ASSO Project
Workpackage WP6.2 - Module SYKSOM

Generalized Kohonen maps

Visualizing large tables of interval data by using the modules SYKSOM and VMAP

A Short Tutorial for Users

Hans-Hermann Bock

Version 1: November 7, 2002

1 Interval-type data for Kohonen maps

Visualizing data in the form of illustrative diagrams is a useful tool in exploratory statistics and data mining. In the case of symbolic data, powerful methods are provided by suitable 'symbolic' adaptations of classical methods such as principal components, factor analysis, etc. which produce two-dimensional displays, e.g., in the space of the first and second 'factors' (see the software modules SPCA and SGCA).

An alternative visualization is obtained by *Kohonen maps* and implemented in the module SYKSOM. Here SOM stands for Self- Organizing Maps and remembers the fact that the underlying algorithm proceeds by sequentially adapting current results (the current 'map') to new information obtained by additional observations or samples, and insofar works in a 'self-organizing' way.

SYKSOM assumes that you have a large table of interval-type data for n items $k = 1, ..., n$ (cities, article groups, persons, families,...) and p interval-valued variables $j = 1, ..., p$. A small-sized example is given by Table 1.

$k \setminus j$	$Var\ 1$	$Var\ 2$	$Var\ 3$
1	$[8.5, 10.0]$	$[13.0, 15.2]$	$[5.0, 8.2]$
2	$[6.3,\ 9.1]$	$[14.1, 16.0]$	$[6.3, 7.2]$
3	$[7.9, 11.8]$	$[11.6, 13.5]$	$[4.9, 6.5]$
4	$[9.0, 11.0]$	$[10.9, 12.5]$	$[7.1, 8.1]$
5	$[6.3,\ 7.2]$	$[12.9, 15.0]$	$[6.2, 7.4]$
6	$[7.1,\ 7.9]$	$[11.5, 12.9]$	$[4.8, 5.7]$
7	$[7.5,\ 9.4]$	$[13.2, 15.0]$	$[6.6, 8.1]$
8	$[6.6,\ 7.8]$	$[12.4, 13.2]$	$[5.7, 7.2]$

Tab. 1: *A data table for $n = 8$ items (cities $k = 1, ..., 8$) and $p = 3$ interval-type variables (price ranges in 1000 Euro for car types $j = 1, 2, 3$)*

Let $x_{kj} = [a_{kj}, b_{kj}]$ denote the interval which forms the entry of the cell (k, j) of this table. Typically, this interval measures the range of observed values of the j-th variable for individuals from item k, e.g., in Table 1 the minimum and maximum price of a car of brand j in the city k. Considering all p variables (columns) simultaneously, the item k is then described by the row k of the table, i.e. the vector of intervals

$$x_k = \begin{pmatrix} [a_{k1}, b_{k1}] \\ \vdots \\ [a_{kp}, b_{kp}] \end{pmatrix}, \qquad \text{for example:} \qquad x_4 = \begin{pmatrix} [\ 9.0, 11.0] \\ [10.9, 12.5] \\ [\ 7.1,\ \ 8.1] \end{pmatrix} \tag{1}$$

or, equivalently, by the rectangle

$$Q_k = [a_k, b_k] \subset \mathbb{R}^p$$

in the p-dimensional space $\mathbb{R}^p$ with sides $[a_{kj}, b_{kj}]$ for $j = 1, ..., p$ where

$$a_k = \begin{pmatrix} a_{k1} \\ \vdots \\ a_{kp} \end{pmatrix} \qquad \text{and} \qquad b_k = \begin{pmatrix} b_{k1} \\ \vdots \\ b_{kp} \end{pmatrix}$$

are the „left lower vertex“ and the „right upper vertex“ of Q_k (see Fig. 1).

