CHAPTER FOUR

NEURAL NETWORK SELF-ORGANIZING MAP

4.1 INTRODUCTION

Neural networks have been successfully applied by many authors in solving pattern recognition problems. Unsupervised classification is an important branch of pattern recognition, which unfortunately has received less attention as an application of neural networks. In the analysis of poverty there is a need to classify households into several classes while no knowledge is known a priori what these classes are, nor are there any training samples with known classification, thus the need to use unsupervised methods of classification exist. Among the many neural network models available the self organizing map is selected as the one most suitable for unsupervised applications. Among the architectures and algorithms suggested for artificial neural networks, the self organizing map has the special property of effectively creating spatially organized internal representations of various features of input signals and their abstractions. The self organizing process can also discover semantic relationships and has been particularly successful in various pattern recognition tasks.

The network architectures and signal processes used to model nervous systems can be roughly divided into three categories:

- Feed forward networks transform sets of input signals into sets of output signals using externally supervised adjustment of the system parameters.
- In feedback networks the input function information defines the initial activity state of a feedback system and after state transitions the asymptotic final state is identified as the outcome of the computation.
- When the neighbouring cells in a neural network compete in their activities by means of mutual lateral interactions they develop adaptively into specific detectives of different signals patterns. This category of learning is called competitive, unsupervised or self organizing. The self organizing map discussed in this chapter belongs to this third category.

In this chapter the self organizing map is presented as a new effective modelling tool for the visualization of high dimensional data. Non linear statistical relationships between high dimensional data are converted into simple geometric relationships of their image points on a low dimensional display, usually a two dimensional grid of nodes. As the self organizing map compresses information while preserving the most important topological and metric relationships of the primary data elements, it may also be thought to produce some types of abstractions. These visualizations and abstractions can be utilized to measure multi-dimensional poverty.

This chapter applies the self organizing map algorithm to the Republic of South Africa Census 2001 data set, examining the data from a data mining point of view. The scope of this chapter is to discuss what can be learned about the levels of poverty of the different households. The self organizing map is used to categorise the different households into the many grades or shades of poverty. The main advantages of the self organizing map are to group similar entities together.

The Poverty Map was an application of the self organizing map that shows a map of the world based on mostly economic indicators.

Figure 4.1.1 shows the resulting map as a self organizing map coloured with values obtained from the self organizing map evaluation. The Poverty Map was obtained by 39 indicators selected from the World Bank Development Indicators (World Bank 2001a).

Figure 4.1.2 is the World Bank self organizing map plotted on the world map with the same colours that were generated in the self organizing map analysis. The light colours indicate low levels of poverty and the darker shades indicate higher levels of poverty.

Figure 4.1.1: World Bank self organizing map

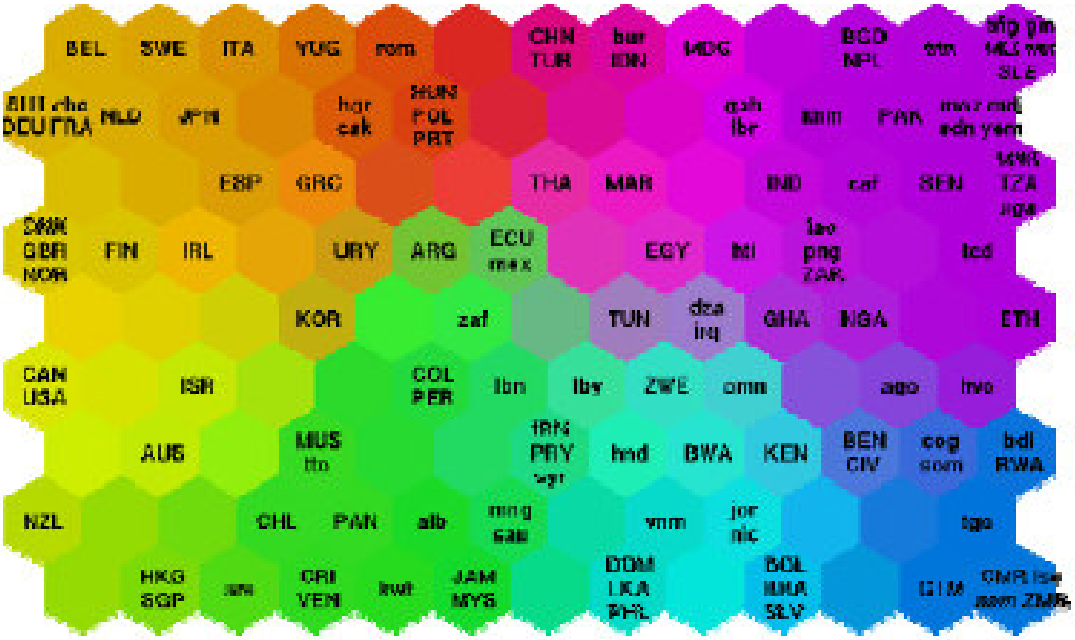
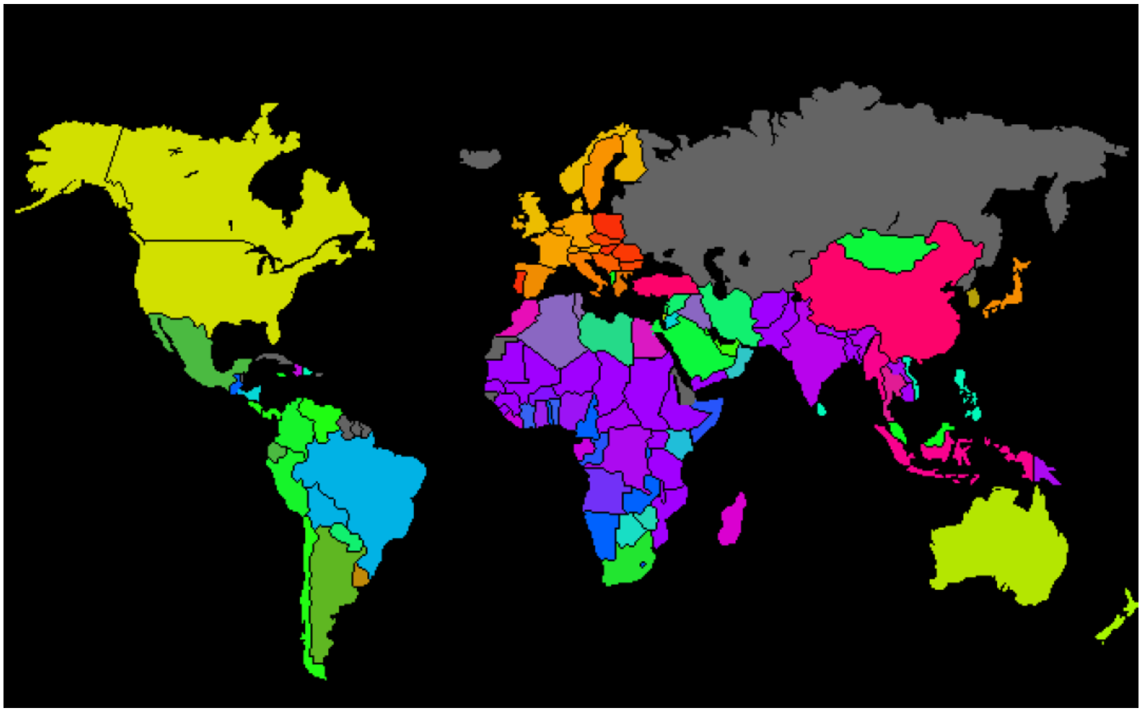


Figure 4.1.2: World Map with results from the self organizing map analysis



Most of the calculations described in this chapter have been performed with the data mining software tool, SAS Enterprise Miner, and the analytical package, SAS Enterprise Guide.

In the SAS Enterprise Miner version 4.3 the SOM/Kohonen node belongs to the Model category of the SAS SEMMA (Sample, Explore, Modify, Model and Assess) data mining process. The SOM/Kohonen node is used to perform unsupervised learning by using Kohonen vector quantization, Kohonen self organizing map, or Batch self organizing map with Nadaraya-Watson or local-linear smoothing. Some of the methodology described in this thesis relies heavily on the SAS online help documentation.

In this section the term step applies to the SAS computations that are done while reading a single case and updating the cluster seeds and the term iteration applies to the SAS computations that are done while reading the entire data set once and updating the cluster seeds.

Section 4.2 introduces the methodology of the Kohonen vector quantization followed by the analysis and results from the application to the data from the Republic of South Africa 10% sample of Census 2001.

Section 4.3 describes the methodology of the Kohonen self organizing map and the analysis applied to the Republic of South Africa Census 2001 data.

