CROATICA CHEMICA ACTA CCACAA **75** (2) 503–515 (2002)

ISSN-0011-1643
CCA-2813 *Original Scientific Paper*

2D Mapping of Large Quantities of Multi-variate Data

Jure Zupan

National Institute of Chemistry, Hajdrihova 19, 1000 Ljubljana, Slovenia
(E-mail: jure.zupan@ki.si)

Received May 2, 2001; revised September 14, 2001; accepted September 17, 2001

A new method for »intelligent« or »content dependent« retrieval of objects from among a large quantities of multi-variate data is devised and explained. The method is based on the combination of two different approaches. One is the multi-branching decision tree and the second is Kohonen neural network. The new method allows a retrieval of similar or identical objects from a number of N objects (N being in the order of 10^6 and more) in a number of comparisons proportional to $\log_9 N$. The method was developed in the connection with the question »how to map millions of multi-dimensional objects like spectra, structures, time-series of process variables, multi-component analyses of food or pharmaceutical products, *etc*.?«. In order to show how the proposed method works, a small example of 572 objects (8-component analyses of various olive oils) is described.

Key words: artificial neural networks, Kohonen learning, binary decision trees, clustering, large databases, chemical analysis, algorithms.

DEDICATION

This paper is dedicated to our friend, scientist, co-worker, teacher, and, first of all, to humanist and philosopher – Prof. Milan Randić on the occasion of his 70-th birthday. The celebration of his birthday has already passed, however, his work, his friendship and his presence with all of us in Ljubljana at the National Institute of Chemistry (even when he is in the States) is among us – each day more than a day before.

INTRODUCTION

Under the term multi-variate data, various types of experimental information are considered. In chemistry, for example, multi-variate data are digitised spectra (with up to several thousand intensities per spectrum), chemical structures (3D positions of constituent atoms or intensities in »spectrum-like« structure representation), time-series of multi-variate analytical data as a result of continuos monitoring (consecutive time windows of the same series of multi-variate analytical data), results of hyphenated spectroscopies from automated analytical measurements, sequences of genes or sequences of amino acids in proteins, *etc*. During many years of scientific collaboration with Prof. Randić from time to time a problem of handling large quantities of multi-dimensional data, especially in the connection with chemical structures and their 3D representation, pops up. Handling of structures does not involve only scanning of structures, but also a retrieval of specific, or »similar« structures from a large collection of structures (output of a structure generator, for example). Additionally, 2D mapping of large quantities of complex objects associated with the effective and flexible handling of objects based on such maps seems to be very useful for extraction of hidden information, like biological or pharmaceutical activities, *etc*.

In recent years, beside the chemical structures, a number of data sets of different data types have been collected, processed, and investigated in different ways and for different purposes. In all investigations where the number of investigated objects is of order thousands and more, with the frequency of access high, and with the speed of retrieval as crucial factors, new methods for handling such data collections have to be explored.

Regardless of the method, once the order of a magnitude of the investigated items exceeds 10^6 and each of these items is represented by thousands of individual pieces of information or measurements, even the most simple task, like finding the largest or most similar item (object) in the entire set, requires special methods. The problem addressed in this paper was initiated by the pharmaceutical companies that are scanning millions of compounds according to different criteria (structural, spectral) against their pharmacological or biological activities, or perform millions of retrievals of data to find individual compounds and/or their closest matches, for labelling, comparisons and routine checking.

The most conventional methods for 2D mapping of multi-variate objects is principal component analysis (PCA).[1,2] However, once having a 2D map featuring projections of multivariate objects as points, handling the points in the map will generate problems whenever the number of objects became too large. Even such a simple task like finding a group of similar objects to the query, a trivial task for a small data collection, requires special algorithms and consumes considerable amount of computer time. Any procedure that employs sequential scanning is not an appropriate way for handling large databases. This fact implies that large databases must be pre-processed and organised in such a way that any handling is as efficient as possible. Of course, the pre-organisation of large databases are very time consuming as well. Therefore, the same economy and efficiency reasoning as for handling individual objects within the large database must be applied for the organisation (pre-processing) of the database itself.