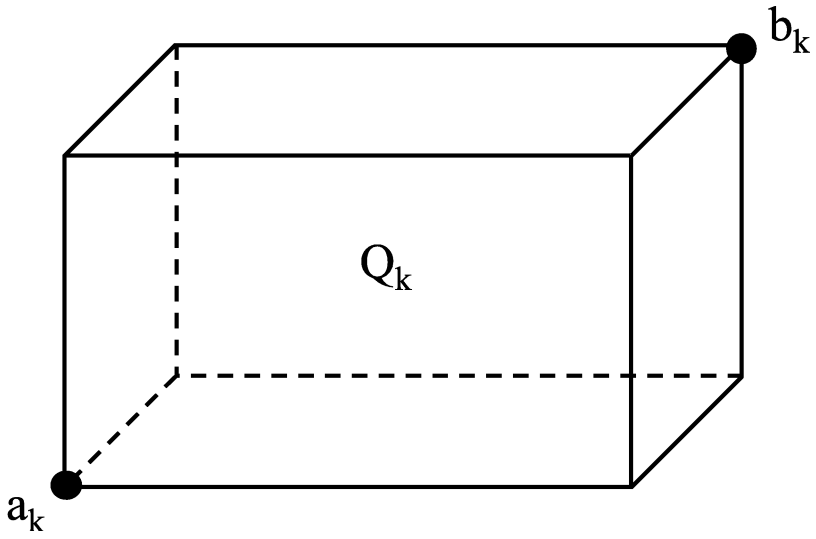

Fig. 1: *The hyperrectangle $Q_k = [a_k, b_k]$ representing item k*

Thus, our data table corresponds to a cloud of n rectangles in R^p which may be more or less clustered (see Fig. 2).

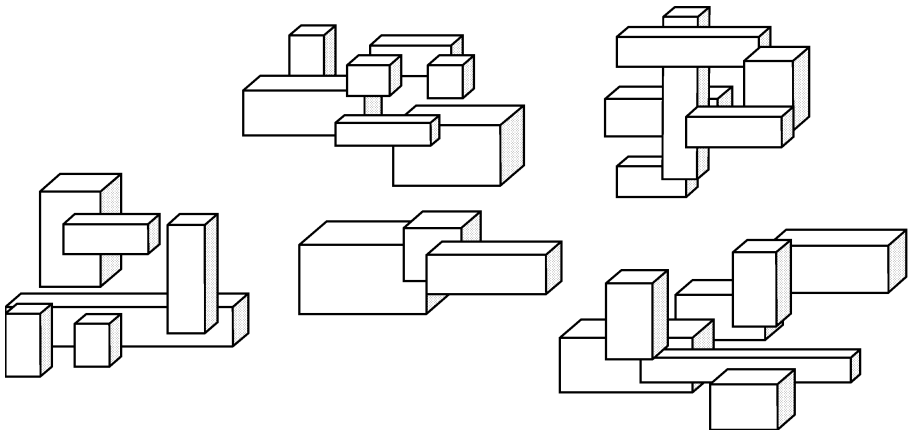

Fig. 2: *All n items (rows of the data table) represented as rectangles Q_k in the p-dimensional space, here with an obvious clustering structure*