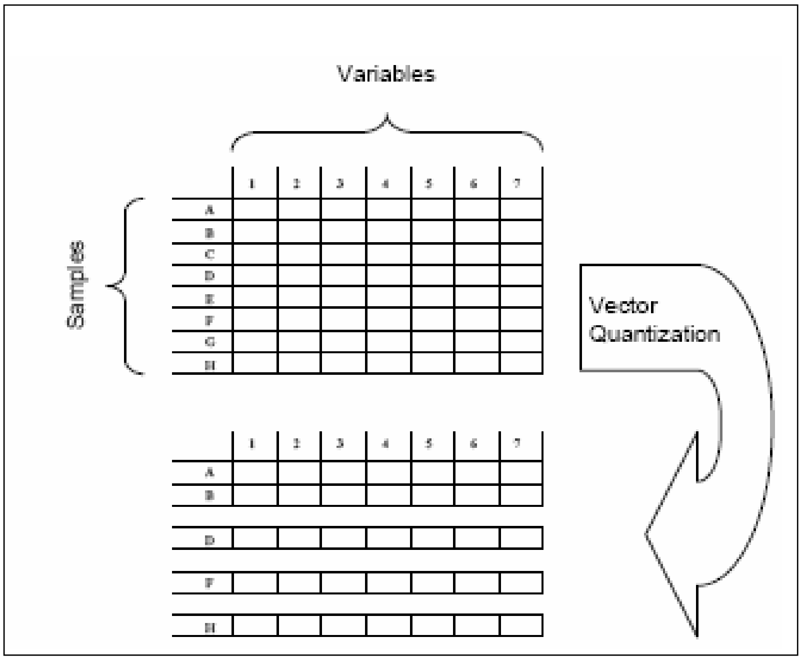
Section 4.4 describes the methodology of the Batch self organizing map and its application on the Republic of South Africa Census 2001 data.

Section 4.5 summarizes results and findings of the chapter.

4.2 KOHONEN VECTOR QUANTIZATION

Vector quantization can be describes as the task of finding a suitable subset that represents a larger set of data vectors. Vector quantization aims at reducing the number of sample vectors or substituting them with representative centroids as shown in figure 4.2.1. The vector quantization method reduces the original set of 8 samples to 5 samples. The resulting centroids can also be an approximation of the vectors assigned to them, for example, their average vector quantization is closely related to clustering.

Figure 4.2.1: Vector quantization reduction



Visualization is very important for data mining as a direct plot of a set of data can provide insights into its structure and underlying distribution that inspection of the numerical data table cannot. However, data sets cannot be visualized on a sheet of paper or on a monitor if their dimensionality is higher than 2.

4.2.1 Methodology

Vector quantization networks are competitive networks that can be viewed as unsupervised density estimators or autoassociators (Kohonen 2001). Each competitive unit corresponds to a cluster, the centre of which is called a codebook vector or cluster seed.

Vector quantization is a classical signal approximation method that usually forms a quantized approximation to the distribution of the input data vectors , $\mathbf{x} \in \Re^{\mathbf{n}}$, using a finite number of so called codebook vectors, $\mathbf{m_i} \in \Re^{\mathbf{n}}$, (i=1,2,…,k) (Kohonen 2001). Once the codebook vector is chosen, the approximation of x requires finding the codebook vector m_c closest to x in the input space determined by the Euclidean distance:

$$||x\text{-}m|| = \min_i \{||x\text{-}m_i||\} \tag{4.1}$$

The optimal selection of the m_i minimizes the average expected square of the quantization error, which is defined as follows:

$$E = \int || x - m_c ||^2 p(x) dx \tag{4.2}$$

where

the integral is taken over the complete metric x space,

dx is the n-dimensional volume differential of the integration space, and

p(x) is the probability density function of x.

Kohonen's learning law is an online algorithm that finds the cluster seed closest to each training case and moves the winning seed closer to the training case. The seed is moved some proportion of the distance between it and the training case; the proportion is specified by the learning rate.

Let

C_j^s be the seed for the jth cluster on the sth step,

X_i be the input vector for the ith training case, and

L^s be the learning rate for the sth step.

The training case X_i is selected and the index n of the winning cluster is determined by

$$n = \arg\min_j \| C_j^s - X_i \| \tag{4.3}$$

The Kohonen update formula is defined as follows:

$$C_n^{s+1} = C_j^s (1 - L^s) + x_i L^s \tag{4.4}$$

for all non winning clusters

$$C_n^{s+1} = C_j^s \tag{4.5}$$

In SAS Enterprise Miner, the Kohonen vector quantization is often used for offline learning in which case the training data is stored and Kohonen's learning law is applied to each case in turn, cycling over the data set many times, that is, incremental training.

4.2.2 Analysis

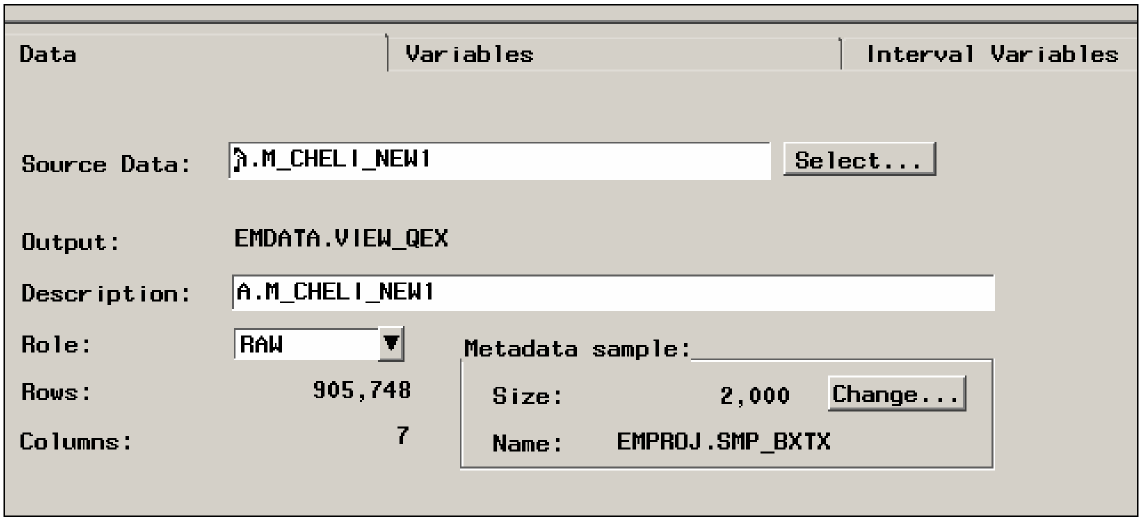
In this section the Kohonen vector quantization technique is applied to the 10% sample data from the Republic of South Africa 2001 Census. In the sample there are 905 748 households and four attributes were selected to measure the dimension of poverty: "access to basic services".

The analysis is conducted using SAS Enterprise Miner's SOM/Kohonen node. The Kohonen vector quantization technique is illustrated using the following four attributes to create a multi-dimensional measure of poverty:

- Access to water,
- Energy source for cooking,
- Toilet facilities, and
- Refuse removal.

The membership function proposed by Cheli and Lemmi (1995) is applied to the four attributes. Figure 4.2.1 shows that the data tab of the SAS Enterprise Miner SOM/Kohonen node. The SAS data set used in this analysis is called M_Cheli_New1 and is stored in the SAS library named A.

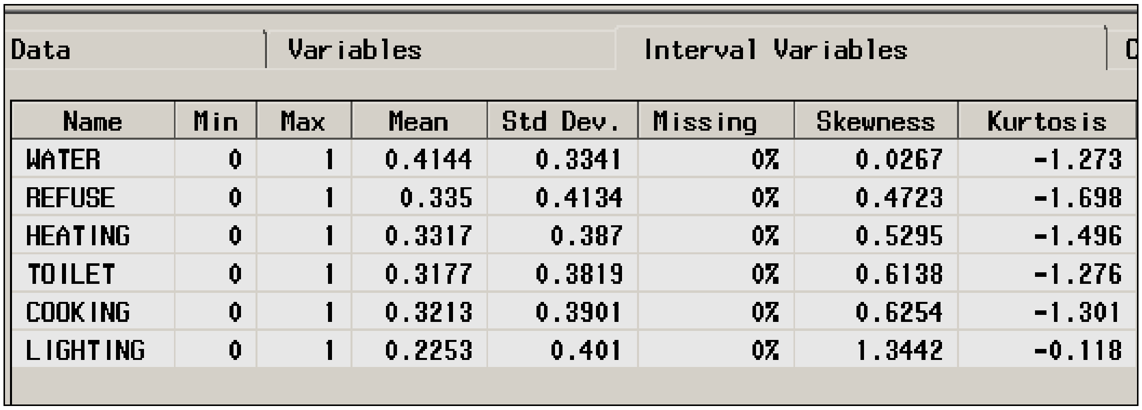
Figure 4.2.1: Input data set: Data tab



Source Data: A.M_CHELI_NEW1
Output: EMDATA.VIEW_QEX
Description: A.M_CHELI_NEW1
Role: RAW
Rows: 905,748
Columns: 7
Metadata sample:
Size: 2,000
Name: EMPROJ.SMP_BXTX

There are 905 748 households in the data set. A sample of 2 000 households is selected to generate the metadata. The option is available to use the entire data set to create the metadata or to change the sample size from 2 000 to any number that the researcher wishes to use.