The organisation always depends on the goal for which the database is intended to use. Different goals require different organisations. If only the identity (*i.e.*, the problem of identification and retrieval of objects identical to the query) is at stake, the fast retrievals can be managed *via* direct access files and hash-code addressing[3,4] or by database organisation in a »decision-tree« like structure.[3,5] Although hash algorithms are excellent tools for locating and retrieving the identical matches, they are almost useless in the case of retrieval of corrupted or incomplete matches. Unfortunately, in chemistry most of the handling of multi-dimensional data involves slightly modified and/or slightly corrupted objects and patterns (spectra for example). The imperfections are mainly due to the measurement errors. Neither two spectra, nor two m-component chemical analyses, nor two m-variate process vectors, *etc*. are completely identical, hence, a fast retrieval of several similar objects to the query is preferred to the retrieval of identical objects. The latter one can fail too easy due to the fact that identical items are seldom in the database. On the other hand, the »decision-tree« pre-processed database[3,5] enables the retrieval of objects similar to the query with only slightly lesser efficiency compared to hash coding, however, it does not provide the mapping ability.

One possible solution seems to be the improvement of the algorithms for the interpretation of 2D maps with multi-dimensional objects mapped as points. Mapping of m-variate objects into two-dimensional plane is associated with a significant loss of information. As pointed out above, the most used linear mapping method in chemistry is PCA. The PCA mapping is made, first, by rotating the objects in the m-dimensional measurement space of original (correlated) variables, second, by aligning the new axis into the directions of largest remaining variance, and finally, by plotting the new objects into the plane of the first two principal components. This simple and very flexible method has, unfortunately enough, two serious flaws which prevents it from more general use. The first one is that the method works best for highly correlated variables and breaks down completely for independent ones. In practice this property makes PCA useless for 2D mapping if less than 50% of variance is collected in the first two principal components. The second flaw is that PCA is only a rotation of axes and it cannot map data into a distorted non-linear co-ordinate system. Of course, the two listed flaws are not the drawbacks of the method itself, but rather the problem of the data. Nevertheless, other mapping methods should be developed.

Among the non-linear methods the most popular mapping method is Kohonen self-organising artificial neural network.[6–9] Kohonen mapping[8,9] is mathematically even simpler than the PCA. Unlike PCA, it has two flaws that are associated with the method itself and not with the data. The first flaw is the digital nature of the Kohonen map. The position of each object in the final map is namely defined by the position of the corresponding »best« or »excited« neuron in the network. This means that the objects can be placed on the map only at discrete positions of neurons and not in between. The other drawback is its extremely time consuming generation procedure. Generation of the Kohonen 2D map is based on finding the »best match« or the »best response« for each input object in an iterative manner. The search for the best response implies scanning all neurons in the network for each single step during the learning. This flaw is not serious for small networks, however, it can be the limiting factor in networks containing hundreds of thousands of neurons for mapping millions of m-dimensional objects that must be sent through the entire network several thousand times.

In order to enable mapping of a large number of m-dimensional objects and at the same time effective handling of the mapped objects, a new method that combines the »decision-tree« structure of the data base and the Kohonen mapping is suggested and explained in this work.

THE METHOD

The proposed method is based on the combination of two techniques: a) three-distance hierarchical clustering[4] and b) Kohonen mapping.[8,9] Figure 1 shows a binary decision tree composed of a root, sub-roots or nodes, branches and leaves. The multi-variate objects (structures, spectra, multi-component analyses, process-vectors, *etc.*) are leaves at the end branches of the tree. The branching points (nodes and root) are the representatives of the objects below them. The representative of any node must describe the group of objects linked together under this particular node in a sufficiently precise way so that a decision about the continuation of the traverse of the query object in the tree can be made. The most usual representation is the average of all objects below the node. The representative should have the same m-dimensional representation as the objects themselves. This requirement (not mandatory for any decision tree) assures that the nodes can be represented and treated in any procedure in the same way and in the same measurement space as the objects.