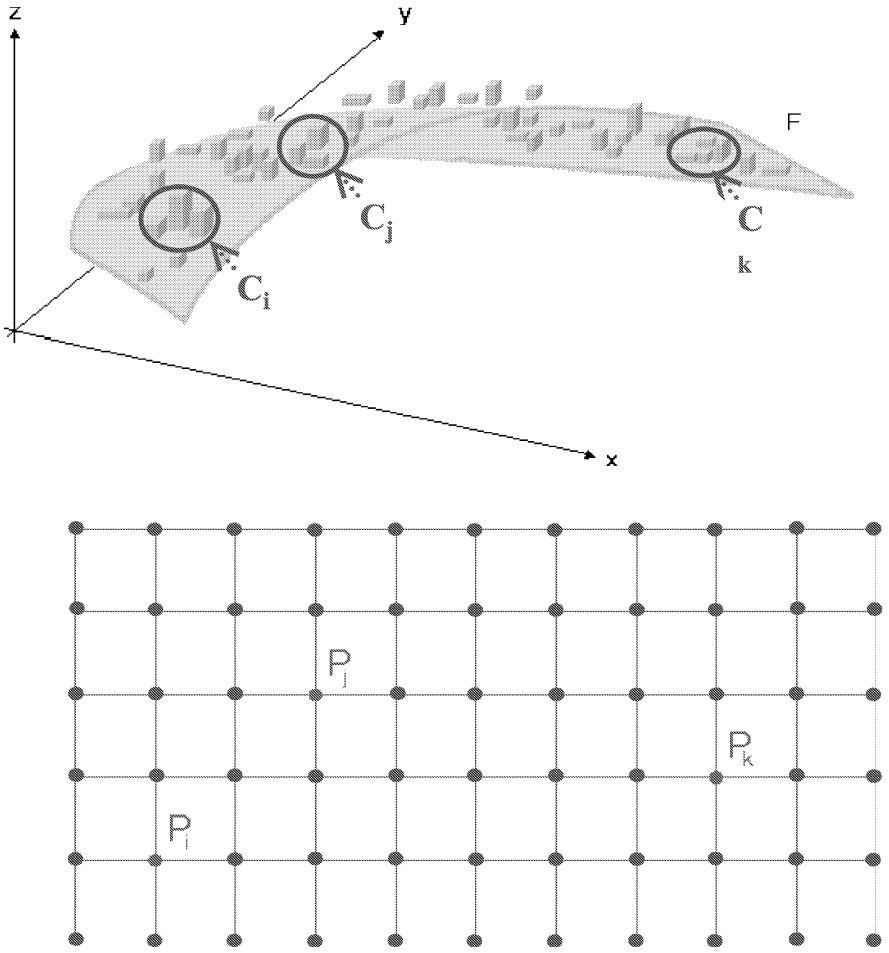

Fig. 3: *A cloud of rectangles in R^3, located near a two-dimensional manifold F, and a rectangular grid L with $a = 6$ rows, $b = 11$ columns, and $m = 6 \cdot 11 = 66$ vertices P_i*

If the number n of items is very large (as might be the case, e.g., for data bases in marketing, public surveys, meteorology), the configuration of the corresponding rectangles in $\mathbb{R}^p$ cannot be easily overlooked or analyzed, and clusters or outliers are hardly to detect in the data table itself. Even for (only) three dimensions, a configuration such as in Fig. 3 might give some problems in a direct approach.

In this situation, Kohonen maps are a useful tool in order to visualize the large-scale configuration of the given n rectangles in $\mathbb{R}^p$ by a *two-dimensional* display which approximates, as much as possible, a two-dimensional manifold which is close to the cloud of rectangles.

Remark:
In contrast to methods such as, e.g., principal component analysis, which work with projections of data onto low-dimensional *linear* hyperplanes, Kohonen maps may reproduce configurations where the data are concentrated near *non-linear* (e.g., bended or twisted) manifolds.

2 What is a Kohonen map? and what its benefits?

A Kohonen map is based on a rectangular lattice L with a given (small) number of rows and columns a and b, respectively, such that we have $m = a \cdot b$ vertices $P_1, ..., P_m$ in L (see Fig. 3). Each vertex P_i represents a small piece of the data, i.e. a subset C_i of items from $\{1, ..., n\}$ which have 'similar' rows in the data table such that C_i can be considered as a small homogeneous cluster (a 'mini-cluster') and can be represented by a 'typical' prototype z_i which is, in the case of interval data, typically defined as some type of 'mean rectangle' of all rectangles in the class C_i.

SYKSOM is a module for constructing m suitable mini-clusters $C_1, ..., C_m$ and an assignment of the classes C_i to the vertices P_i in a way such that, finally, all pairs of classes C_i, C_j which are *neighbouring in* $\mathbb{R}^p$ (i.e., have similar prototypes z_i and z_j) are represented by vertices P_i, P_j which are *neighbouring in* L (i.e. such that the 'path distance' between P_i and P_j along the edges of L is small). Insofar the lattice L, together with the class prototypes, approximates the overall-structure of the data in an illustrative way, at least to the extent as it is possible with a two-dimensional lattice.

The (mini-)clusters $C_1, ..., C_m$ constructed by the algorithm SYKSOM cannot be directly seen in the lattice. However, the very benefit from the Kohonen approach resides in the fact that we can mark each vertex of L with a special icon, letter, zoom star or diagram which describes the properties of the corresponding mini-cluster C_i (see Fig. 4).

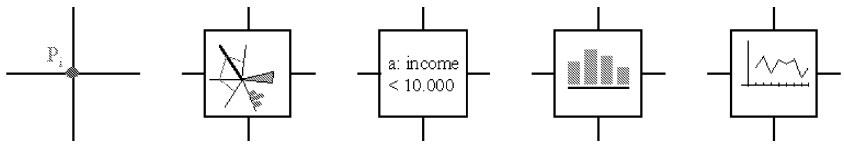

Fig. 4: *Typical icons, codes and diagrams that are used in SYKSOM for displaying specific properties of the clusters.*

Then we obtain a diagram as in Fig. 5 (here with letters for ease of presentation) which is called the *Kohonen map* or a *semantic map*.