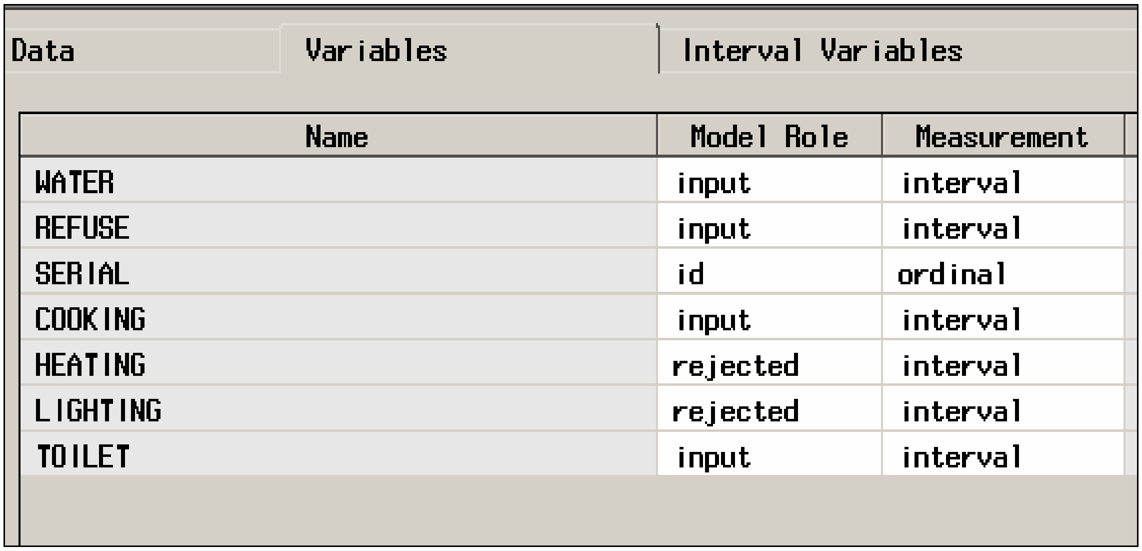
Figure 4.2.2: Input data set: Interval Variables tab.



Name	Min	Max	Mean	Std Dev.	Missing	Skewness	Kurtosis
WATER	0	1	0.4144	0.3341	0%	0.0267	-1.273
REFUSE	0	1	0.335	0.4134	0%	0.4723	-1.698
HEATING	0	1	0.3317	0.387	0%	0.5295	-1.496
TOILET	0	1	0.3177	0.3819	0%	0.6138	-1.276
COOKING	0	1	0.3213	0.3901	0%	0.6254	-1.301
LIGHTING	0	1	0.2253	0.401	0%	1.3442	-0.118

Figure 4.2.2 shows the interval variables tab in the input data set. This tab lists the variables that are in the data set and shows the descriptive statistics together with the percentage of missing values. In this calculation there are no missing values. The membership function is used in the calculation. The minimum value of the membership function will always be zero and the maximum value will always be 1.

Figure 4.2.3: Input data set: Variables tab



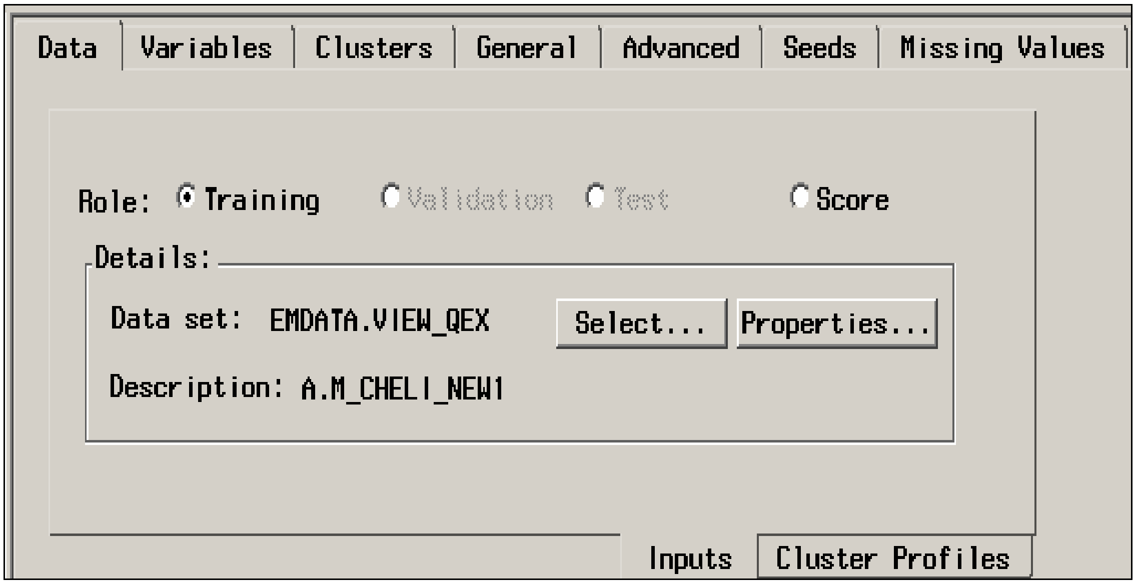
Name	Model Role	Measurement
WATER	input	interval
REFUSE	input	interval
SERIAL	id	ordinal
COOKING	input	interval
HEATING	rejected	interval
LIGHTING	rejected	interval
TOILET	input	interval

Figure 4.2.3 shows the variables tab of the input data set. In this data set there are seven variables. SAS Enterprise Miner automatically recognises the variable Serial as an

identification variable and selects the model role as "id". The other six variables are given the model role of "input". In this analysis only four attributes are used and their model role remains as "input" and the model role for the other two attributes is set to "rejected". The measurement role for each attribute is set to "interval".

Figure 4.2.4 shows the data tab of the SOM/Kohonen node. The role of the data set is set to training. The properties tab gives the metadata which includes the date when the data set was created and modified. This tab has a table view option to view the variables in the data set.

Figure 4.2.4: SOM/Kohonen node: Kohonen vector quantization: Data tab



Role: ● Training ○ Validation ○ Test ○ Score

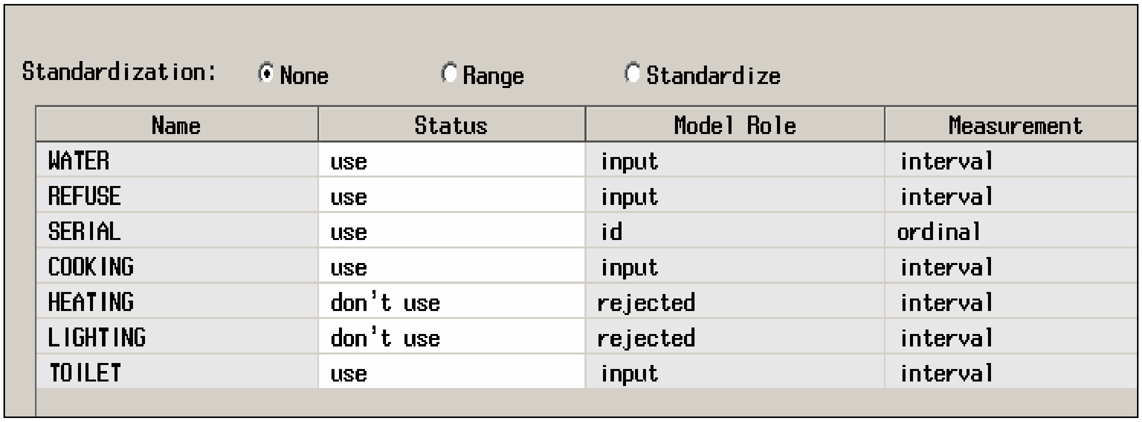
Details:

Data set: EMDATA.VIEW_QEX

Description: A.M_CHELI_NEW1

Figure 4.2.5 shows the variables tab of the SOM/Kohonen node. All the variables are listed and the variables that were rejected in the input data node are shown as rejected in the model role with the status shown as don't use. The status column is not greyed allowing for the status of the variables to be changed to use. This tab also has the option to standardize the variables. All the membership function values for the attributes are between zero and one, therefore standardization is not necessary and the "none" option is selected.