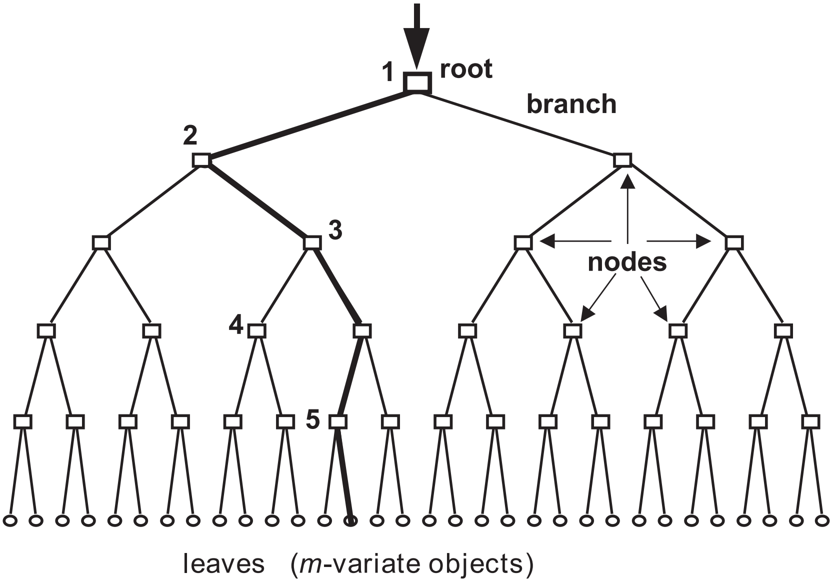

Figure 1. Perfectly balanced binary decision tree of 32 objects. Each object can be accessed in a $\log_2 32 = 5$ decisions. Each decision tree is composed of nodes (the root is the upper-most node), branches, and objects (leaves). Each node representing all objects below it is described in the same manner as the objects.

If binary decisions at each node are substituted by multi-branch decisions the access to any object becomes even faster, providing of course, that the multi-branch decision tree is balanced to a higher degree. Figure 2 shows a nine-branch decision tree. If the branching number of the tree increases from 2 to 9, the number of comparisons N_c decreases. For example, in the case of one million objects $N_c = \log_9(10^6) = 6 \log_9 10 \sim 6$.

The branching number of nine decisions per node (shown in Figure 2) is chosen on purpose. The reason for the choice is the fact that nine possible directions along one of them the query object must traverse from a node in the particular decision level to the next level, enable the organisation of nodes into groups of 3 × 3 ordered in a planar manner. Thus all nodes in each decision level k can be are arranged in a $3k \times 3k$ map. The planar $3k \times 3k$ arrangement of nodes is forming the grid for the mapping of m-variate nodes into the 2D plane. However, the planar 3 × 3 arrangement of the nodes itself bears no essential improvement unless the nine-outcome decision itself ex-

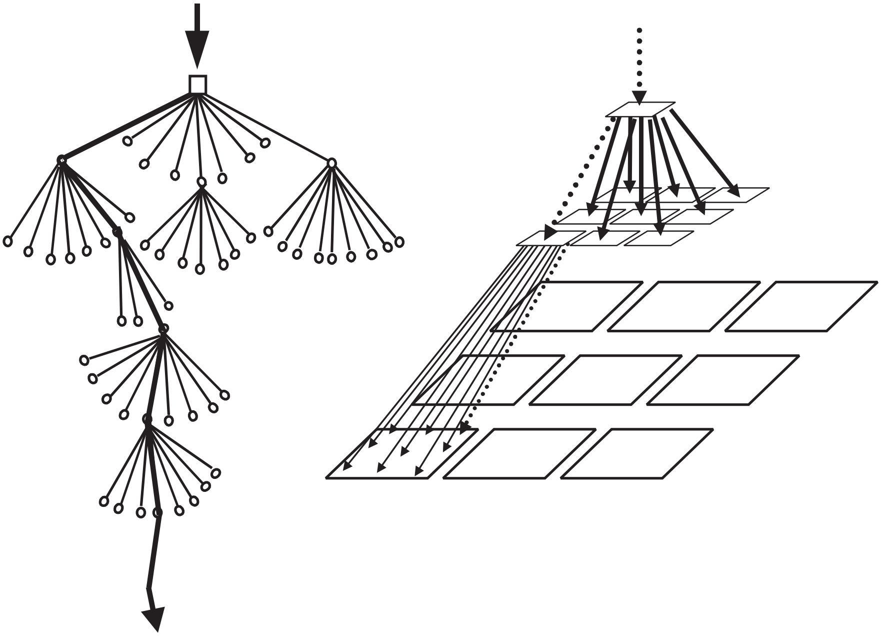

Figure 2. Nine-branch-decision tree is a »bridge« between binary (or two- branch) decision trees and decision hierarchy that can be shaped into a series of 2D maps. Nodes of the nine-branch decision structure (left) arranged in a decision plane composed of 3 × 3 »decision units« (right).