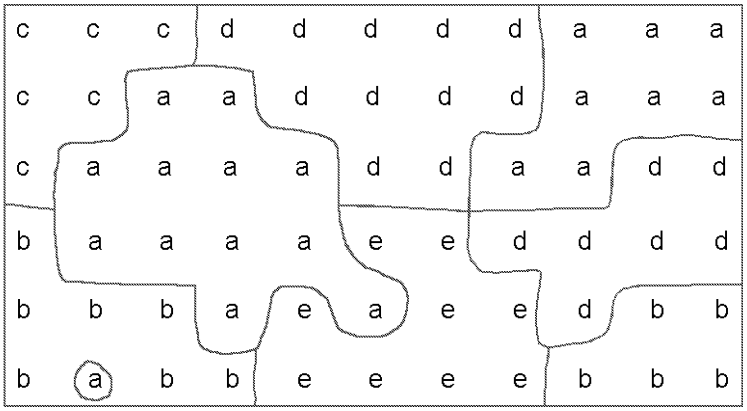

Fig. 5: *A Kohonen map where the clusters (vertices) have been marked with labels $a, b, ..., e$ which describe specific properties of the clusters (e.g., in the case of the variable 'income': a = 'poor', ..., e = 'rich') such that various homogeneous regions can be recognized in the map where, e.g., 'poor' people (= a) prevail.*

This landscape with icons, letters, diagrams etc. provides an excellent tool for resolving various exploratory problems, e.g.:

– to locate interesting patterns (prototypes) in the data set (in $\mathbb{R}^p$);

– to reveal similar mini-clusters and insofar similar patterns located in different domains of the underlying feature space;

– to recognize the distribution of properties in the feature space and to see the topological structure of the data table;

– to aggregate neighbouring mini-clusters in order to obtain large *global* clusters of items which divide the whole feature space into practicable and interpretable segments;

– to discover dependencies between variables, and tendencies in their values in different parts of the map;

– to discover 'aberrant' or outlying mini-clusters whose items behave quite differently from the bulk of data;

– to locate new items (new data rectangles) in the framework of the items of the original data table from which the map has been obtained.

3 SYKSOM's output: lists and Kohonen maps

Essentially, the module SYKSOM provides (only) some basic information:

1. A list with all m mini-clusters $C_1, ..., C_m \subset \{1, ..., n\}$.

2. A list which contains the mini-clusters C_i and their prototype rectangles z_i (with $2 \times p$ coordinates) in terms of lists and/or diagrams.

3. A list which describes the assignment of the mini-clusters $C_1, ..., C_m$ to the vertices $P_1, ..., P_m$ of the lattice L.

Remark:
Typically, the notation is chosen such that C_i is assigned to the vertex P_i. Note that the position of a vertex P_i in the lattice is sometimes given in terms of the integer coordinates (u, v) in the lattice L, with P_1 in the lower left corner (see Fig. 6). Thus:

$$\begin{array}{llllll}
P_{a(b-1)+1} \hat{=} (a,1) & P_{a(b-1)+2} \hat{=} (a,2) & \cdots & P_m \hat{=} (a,b) \\
\vdots \quad \vdots \quad \vdots \quad \vdots & & & \\
P_{a+1} \quad \hat{=} (2,1) & P_{a+2} \quad \hat{=} (2,2) & \cdots & P_{2a} \hat{=} (a,2) \\
P_1 \quad \hat{=} (1,1) & P_2 \quad \hat{=} (2,1) & \cdots & P_a \hat{=} (a,1)
\end{array}$$

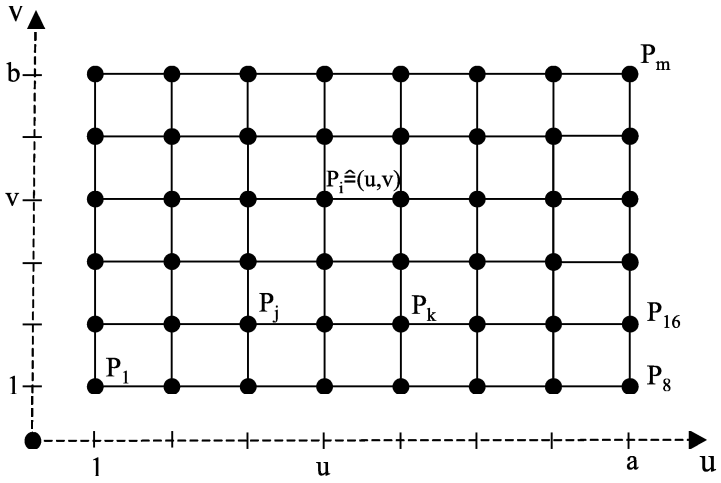

Fig. 6: *Coordinates in a rectangular lattice* $\mathcal{L}$ *with* $m = 8 \times 6 =$ 48 *vertices corresponding to* 48 *mini-clusters.*

As mentioned before, the very benefit from the SYKSOM module in the ASSO software results from the possibility to display visually the properties of the resulting (mini-)clusters at the location of the corresponding vertices in the lattice L (i.e., on the screen), either in terms of the original variables or in terms of additional variables which have been recorded in the data base. This display is called a *Kohonen map* and refers here to symbolic data.