Figure 4.2.5: SOM/Kohonen node: Kohonen vector quantization: Variables tab

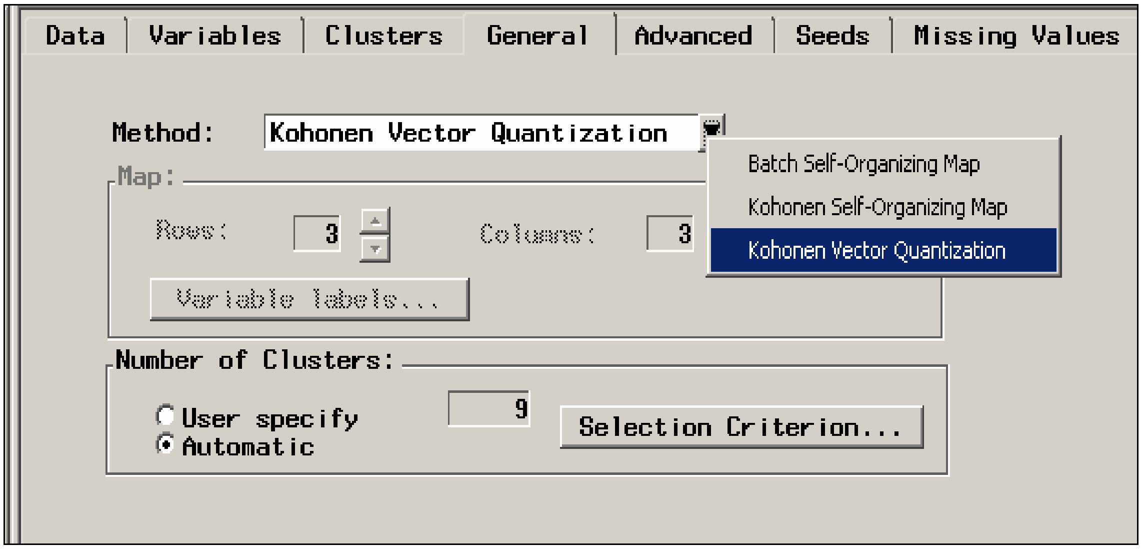


Standardization: ● None ○ Range ○ Standardize

Name	Status	Model Role	Measurement
WATER	use	input	interval
REFUSE	use	input	interval
SERIAL	use	id	ordinal
COOKING	use	input	interval
HEATING	don't use	rejected	interval
LIGHTING	don't use	rejected	interval
TOILET	use	input	interval

Figure 4.2.6 shows the general tab in the SOM/Kohonen node. For this analysis Kohonen vector quantization is selected as the method. In the Kohonen vector quantization networks, the number of clusters could be user specified or automatically selected. If the automatic option is chosen then the selection criteria tab must be used to specify the various options, for example, the minimum and maximum number of clusters and the clustering cubic criterion cut-off.

Figure 4.2.6: SOM/Kohonen node: Kohonen vector quantization: General tab



Method: Kohonen Vector Quantization

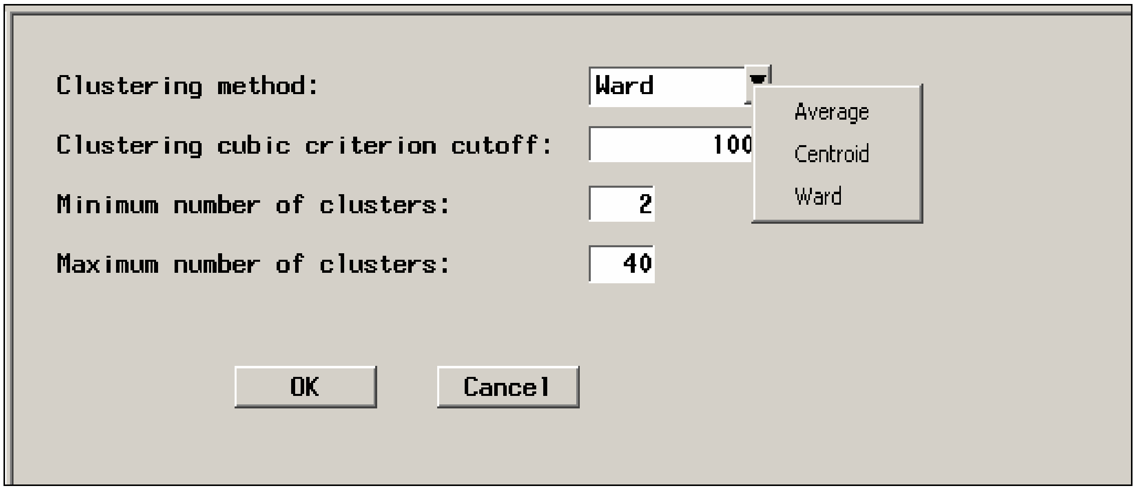
Map:

Rows: 3 Columns: 3

Number of Clusters:

○ User specify 9
● Automatic

Figure 4.2.7: SOM/Kohonen node: Kohonen vector quantization: Selection Criteria tab



Clustering method: Ward

Clustering cubic criterion cutoff: 100

Minimum number of clusters: 2

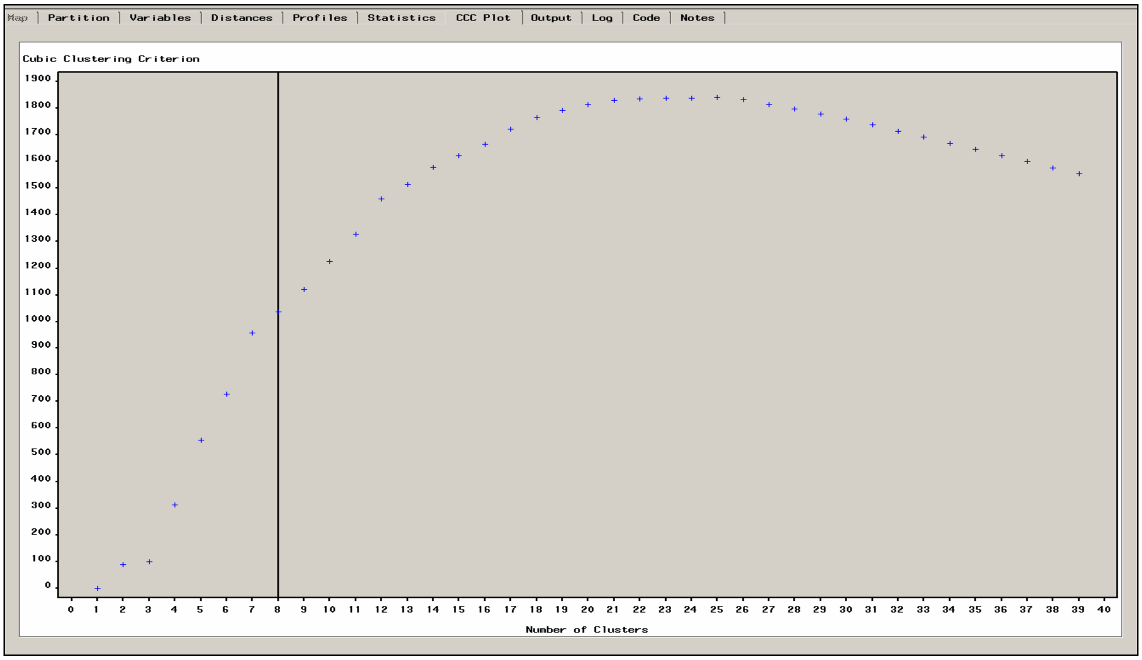
Maximum number of clusters: 40

Figure 4.2.7 shows the selection criteria tab of the SOM/Kohonen node. The available clustering methods are Average, Centroid and Ward methods. In this calculation the Ward method is selected.

The minimum number of clusters is specified as two and the maximum number of clusters is specified as forty. A cut-off value for the cubic clustering criterion (CCC) must be stated. If the cubic clustering criterion suggests the number of clusters below the minimum number of clusters then the minimum number of clusters will be created. Likewise if the cubic clustering criterion suggests a higher number of clusters than the maximum number of clusters then the maximum number of clusters will be created. In this analysis the cubic clustering criterion is set to 1 000.

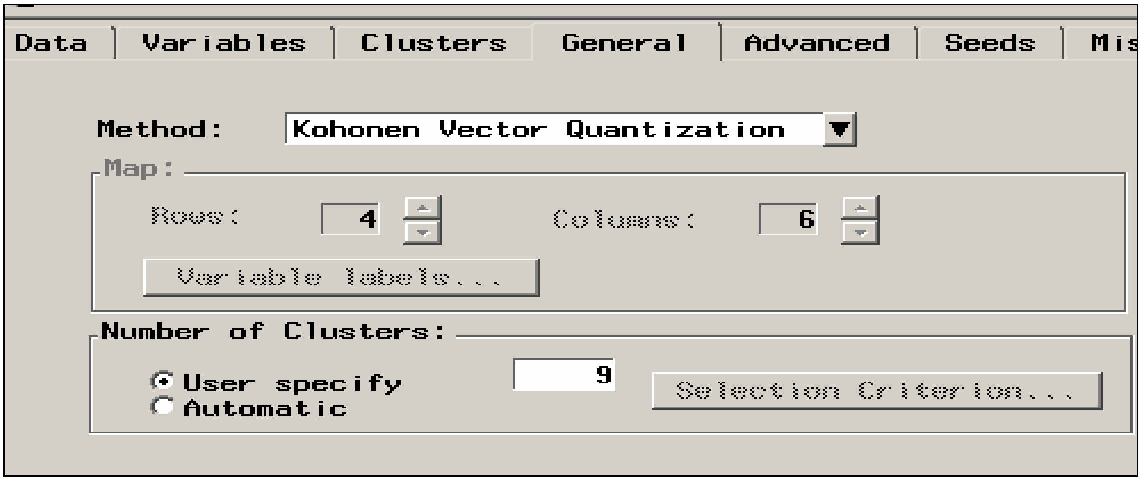
Figure 4.2.8 shows the cubic clustering criteria plot for the Kohonen vector quantization analysis. The cubic clustering criterion cut-off of 1 000 suggests that the number of clusters to be created is 8. If the cubic clustering criterion cut-off was set as 500 then the number of clusters created will be four.

