all kinds of versions for one CASE.

The second schema (Fig. 1b) is more interesting. Here we have two tunable schemas $A$ and $B$, which some $N:M$ relations interleave.

[8] Weights $G(i)$ constitute a monotonic system on data matrix rows, weights $G(j)$ – on data matrix columns, but weights $g(a_{ij})$ or $g'(a_{ij})$ on data matrix elements (notice –JM),

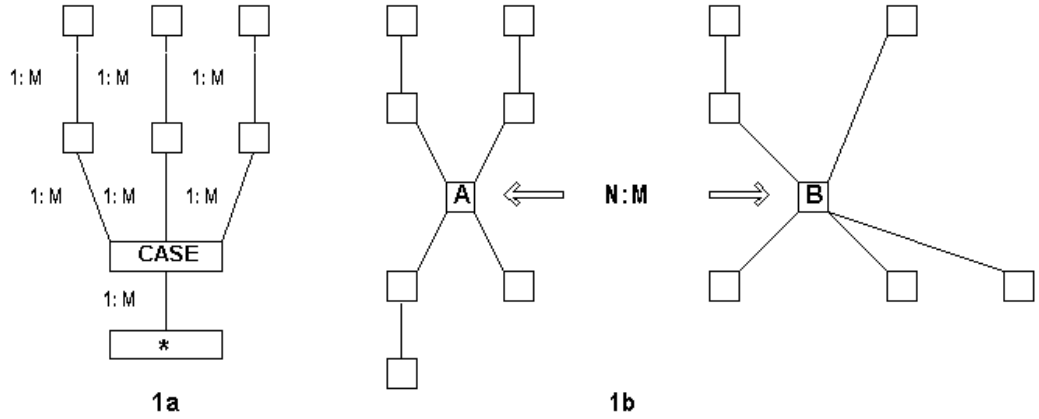


Figure 1

It is easy to go through and add some more easily tunable schemas to $A$ and $B$. For those standardized schemas we have built up specific languages, to describe data and the desired results with nonprocedural languages. Our main result along this line is:

Standardized data base scheme + specification,
languages = automatic program generation.

We get our first implementation draft usually very quickly running. After that the "appetite" of our client is likely to grow and the tuning starts to make the system more effective. What is really important – the client is able to use the system from the very beginning. That is psychologically important. The client feels that s/he her/himself was creating the system and the frustration are usually minimal.

We have developed some fairly large permanent data systems with our standard technology. For example, cancer registers in Estonia and Lithuania are built using this technology. The Estonian Ministry for health applies the technology for a statistical system [8].

**References**

1. Ahmad N., Rao K. R., Ortogonal transforms for digital signal processing. *Berlin, Heidelberg, New York*, Springer, 1975.
2. Vyhandu L., "Some problems of data analysis theory," *Trans. of Tallinn Tech. Univ.*, 1974, No. 366, pp. 3-15.

3. Thurstone L. L., Multiple factor analysis, Chicago, *Univercity of Chicago Press*, 1947.
4. Vyhandu L., et al., "Nonlinear transformations of a set of hyperspace points onto a plane," *Programs for direct synthesis of models. III, Kiev*, Institute of Cybernetics, 1975 (in russian).
5. Mullat J., Vyhandu L., "Monotonic systems in scene analysis," Symposium, *Mathematical Processing of Cartographic Data*, Tallinn, 1979, pp. 63-66. http://www.datalaundering.com/download/cardraf.pdf
6. Vyhandu L. et al., "A system to manage and process discrete information," *Control Systems and Machines*, 1981, No.1, pp. 99-102 (in Russian).
7. Vooglaid A. et al., "Input languages of ELMA system," *Trans. of Tallinn Tech. Univ.*, 1982, No. 524, pp. 79-96 (in Russian).
8. Vyhandu L. et al., "Technology of building problem-oriented data processing systems," *Trans. of Tallinn Tech. Univ.*, 1983, No. 554, pp.13-19 (in Russian).