ploits the planar (*i.e.* metrical or »topological«) layout of nodes which are representatives of objects laying below them. Evidently, topological meaningful arrangement of m-variate objects can be achieved by Kohonen self-organised mapping.[5,7]

Thus, the essence of the new proposed method is the combination of a decision tree and Kohonen self-organised maps, *i.e.*, the nine-branch decision hierarchy based on $3k \times 3k$ levels of small 3×3 Kohonen networks or »decision nodes«. Each of the 3×3 Kohonen networks is acting as a nine-branch decision node. Each decision level of the 9-branch decision hierarchy is composed of 3×3 Kohonen networks of the same size as the entire Kohonen network on the level above. The root of the decision tree is thus a single 3×3 Kohonen network, labelled as KN^1 consisting of only nine neurons. The query (either for training or for the retrieval) enters the KN^1 and depending to the position of the winning neuron is directed into the 3×3 Kohonen node of the layer KN^2 below (Figure 3). Here, the term *level* (associated with levels in the decision tree) is substituted by the term *layer*, used for describing the artificial neural network architecture.

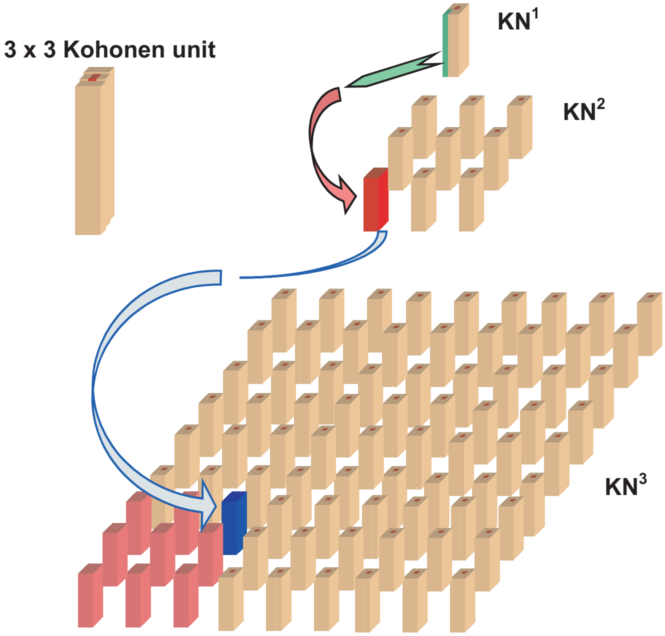

Figure 3. Small (3 × 3) Kohonen networks arranged as decision units in three-layer nine-branch decision hierarchy.

The retrieval of the query objects from such a hierarchical scheme of Kohonen networks is straightforward. The position of the most similar neuron $[i^k, j^k]$ in the k-th decision layer, KN^k, determines the position $[i_c^{k+1}, j_c^{k+1}]$ of the central neuron $W_{ic,jc}^{k+1}$ in the decision layer KN^{k+1} below. The relation between both positions, regarding the fact that the decision layer KN^{k+1} below is 3 × 3 times larger than the decision layer KN^k above, is determined by a simple formula:

$$i_c^{k+1} = 3(i^k-1) + i^k \tag{1a}$$

$$j_c^{k+1} = 3(j^k-1) + j^k \ . \tag{1b}$$

The search for the closest match to the query object X in the decision layer KN^{k+1} is then performed in the 3 × 3 neighbourhood of the neuron at the position $[i_c^{k+1}, j_c^{k+1}]$. Besides limiting the search area to a small 3 × 3 Kohonen unit, there is another advantage of hierarchical layout of small units (or nodes). There is no actual separation between these 3 × 3 Kohonen units and either the training or the search procedure can always be extended arbitrarily over the area of 3 × 3 neurons. Regardless of the extension of the procedure beyond the 3 × 3 limit of the neighbourhood both equations (1) stay the same. By extending the neighbourhood's area while keeping equations (1) for determination of the centres intact, strong overlap and interaction between all 3 × 3 Kohonen neighbouring units is achieved. Such an overlap, especially during the training, has a beneficial influence on the quality of mapping, on the more realistic distribution of objects, and consequently on the efficiency of content dependent retrievals.