Figure 4.2.8: SOM/Kohonen node: Kohonen vector quantization: CCC Plot tab



To make a meaningful comparison with the results of later sections the option is set to user specified and the number of clusters is set to 9. This can be seen in figure 4.2.9. Note that the map option is dimmed as this is only applicable to the Kohonen and Batch self organizing maps.

Figure 4.2.9: SOM/Kohonen node Kohonen vector quantization: User specify tab



Method: Kohonen Vector Quantization

Map:

Rows: 4 Columns: 6

Number of Clusters:

● User specify 9
○ Automatic

The SOM/Kohonen node is run for the Kohonen vector quantization analysis and the following results are obtained:

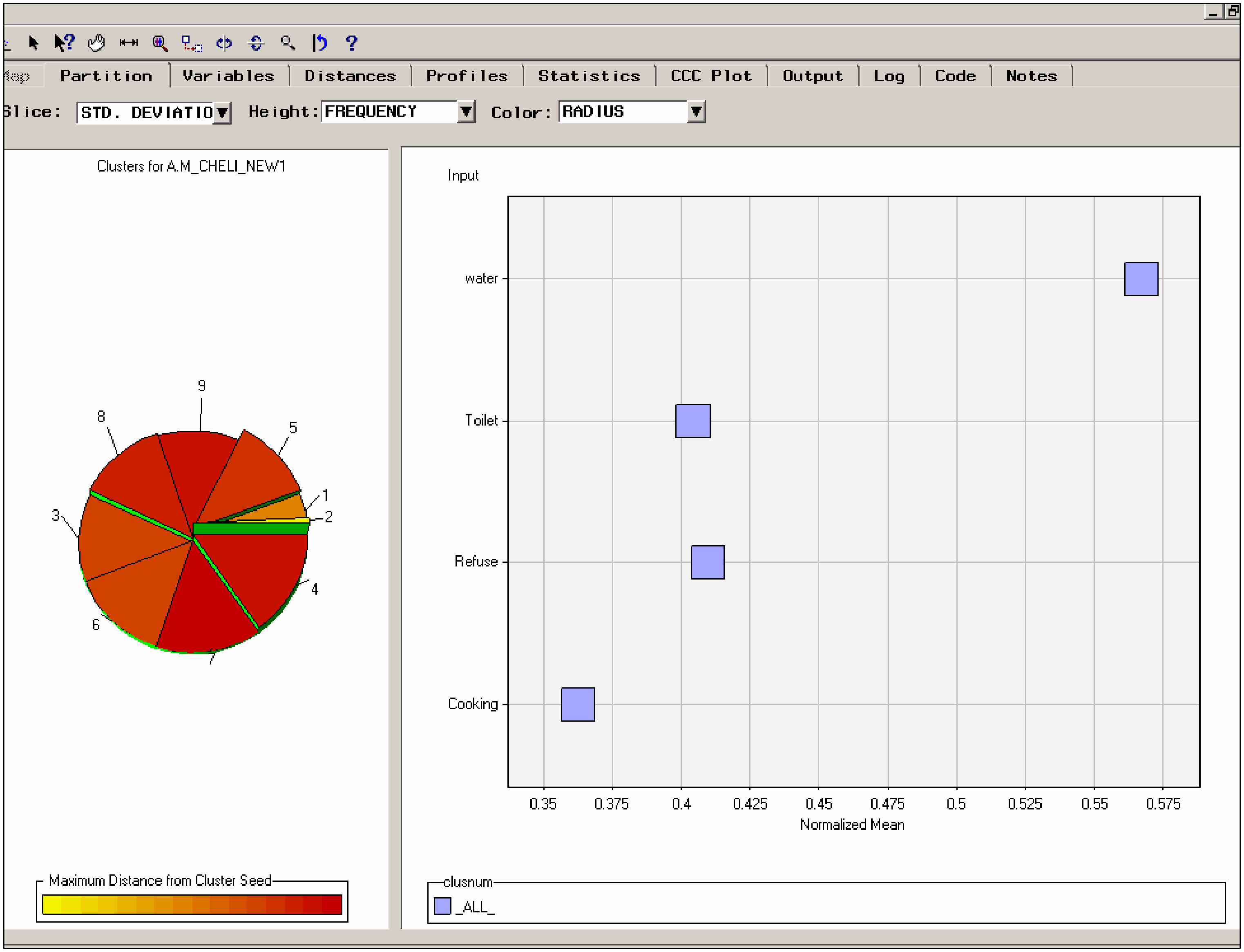
- The partition tab contains a graphical representation of the key characteristics of the clusters that are generated from the vector quantization method.
- The variables tab lists all the inputs that were used in the Kohonen vector quantization analysis.
- The Distance Tab provides a graphical representation of the size of each cluster and the relationship among clusters.
- The Profile Tab provides a graphical representation of the categorical and interval variables.
- The Statistics Tab displays information about each cluster in a tabular format.
- The CCC Plot displays a plot of the Cubic Clustering Criterion, which is plotted against the number of clusters that the SOM/Kohonen node automatically generates.
- The Output Tab displays the output that is generated from running the SAS/STAT DMVQ procedure.

Figure 4.2.10 shows the Kohonen vector quantization partition tab of the SOM/Kohonen node results browser. On the left is the three dimensional pie chart and on the right is the plot of the input means over all the clusters.

The three dimensional pie chart in figure 4.2.10 has the following settings:

- Height is determined by the frequency.
- Colour is set to Radius, which is the distance from the farthest cluster member to the cluster seed.
- Slice is set to standard deviation, which is the root mean square standard deviation distance between cases in the cluster.

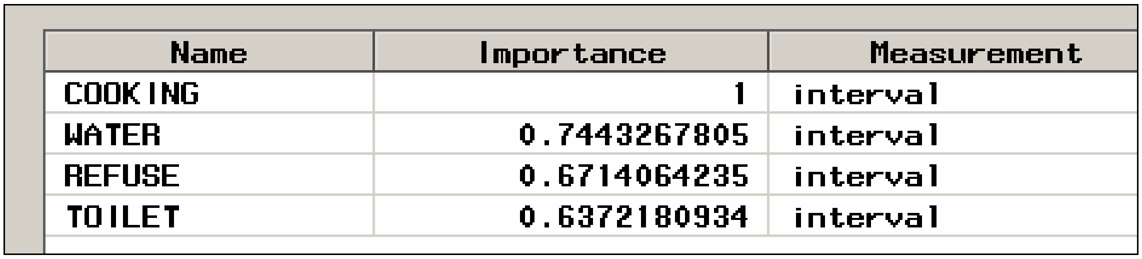
Figure 4.2.10: SOM/Kohonen node: Kohonen vector quantization: Partition tab



Slice: STD. DEVIATIO
Height: FREQUENCY
Color: RADIUS

The grid plot on the right of figure 4.2.10 displays the plot of the input means for the four attributes that are used in the analysis over all clusters. The input means are normalized to fall between the values 0 to 1. The attributes are ranked according to the normalized input means with the attribute with the largest normalized input means first. In this case the attribute access to water is first with the largest normalized input mean.

Figure 4.2.11: Kohonen vector quantization: Variables tab



Name	Importance	Measurement
COOKING	1	interval
WATER	0.7443267805	interval
REFUSE	0.6714064235	interval
TOILET	0.6372180934	interval

Figure 4.2.11 is the variables tab of the Kohonen vector quantization results. The four attributes used in the analysis are shown with an importance value. The importance value ranges between zero and one with the attribute that has the largest contribution to the cluster formation having an importance value close to one. In this analysis the attribute energy source for cooking has an importance value of 1 and the other attributes have fairly high importance values, suggesting that they have also contributed to the cluster formation.

In the statistics tab the cluster segments are given together with the frequency for each segment and the cluster means for each attribute. The statistics for the Kohonen vector quantization results are shown table 4.2.1. The last column of table 4.2.1 shows the Euclidean distance measure for each cluster measured back to the origin and sorted in ascending order. The clusters in the table are ranked from the households experiencing the least poverty to the households experiencing maximum deprivation with respect to the poverty dimensions “access to basic services”.