In the following we describe some options for display which must typically be run by using the module VMAP after SYKSOM.

(a) We may select two of the p interval-type variables from which the clusters have been built, j and j' say, and can display the class prototypes $z_1, ..., z_m$ in the two-dimensional Euclidean space spanned by the selected coordinates j and j' (projection of the z_i's onto this space; see Fig. 7). This provides some idea about how clusters are distributed in this space and (after varying the variables j, j') in the whole space R^p.

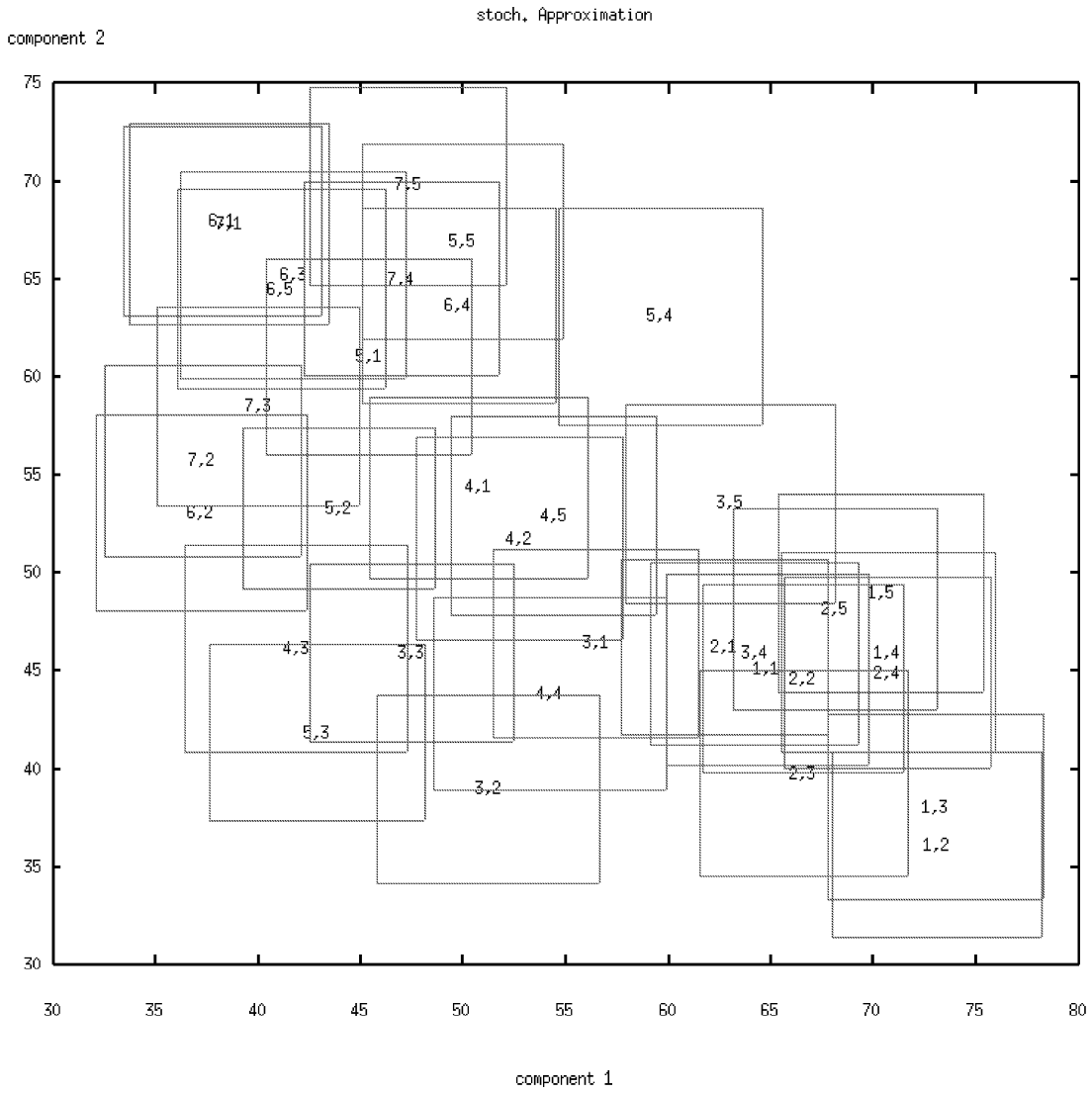

Fig. 7: *Projection of the class prototype rectangles $z_1, ..., z_m$ (here: $m = 7{\cdot}5 = 35$) onto the space of two selected variables $j = 1$ and $j' = 2$ (the underlying sample consisted of 10.000 randomly simulated rectangles in a square of $\mathbb{R}^5$).*