The neighbourhood limits around the central neuron to which either the search for the closest match or the correction of weights is performed depends on the needs and the chosen priorities. If the efficiency and speed of the training an/or retrieval is paramount, the neighbourhood should be small (*i.e.*, the first neighbours or 3 × 3 area). On the other hand, if the separation of large clusters and better insight into inner structure of clusters is required, neighbourhoods larger than 3 × 3 (*i.e.*, 4 × 4 or 5 × 5 or even larger) should be chosen.

ALGORITHM

The actual correction of weights, the so-called training of the self-organised Kohonen maps is made using the standard Kohonen learning procedure.[6–9] Only a brief description of the algorithm is given here. The interested reader can consult a number of texts in this matter.[8,9]

Step 1. For start, a set of $\{(3 \times 3 \times m), (9 \times 9 \times m), (27 \times 27 \times m), (81 \times 81 \times m), (243 \times 243 \times m), (729 \times 729 \times m)\}$ Kohonen weight layers (from KN^1 to KN^6) has to be randomised. This six-layer nine-branch decision hierarchy of 3 × 3 Kohonen units (or decision nodes) enables efficient mapping, storing, and display of more than half a million m-dimensional objects. In any layer KN^k the neurons have m weights and are labelled as $W_{i,j}{}^k = (w_1,w_2,..w_m)_{i,j}{}^k$. The neighbourhood range $p = 1$ allows to search the neurons in the area in which 3 × 3 KNs are situated. With other words $p = 1$ means that only the first ring of neighbours around the selected neuron is considered, $p = 2$ means two closest rings of neighbours *etc*.

Step 2. The starting point is the 3 × 3 Kohonen layer KN^1 at the top of the decision hierarchy ($k = 1$) shown in Figure 3. Evidently, its central neuron has always the position $[i_c{}^1, j_c{}^1] = [2,2]$

Step 3. The »most excited« neuron (the »closest match«) in the layer KN^k excited by the object $\mathrm{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots \mathrm{x}_m)$ is obtained by finding the neuron having the smallest distance to it within the pre-specified neighbourhood range p ($d = 0, 1$ if $p = 1$):

$$[i^k,j^k] \leftarrow \min \{d(\mathrm{X},W_{i,j}{}^k)\} \text{ for all neurons } W_{i,j}{}^k \text{ in the selected neighbourhood } p. \tag{2}$$

Step 4. The weights of all neurons W_{ij}^{k} at the particular distance d from the central neuron $[i^k, j^k]$ (but still within the neighbourhood range p) are corrected according to:

$$W_{ij}^{k} = \alpha\,[1\text{–}d/(p + 1)](\mathrm{X}\ \text{-}W_{ij}^{k}) \quad d = 0,1\ldots p \tag{3}$$

where α is a learning constant. At the beginning of the training α is set to less than 1.0 (usually around 0.5). During the learning α decreases in most applications towards zero (usually towards a small value: around 0.001). For $p = 1$ the expression $[1-d/(p+1)]$ is equal to 1.0 or to 0.5 for $d = 0$ *or* $d = 1$, respectively. If larger neighbourhoods, $p > 1$, are considered, the expression $[1-d/(p+1)]$ is linearly decreasing and describes the so-called triangular diminishing of the corrective influence on the weights that is related to the distance of the particular neuron W_{ij}^{k} from the central neuron $W_{\mathrm{ic},j\mathrm{c}}^{k}$.

Step 5. Once the weights in the layer KN^{k} are corrected, the central neuron $W_{\mathrm{ic},j\mathrm{c}}^{k+1}$ in the layer KN^{k+1} is determined according to equation (1). The procedure continues at step 3 until the last layer of neurons, $\mathrm{KN}^{\mathrm{last}}$, is corrected.