Table 4.2.1: Kohonen vector quantization: Statistics tab

Poverty Groupings	**VQ_Clusters**	**Frequency**	**Water**	**Refuse**	**Cooking**	**Toilet**	**Distance**
no deprivation	2	252 043	0.000	0.001	0.001	0.002	0.002
very little deprivation	1	127 488	0.473	0.002	0.002	0.011	0.473
little deprivation	8	76 209	0.383	0.003	0.598	0.021	0.710
below average deprivation	3	25 111	0.452	0.006	0.027	0.760	0.885
average deprivation	7	37 236	0.345	0.824	0.021	0.098	0.899
above average deprivation	9	51 495	0.620	0.011	0.575	0.765	1.141
extreme deprivation	6	46 063	0.665	0.832	0.771	0.143	1.323
very extreme deprivation	4	121 396	0.678	0.839	0.286	0.727	1.332
maximum deprivation	5	168 707	0.748	0.852	0.926	0.810	1.673

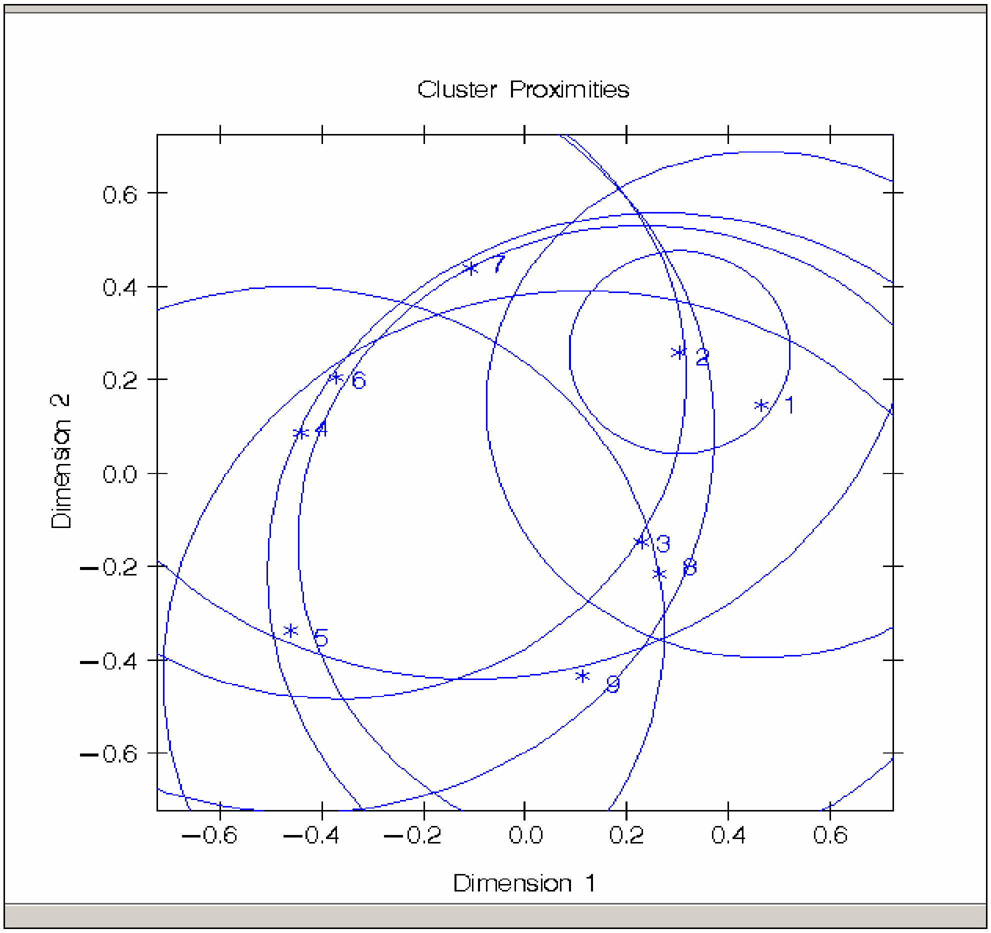
In cluster 2 there are 252 043 household that experience zero deprivation because all the attributes have a cluster mean of zero or very close to zero. In cluster 5 there are 168 707 households that experience maximum deprivation in respect of basic services. Cluster 5 satisfies the intersection definition of poverty, that is, all the households experience poverty in every attribute.

Clusters 1, 3, 4, 6, 7, 8, 9 satisfy the union definition of poverty, that is, the households experience poverty in at least one attribute. For example, the households in cluster 1 experience poverty only in the attribute “access to water” while the households in cluster 6 experience poverty in three attributes, “access to water”, “refuse removal” and “energy source for cooking”. This analysis technique divides the households into 9 clusters each experiencing different levels of poverty.

The Kohonen vector quantization results distance tab shown in figure 4.2.12 gives a graphical representation of the size of each cluster and the relationship among the clusters. The axis is determined from the multi-dimensional scaling analysis. The asterisks represent the cluster centres and the circles represent the cluster radii. The radius of each cluster is dependent on the most distant case in that cluster.

Cluster 2 has the smallest radii indicating that all the household attributes are close together. In this cluster all the households experience zero poverty and the membership function values are very close to zero.

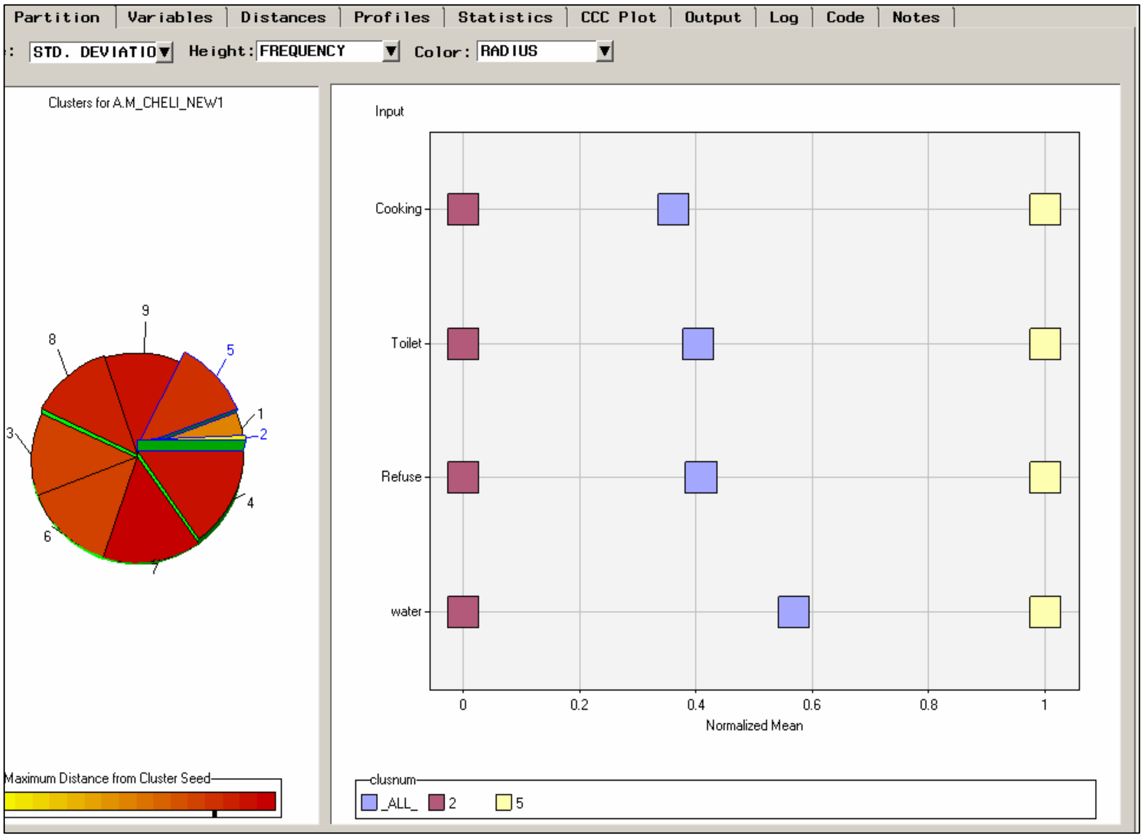
Figure 4.2.12: Kohonen vector quantization: Distance tab



The radii might give the impression that the clusters overlap, but in fact each household is assigned to only one cluster. Figure 4.2.12 shows the clusters with households that are experiencing the most deprivation on the extreme left, that is, clusters 4, 5 and 6. The clusters plotted on the right, cluster 2 and cluster 1 comprise households that experience zero deprivation.

The normalized means of the cluster with households that experience the least deprivation (cluster 2) and the cluster with households that experiences the most deprivation (cluster 5) are compared in figure 4.2.13.

Figure 4.2.13: SOM/Kohonen node: Kohonen vector quantization: Partition tab



: STD. DEVIATIO▼ Height: FREQUENCY ▼ Color: RADIUS ▼

The plot ranks the attributes based on how spread out the input means for the selected clusters relative to the overall input means are. The input mean of the attribute with the greatest spread is “cooking” and is listed first and the input mean of the attribute with the smallest spread is “water” and is listed last. The input means for cluster 2 are all either zero or very close to zero, while the input means for cluster 5 are all equal to one.