(b) By clicking the vertex P_i, the properties of the corresponding mini-class C_i are displayed in the form

– of a *zoom star* (where different axes correspond to the p variables of the class prototype z_i);

– of a *bar diagram* with a bar for each interval-type variable j whose position (on the vertical axis) is identical to the j-th side of the class prototype z_i (see Fig. 8; note that this display is useful only if the variables have comparable ranges);

– a *histogram* which describes the distribution of a variable in the clusters.

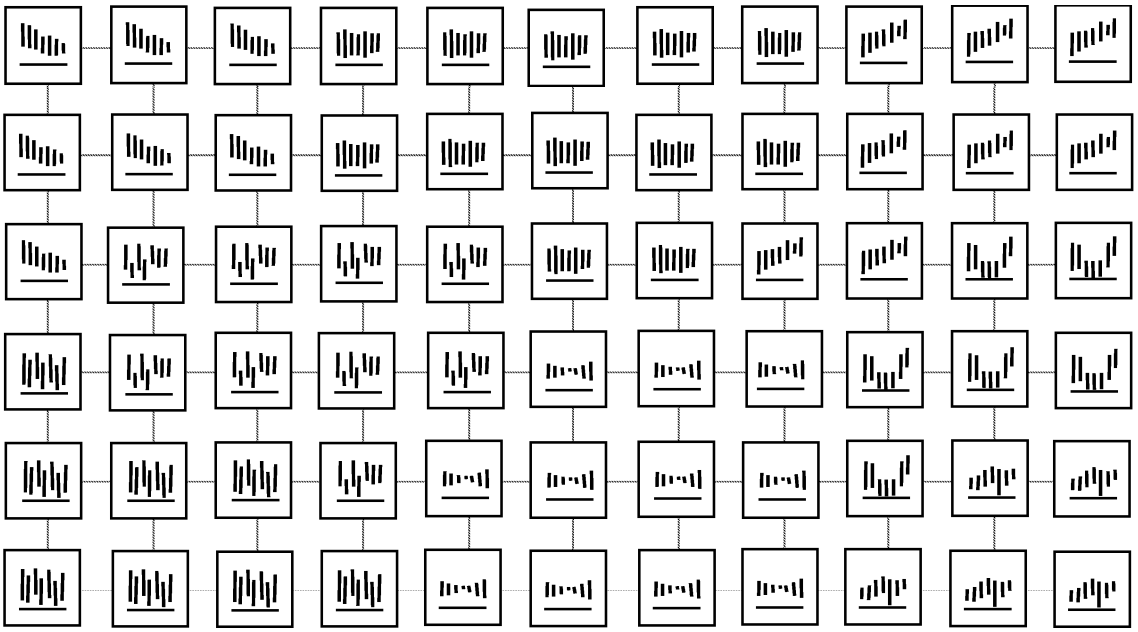

Fig. 8: *Display of the class-specific bar diagrams in the case of* $p = 8$ *interval-type variables, simultaneously for all mini-clusters* $C_1, ..., C_m$.

(c) We may also choose an 'interesting' selection of the p *original* interval-type variables from which the mini-clusters have been constructed, or from a set of *secondary* variables (e.g., qualitative variables), and display the structure of the classes C_i in terms of those variables. See, e.g., Fig. 9 for a histogram of a qualitative variable, or Fig. 10 for the case of a time series.

(d) As illustrated by Fig. 8–10, we may also display the class properties *simultaneously* for all m classes which facilitates the comparison among different vertices or mini-clusters.

(e) Suppose that for our n items, we have additionally recorded a *qualitative* secondary variable Y with categories $a, b, ..., e$ say. Then we can mark in the map, at each vertex P_i, the letter of the category which dominates in the corresponding cluster C_i (see Fig. 5). Then we obtain a clear view about the overall distribution of the categories $a, b, ..., e$ in the data set. VMAP allows to do this simultaneously for all vertices.