Step 6. The procedure from step 3 to step 5 is repeated for all m-variate objects $\{\mathrm{X}_s\}$, $s = 1\ldots r$; r being the number of all objects.

Step 7. Steps 3 to 6 are called »one epoch« of training. Generation of 9-branch decision hierarchy is accomplished after the pre-specified number (usually several hundreds) of learning epochs, steps 3 to 6, is performed.

During each epoch of training the learning constant α and/or the maximum neighbourhood limit p can be diminished. In the case that at the start $p = 1$ was chosen (3×3 units are taken into consideration), p is set to zero ($p = 0$) exactly in the middle of the training. From then on, according to equation (3), only the central neuron is corrected ($d = p = 0$).

For the retrieval the same pre-specified number of neighbouring neurons is always considered. In most cases p is selected to be $p = 1$, although $p = 2$, 3, or more can be used for searching. The increase of the search area does not improve the results, if the same values of p are not used in the training as well.

EXAMPLE

In order to show how the proposed method works a small example of multi-component analyses of 572 olive oils from nine regions in Italy is shown. The applied set of data is thoroughly described in the literature[10,11] and is used often as a standard testing set for evaluation of new chemometric methods.[11] Each of the 572 olive oils was analysed for the content of eight fatty acids (palmitic, palmitoleic, stearic, oleic, linoleic, arachidic, linolenic, and eicosenoic). The oils originate from nine Italian regions: North Apulia (No. 1), Calabria (No. 2), South Apulia (No. 3), Sicily (No. 4), Inner Sardinia (No. 5), Coastal Sardinia (No. 6), East Liguria (No. 7), West Liguria (No. 8), and Umbria (No. 9). The data matrix thus contains 4576 concentrations of various fatty acids in various oils. The concentrations of different fatty acids vary from 60–85% of oleic acid to as low as 0.0–0.6% of ecosenoic acid. Each variable (fatty acid concentration) was therefore normalized between its maximal and minimal value among all 572 oils. Mappings with both methods were executed using the same normalized data file. PCA was calculated using the *TeachMe* program,[12] while Kohonen hierarchical mapping was implemented on the home-made program.

Figures 4 and 5 show 2D maps of 572 oils obtained by the PCA and the new proposed method. On both maps each olive oil is assigned with the

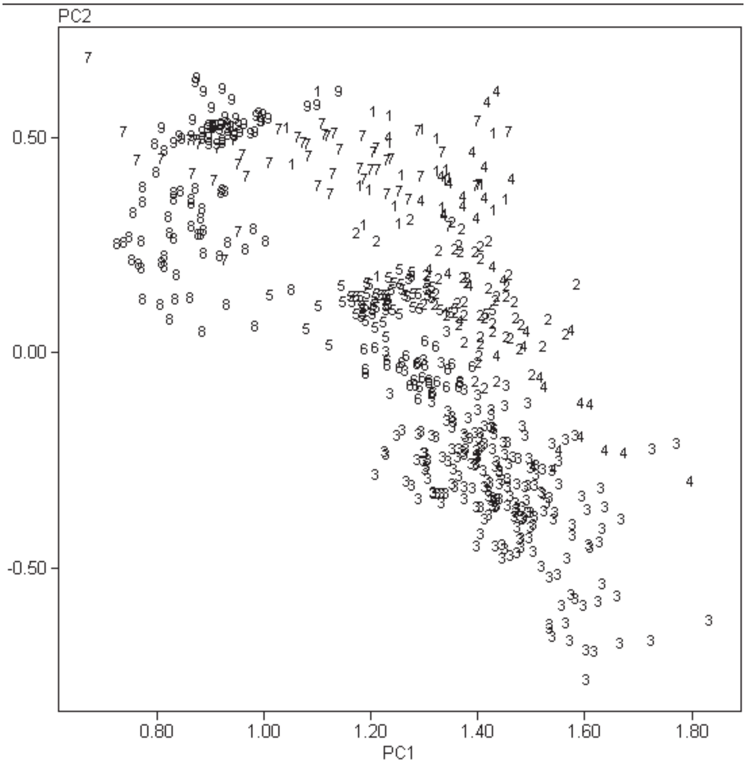

Figure 4. PCA plot of 572 olive oils' 8-variate chemical analyses. Comparison between the separation of olive oil classes in this figure with the one made by the proposed 2D Kohonen mapping (Figure 5) reveals worse separation of classes achieved by PCA. Both maps are produced from the exactly the same data file.