From a poverty measurement point of view on the pie chart, it is difficult to identify the Kohonen vector quantization cluster that has the best off households and the cluster that has the most deprived households. To overcome this problem a Kohonen self organizing map is generated.

4.3 KOHONEN SELF-ORGANIZING MAPS

The self organizing map is a very popular artificial neural network (ANN) algorithm based on unsupervised learning. The self organizing map has proven to be a valuable tool in the visualization of high dimensional data in data mining and in the larger field of Knowledge Discovery in Databases (KDD). It was originally developed by Kohonen in 1985 and is mostly used to convert the non linear statistical relationships between high dimensional data into simple geometric relationships of their image points on a low display, usually a regular two dimension grid of nodes. It has been subject to extensive research and has applications ranging from full text and financial data analysis, pattern recognition, image analysis, process monitoring and control to fault diagnosis. The self organizing map training algorithm is very robust; although there are some choices to be made regarding training length, map size and other parameters.

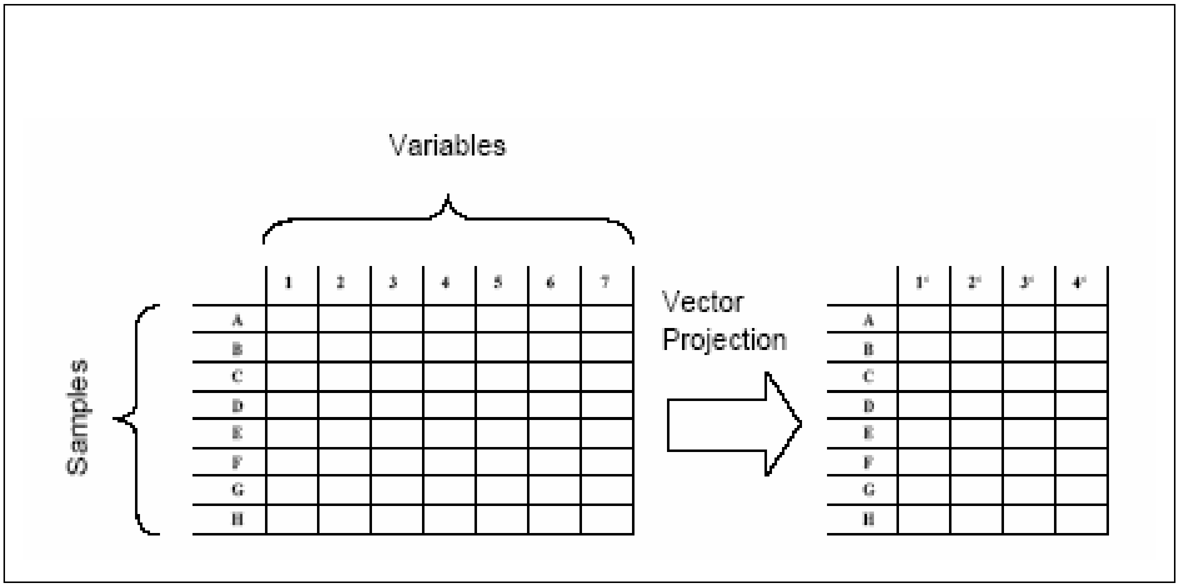
A self organizing map is a competitive network that provides a topological mapping from the input space to the clusters that are intended for clustering, visualization, and abstraction (Kohonen 2001).

The self organizing map was inspired by the way in which various human sensory impressions are neurologically mapped into the brain such that spatial or other relations among stimuli correspond to spatial relations among the neurons. In a self organizing map, the neurons (clusters) are organized into a two-dimensional grid. The grid exists in a space that is separate from the input space; any number of inputs can be used, provided the number of inputs (attributes) are greater than the dimensionality of the grid space.

A self organizing map tries to find clusters such that any two clusters that are close to each other in the grid space have seeds close to each other in the input space. Their learning algorithm is computationally extremely light, and consists of a low-dimensional grid that contains a number M of neurons. In this chapter, only the two dimensional grid will be considered, since grids of higher dimensions are difficult to visualize. The neurons are arranged in a rectangular way in figure 4.3.1, the position of the neurons in the grid. The distances between the neurons and the neighbourhood relations are very important for the learning algorithm. Each neuron has a so-called prototype vector (also codebook vector) associated with it, which is a vector of the same dimension as the input data set that approximates a subset of the training vectors.

Vector projection aims at reducing the input space dimensionality to a lower number of dimensions in the output space, and mapping vectors in input space to this lower dimensional space. In this section only two dimensional output spaces for visualization is discussed. Figure 4.3.1 shows the principle of vector projection, reducing a data set with seven variables to a data set with four variables; the resulting variables are usually obtained by complex algorithms.

Figure 4.3.1: The vector projection method of reduction



Vector projection leads to loss of information in almost all cases but the vector projection mapping occurs in a way that the distances in input space are preserved as well as possible, such that similar vectors in input space are mapped to positions close to each other in output space, and vectors that are distant in input space are mapped to different coordinates in output space. The algorithms emphasize the preservation of distances of vectors that are close to each other, while not necessarily preserving relatively large distances. The self organizing map is a vector projection method.

4.3.1 Methodology

The dimension of the sample vectors is the input dimension, and is much larger than two the dimension of the grid named output dimension. The self organizing map is a vector projection algorithm, since it reduces the number of dimensions in the high dimensional input space to two dimensions, the dimensions of the output grid. Once the codebook vectors are initialized, usually with random values, training begins. The training set of samples is presented to the self organizing map algorithm, and once all the samples have been selected, this process is repeated for t training steps. One complete round of training, when all of the samples have been selected once, is designated as an epoch. The number of training steps is an integer multiple of the number of epochs. For training and visualization purposes, the sample vectors are assigned to the most similar prototype vector, or best-matching unit (BMU).

Kohonen (2001) describes the self organizing map as a non linear, ordered, smooth mapping of high dimensional input data manifolds into the elements of a regular low dimensional array where the mapping is implemented as follows:

Assume that the set of input variables is defined as a real vector

$$x = [\ a_1, a_2, \ldots, a_n\ ]^T \in \Re^n \qquad (4.6)$$

Each element in the self organizing map array is associated with a parameter real vector

$$m_i = [\ \mu_{i1}, \mu_{i2},\ldots,\mu_{in}\ ]^T \in \Re^n \qquad (4.7)$$

which is named a model.

A general distance measure between x and m_i is denoted $d(x, m_i)$. The image of an input vector x on the self organizing map array is defined as the array element m_c that matches best with x with the following index:

$$c \ = \ \arg\min_t \{\ d(x, m_i)\ \} \qquad (4.8)$$

Self organizing maps differ from the vector quantization since the m_i is defined in such a way that the mapping is ordered and descriptive of the distribution of x. Kohonen (1995) also emphasizes that the models m_i need not be vectoral variables, it will suffice if the distance measure $d(x, m_i)$ is defined over all occurring x items and a sufficiently large set of models m_i.

The self organizing map defines a mapping from the input data space onto a two dimensional array of nodes. The parametric model vector, $m_i = [\ \mu_{i1}, \mu_{i2},\ldots,\mu_{in}\ ]^T \in \Re^n$, must be initialized before recursive processing can begin. Random numbers are selected for the components of the m_i to demonstrate that starting from an arbitrary initial state, in the long run, the m_i will attain two-dimensionally ordered values. This is the basic effect of the self organization.

In the simplest case, an input vector, $x = [\ a_1, a_2,\ldots,a_n\ ]^T \in \Re^n$ is connected to all neurons in parallel via variable scalar weights μ_{ij}, which in general are different for different neurons. The input x is compared with all the m_i and the location of the best match in some metric is defined as the location of the response. The exact magnitude of the response need not be determined, the input is simply mapped onto this location, like a set of decoders.

Let $x \in \Re^n$ be a stochastic data vector. The self organizing map can be seen as a "non linear projection" of the probability density function p(x) of the high dimensional input data vector x onto the two dimensional display.

Vector x may be compared with all the m_i in any metric, in many practical applications, the smallest of the Euclidean distances $\|x-m_i\|$ can be made to define the best matching node, signified by the subscript c:

$$c = \arg\min_i \{\|x-m_i\|\} \qquad (4.9)$$

which means the same as

$$\|x-m_c\|=\min_i\{\|x-m_i\|\} \qquad (4.10)$$

During learning or the process in which the non linear projections is formed, those nodes that are topographically close in the array up to a certain geometric distance will activate each other to learn something from the same input x. This will result in a local relaxation or smoothing effect on the weight vectors of neurons in this neighbourhood, which in continued learning leads to global ordering. Consider the eventual convergence limits of the following learning process, whereupon the initial values of the $m_i(0)$ can be arbitrary,

$$m_i\ (t+1) = m_i\ (t) + h_{ci}(t)\ [x(t) - m_i\ (t)] \qquad (4.11)$$

where

t = 0, 1, 2,…is an integer, the discrete time coordinate.