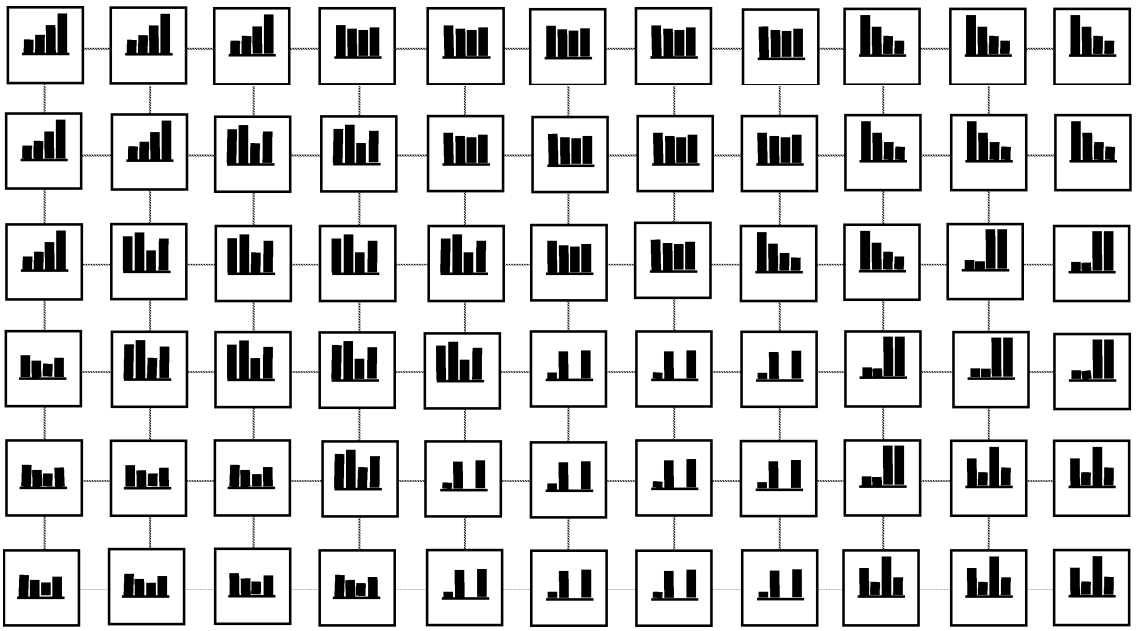

Fig. 9: *Display of the class-specific histogram for a qualitative variable with 4 categories, simultaneously for all mini-clusters* $C_1, ..., C_m$.

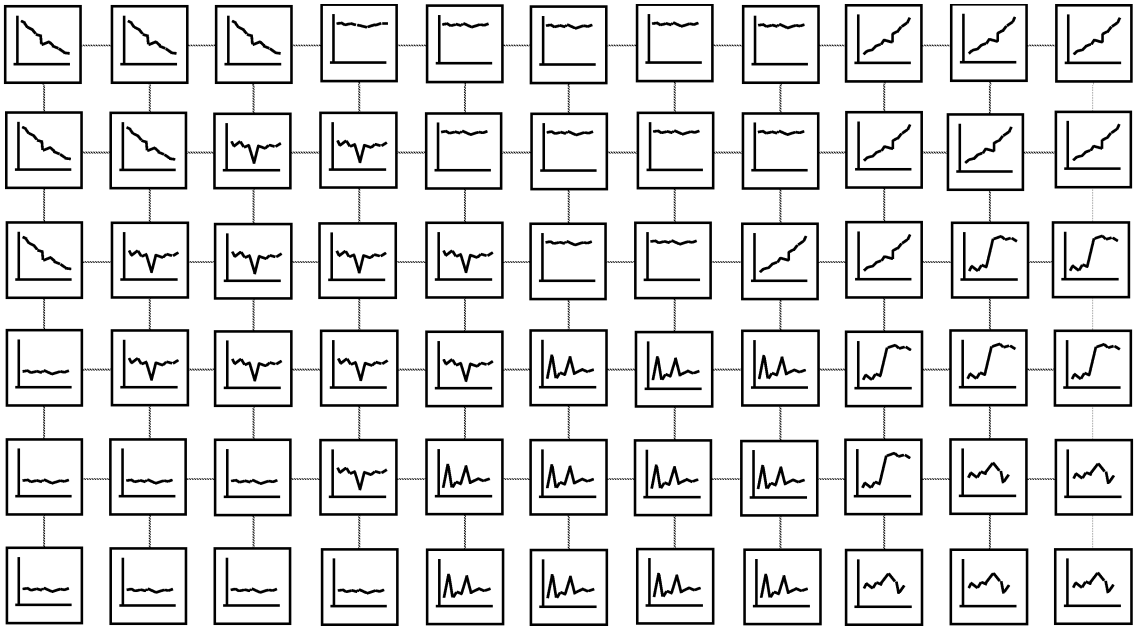

Fig. 10: *Display of the class-specific profiles of a time series which was recorded for each item (as secondary variables), simultaneously for all mini-clusters* $C_1, ..., C_m$.