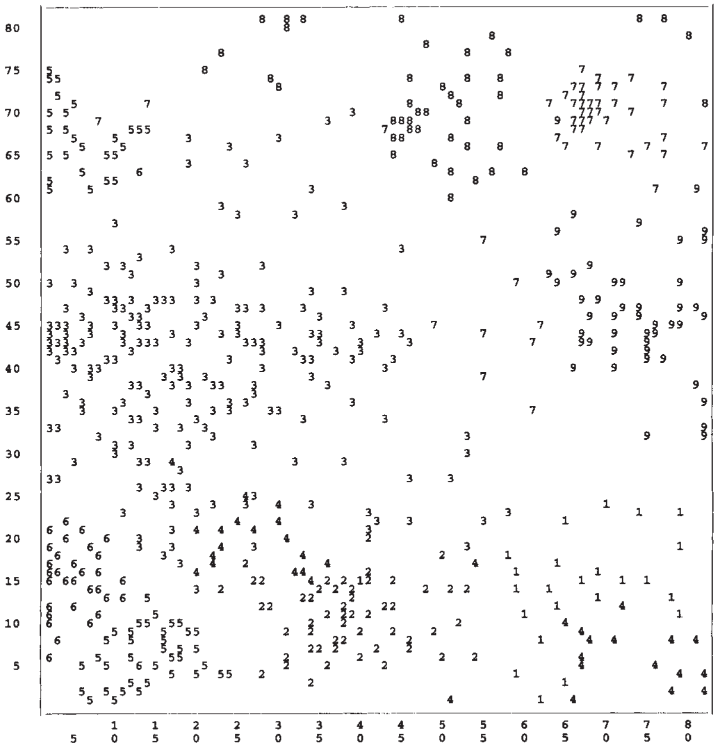

Figure 5. Display of 2D projection of 572 olive oils' 8-variate chemical analyses on the fourth Kohonen neural network layer (81 × 81 neurons) produced by the new proposed method (compare with the PCA plot of the same data shown in Figure 4). Oils belonging to one of the nine groups (labelled with numbers from 1 to 9) are much better separated than by the use of PCA.

identification number of the region in Italy where it originates from. Therefore, on each map exactly 572 objects (very few of them are completely overlapped) carrying labels from 1 (N. Apulia) to 9 (Umbria) are shown. Due to its linear nature the 2D map produced by PCA (Figure 4) shows relatively poor spread of the data across the entire plane compared to the spread of data produced by the 2D Kohonen hierarchy. The Kohonen hierarchy shown in this example is employed only up to the fourth level, *i.e.,* up to the 81 × 81 map which is sufficient for this size of a database. Employing larger maps would not increase the selectivity of the clustering. Already on this map much nicer separation of clusters compared to the PCA separation can be seen. The separation between the southern (No.1 to No. 4), Sardinean (No. 5 and No. 6) and northern (No. 7 to No. 9) olive oils is clearly seen on Kohonen mapping while PCA, although it separates southern from northern oils, leaves the Sardinean oils (No. 5 and No. 6) almost completely overlapped with the southern ones.

The real advantage of the Kohonen mapping is the retrieval of similar or »neighbouring« oils for a given query. The retrieval of all neighbours (occupants of the neighbouring cells) is almost instantaneous and available at the same moment as the object is mapped, *i.e.*, after only 36 comparisons (= 9 neighbours times 4 layers). In order to retrieve oils similar to the query using the PCA method all 572 oils have to be checked again and a list of closest matches maintained after check of each oil. In the small data bases of thousands of objects such an effort does not count much, however, when inspecting millions of complex analyses, or spectra, or chemical structures, or process vectors of fast reactions, the speed and efficiency of such searches are essential.