(e) In a similar way, we may also select another variable (from the original interval data or from secondary variables), define 'categories' as above on the basis of that variable, and then display them in the Kohonen map as before.

Various other manipulations are under consideration (e.g., step-wise or hierarchical Kohonen maps).

4 How is SYKSOM working?

The Kohonen map approach is often assigned to the *neural network* domain or to the field of *sequential learning*. This resides in the fact, that the underlying algorithm (and also SYKSOM) includes the data in a *sequential order* $x_1, x_2, x_3, ...$, thus *not simultaneously* as a bulk of data. This sequential process works essentially as follows:

Suppose that, at some 'time' t, i.e. after having observed t data rectangles $x_1, ..., x_t$, a preliminary result has been obtained in the form of m mini-clusters $C_1^{(t)}, ..., C_m^{(t)}$ (assigned to the vertices $P_1, ..., P_m$) and m class prototypes $z_1^{(t)}, ..., z_m^{(t)}$. Now we 'observe' (include) the next data rectangle x_{t+1} and assign it to the 'closest' (i.e., most similar) class. Then we adapt all classes $C_1^{(t)}, ..., C_m^{(t)}$ and prototypes $z_i^{(t)}, ..., z_m^{(t)}$ in a suitable way and proceed to the next data x_{t+2} etc. After all n available data have been processed, we repeat the procedure in a cyclic way. After a (prespecified) number of cycles or when attaining a stationary result, the algorithm stops and provides the definite classes and prototypes.

When specifying the various steps of the algorithm, there may be several different options and choices. Thus, a more detailed description of SYKSOM is as follows:

• *Input:*
A sequence of interval vectors $x_1, x_2, ...$ in $\mathbb{R}^p$ of the type 1

• *Initial specifications for $t = 0$:*
Define an initial set of m (symbolic) cluster prototypes $z_1^{(0)}, ..., z_m^{(0)}$ in $?R^p$ and m empty classes $C_1^{(0)}, ..., C_m^{(0)}$ of items. For all i, assign the class $C_i^{(0)}$ to the vertex P_i of the lattice.

• *Iteration from t to $t+1$ items (data rectangles):*

1. Observe and include the $(t+1)$-th data rectangle x_{t+1}.

2. Determine, from the set of all current class prototypes $z_1^{(t)}, ..., z_m^{(t)}$ (i.e., those constructed from $x_1, ..., x_t$), a prototype $z_{i^*}^{(t)}$ which is closest to the new data rectangle x_{t+1} in the sense

$$d(x_{t+1}, z_{i^*}^{(t)}) = \min_{i=1,...,m} d(x_{t+1}, z_i^{(t)}).$$

Here d is a measure for the dissimilarity of the underlying rectangles. SYKSOM provides two different choices for d.

3. Assign the item $t+1$ (i.e., the data rectangle x_{t+1}) to the class $C_{i^*}^{(t)}$ leaving all other classes unchanged. This yields the new classes

$$\begin{aligned} C_{i^*}^{(t+1)} &:= C_{i^*}^{(t)} \cup \{t+1\} \qquad \text{for } j = i^* \\ C_j^{(t+1)} &:= C_j^{(t)} \qquad\qquad\qquad \text{for all } j \neq i^*. \end{aligned}$$

4. Update the prototype $z_{i^*}^{(t)}$ of the old, modified class $C_{i^*}^{(t)}$ as well as *all other* prototypes $z_j^{(t)}$, with a suitable 'updating formula' which essentially means shifting the class prototy-

pes more or less towards the new data vector x_{t+1}, *in dependence* on the neighbourhood between the corresponding vertices P_{i^*} and P_j in the lattice L. Without mentioning the exact updating formula here, we can say that the modification will be large for all classes $C_j^{(t+1)}$ where P_j is close to P_{i^*} in L, whereas the modification is small (or even 0) if P_j is far away from P_{i^*} in L.

This updating step yields m new class prototypes (rectangles) in $\mathbb{R}^p$:

$$z_1^{(t+1)}, z_2^{(t+1)}, ..., z_m^{(t+1)}.$$

• *Iteration and stopping:*
The steps 1. to 4. are iterated until 'convergence' according to a given stopping criterion, eventually after repeating several times the process for all available data (in case there is a finite number n).