CONCLUSION

A method that can simulate content dependent retrieval and at the same time offers 2D map of multi-variate objects and is applicable for large databases is discussed. To train a Kohonen decision network hierarchy having in the last layer close to one million (*i.e.*, approximately 1000 × 1000) neurons compared with training the stand alone Kohonen network of the same size with the standard Kohonen learning procedure requires four orders of magnitude less computation time. Standard Kohonen learning procedure requires hundreds (if not thousands) of entries for each object from the training set to be input into the network and checking the query with each and every neuron in the network. This procedure requires to perform 10^{14} comparisons to generate a 10^6 neurons Kohonen network for 10^6 objects in a hundred epochs. The new proposed method of hierarchical ordering requires for the same task about four to five orders of magnitude lower number. This reduction is possible because each object in each epoch should be compared at each layer of neurons (there are 6 layers) only 9 times making altogether 54 comparisons instead of one million needed in the standard method.

The same is true for retrieval. In the new scheme a nine-branch decision three shortens the number of comparisons, N_c, needed for a query object to be mapped in the network containing N cells, from N to $\log_9 N$. Once the one million neurons map is completed each retrieval is almost instant being four to five orders of magnitude lower than that for a standard Kohonen retrieval. Such efficiency makes the proposed method the fastest Kohonen mapping procedure.

Acknowledgement. – The Financial support of the Ministry of Education, Science and Sport of Slovenia of this research through the scientific programme grants P-104–508 and P-104–507 is gratefully acknowledged.

REFERENCES

1. D. L. Massart, B. G. M. Vandeginste, L. M. C. Buydens, S. De Jong, P. J. Lewi, and J. Smeyers-Verbeke, *Chemometrics*: *Handbook of Chemometrics and Qualimetrics*, Part A, Elsevier, Amsterdam, 1997, pp. 519–553.
2. K. Varmuza and H. Lohninger, *Principal Component Analysis of Chemical Data*, in: J. Zupan (Ed.), *PCs for Chemists*, Elsevier, Amsterdam 1989, pp. 43–64.
3. D. E. Knuth, *The Art of Computer Programming, Sorting and Searching*, Addison-Wesley, Reading, Mass, 1975, Chapter 6.4.
4. J. Zupan, *Clustering of Large Data Sets*, Research Studies Press, (John Wiley & Sons), Chichester, 1982.
5. J. Zupan, *Algorithms for Chemists*, John Wiley & Sons, Chichester, 1989, pp. 34–43.
6. T. Kohonen, *Biol. Cybern.* **43** (1982) 59–69.
7. T. Kohonen, *Self-Organization and Associative Memory*, Third Edition, Springer-Verlag, Berlin, 1989.
8. J. Zupan and J. Gasteiger, *Neural Networks for Chemists*, VCH, Weinheim, 1993, pp. 79 –95, 167–195, and 261 –291.
9. J. Zupan, M. Novič, and I. Ruisanchez, *Chemom. Intell. Lab. Syst.* **38** (1997) 1–23.
10. (a) M. Forina and C. Armanino, *Ann. Chim.* (Rome) **72** (1982) 127–143; (b) M. Forina and E Tiscornia, ibid. 144–155.
11. J. Zupan and D. L. Massart, *Anal. Chem.* **61** (1989) 2098–2182.
12. H. Lohninger, *Teach/Me*, Data Analysis Program, Springer Verlag, Berlin, 1999. (http://www.vias.org/teachme)

SAŽETAK

2D prikazivanje velikih vrijednosti multivarijatnih podataka

Jure Zupan

Predložena je i objašnjena nova metoda za »inteligentno« ili »sadržajno ovisno« pronalaženje objekata između velike količine multivarijatnih podataka. Metoda se temelji na kombinaciji dvaju različitih pristupa. Jedan je višestruko razgranano stablo odlučivanja, a drugi je Kohonenova neuronska mreža. Nova metoda dopušta pronalaženje sličnih ili identičnih objekata iz ukupno N objekata ($N \geq 10^6$) u broj usporedaba proporcionalnih $\log_9 N$. Metoda je razvijena u vezi s pitanjem »kako prikazati milijune višestruko dimenzijskih objekata poput spektara, struktura, vremenskih nizova procesnih varijabli, multi-komponentnih analiza hrane ili farmaceutskih produkata, itd?«. U svrhu demonstriranja kako predložena metoda radi, opisan je primjer od 572 objekata (8-komponentna analiza različitih maslinovih ulja).