In the relaxation process the function $h_{ci}(t)$ has a very central role, it acts as the so called neighbourhood function, a smoothing kernel defined over the lattice points.

The neighbourhood function can be written as

$$h_{ci}(t) = h(\| r_c-r_i\|,t) \qquad (4.12)$$

where $r_c \in \Re^2$ and $r_i \in \Re^2$ are the location vectors of nodes c and i respectively, in the array.

With increasing $\| r_c - r_i \|$, $h_{ci}(t) \to 0$. The average width and form of h_{ci} define the stiffness of the elastic surface to be fitted to the data points.

The basic principles of the self organizing map seem simple, the process behaviour, especially relating to the above more complex input representations has been difficult to describe in mathematical terms. The first approach discusses the process in its simplest form, but it seems that similar results are obtainable with more complex systems. The self organizing ability will be justified analytically using a very simple system model. The reasons for the self ordering phenomena are actually subtle and have been proven only in the simplest cases. In this discussion a basic Markov process is explained to help understand the nature of the process and is restricted to a one dimensional linear open ended array of functional units to each of which a scalar values input signal, ξ , is connected.

Let the units be numbered 1, 2, ... , j. Each unit i has a single scalar input weight or reference value μ_i whereby the similarity of between ξ and μ_i is defined by the absolute value of their difference $| \xi - \mu_i|$.

The best match is defined as follows:

$$| \xi - \mu_c| = \min_c\{| \xi - \mu_i|\} \qquad (4.13)$$

The set of units N_c selected for the updating is defined as follows:

$$N_c = \{\max (1, c-1), c, \min (j, c+1)\} \qquad (4.14)$$

In other words, unit i has the neighbours i-1 and i+1, except at the end points of the arrays, where the neighbour of unit 1 is 2, and the neighbour of unit j is j-1. Then N_c is simply the set of units consisting of unit c and its immediate neighbours.

The neighbourhood kernel determines the influence on the neighbouring model vectors. The learning process gradually shifts from an initial rough learning phase with a large influence area and fast-changing prototype vectors to a fine-tuning phase with small neighbourhood radius and prototype vectors that adapt slowly to the samples. The self organizing map algorithm contains elements of competitive learning and cooperative learning. Competitive learning is covered by selection of the best-matching unit, which is updated to the largest extent. Cooperative learning updates the most similar model vector and also moves its closest neighbours in the direction of the sample, creating similar areas on the map. After training is completed, the self organizing map has folded onto the training data, where neighbouring units usually have similar values.

Each prototype is also associated with a Voronoi region in input space, which is defined as follows:

$$V_k = \{x : \| x - m_k \| < \| x - m_j \| \ \forall j \neq k\} \qquad (4.15)$$

These regions reflect the area in input space for which a prototype is a best-matching unit. Input space is thus divided into these non-overlapping Voronoi regions. If a unit's Voronoi region does not contain any sample vectors, it is named an interpolating unit, which occurs if neighbouring regions on the lattice contain distant prototypes in output space.

The Kohonen self organizing map algorithm requires a kernel function

$$K^s(j,n)$$

where

$$K^s(j,j) \;=\; 1$$

and

$K^s(j,n)$ is a nonincreasing function of the distance between seeds *j* and *n* in the grid space.

For seeds that are far apart in the grid space the kernel function is usually equal to zero, that is,

$$K^s(j,n) \;=\; 0$$

As each training case is processed, all the seeds are updated as

$$C_n^{s+1} = C_n^s(1 - K^s(j,n)L^s + X_i K^s(j,n)L^s \qquad (4.15)$$

with the kernel function changing during training as indicated by the superscripts.

The neighbourhood of a given seed is the set of seeds for which the kernel function is greater than zero, that is,

$$K^s(j,n) \;>\; 0$$

To avoid poor results, it is usually recommended to start with a large neighbourhood and to let the neighbourhood gradually shrink during training.

If $K^s(j,n) \;=\; 0$ for $j \neq n$,

then the self organizing map update formula reduces to the formula for Kohonen vector quantization.

If the neighbourhood size (for example, the radius of the support of the kernel function) is zero, then the self organizing map algorithm degenerates into simple vector quantization.

Therefore, it is important not to let the neighbourhood size shrink all the way to zero during training if topological mapping is required. Consequently the choice of the final neighbourhood size is the most important tuning parameter for self organizing map training.

The learning rate $a(t)$ is also decreasing monotonically with time, and should end at zero when training is complete. Surprisingly, the results do not vary significantly for different choices of any of the functions and parameters above, thus the self organizing map is a very robust algorithm with regard to its configuration.

To achieve good topological ordering, it is advisable to specify a final neighbourhood size greater than one. Determining a good neighbourhood size usually requires trial and error.

For highly nonlinear data, use a Kohonen self organizing map, which by default behaves as follows:

- The initial seeds are randomly selected cases.
- The initial neighbourhood size is set to half the size of the self organizing map.
- The neighbourhood size is gradually reduced to zero during the first 1 000 training steps.
- Incremental training is used.
- The learning rate is initialized to 0.9 and linearly reduced to 0.02 during the first 1 000 training steps.

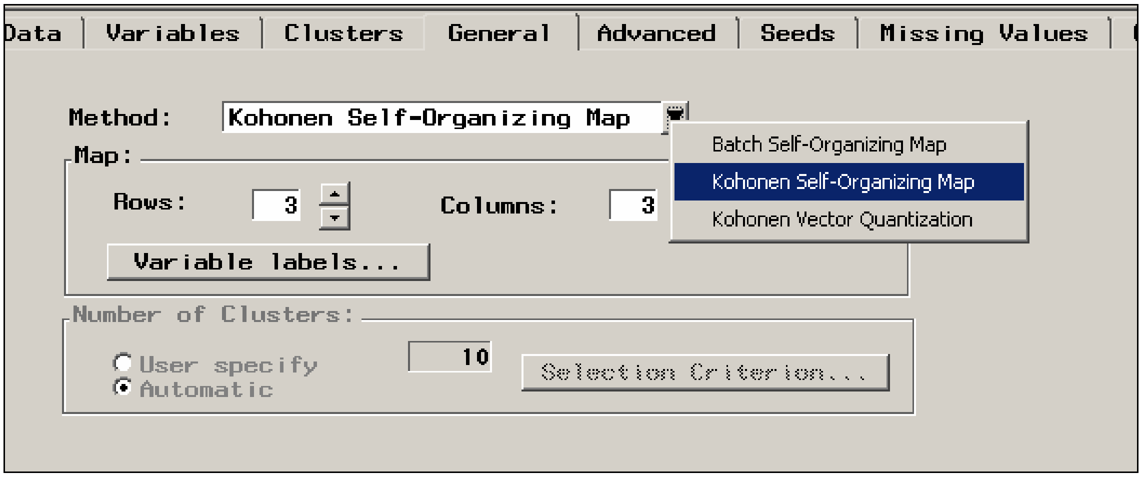
4.3.2 Analysis

In this section the Kohonen self organizing map technique is applied to the 10% sample data from the Republic of South Africa 2001 Census. In the sample there are 905 748 households and four attributes were selected to measure the dimension of poverty: access to basic to services. The analysis is conducted using SAS Enterprise miner's SOM/Kohonen node. The Kohonen self organizing map technique to measure the dimension "access to basic services" is illustrated using the following four attributes:

- access to water,
- energy source for cooking,
- toilet facilities, and
- refuse removal.

The membership function proposed by Cheli and Lemmi (1995) is applied to the above four attributes. The data set used in this calculation is the same that was used in section 4.2.2 in the Kohonen vector quantization analysis.

Figure 4.3.2: The SOM/Kohonen node: Kohonen self organizing map: General tab



Method: Kohonen Self-Organizing Map

Map:

Rows: 3

Columns: 3

Number of Clusters:

○ User specify

● Automatic

10

In the SOM/Kohonen node general tab as shown in Figure 4.3.2, the Kohonen self organizing map is selected for the method. The number of rows and number of columns in the map need to be selected before the node can be run. There are no restrictions on the number of rows and the number of columns and the number of rows does not have to be the same as the number of columns. In this application the number of rows is set to three and the number of columns is set to three. The number of clusters is dimmed when the Kohonen self organizing map is selected. In this calculation the mapping is made onto a grid, where the number of rows and number of columns need to be determined before the node is run.

The SOM/Kohonen node is run for the Kohonen self organizing map analysis with the above mentioned settings and the following results are obtained:

- The Map Tab contains a topological mapping of all the input attributes to the clusters and a plot of the input means for all the attributes that are used in the analysis.

- The Variables Tab lists all the input attributes that are used in the Kohonen self organizing map analysis.

- The Distances Tab provides a graphical representation of the size of each cluster and the relationship among segments.

- The Profiles Tab provides a graphical representation of the categorical attributes and interval attributes for each segment.

- The Statistics tab displays information about each segment in a tabular format.

- The Output tab displays the output that is generated from running the underlying SAS/STAT DMVQ procedure.

Figure 4.3.3: SOM/Kohonen node: Map tab

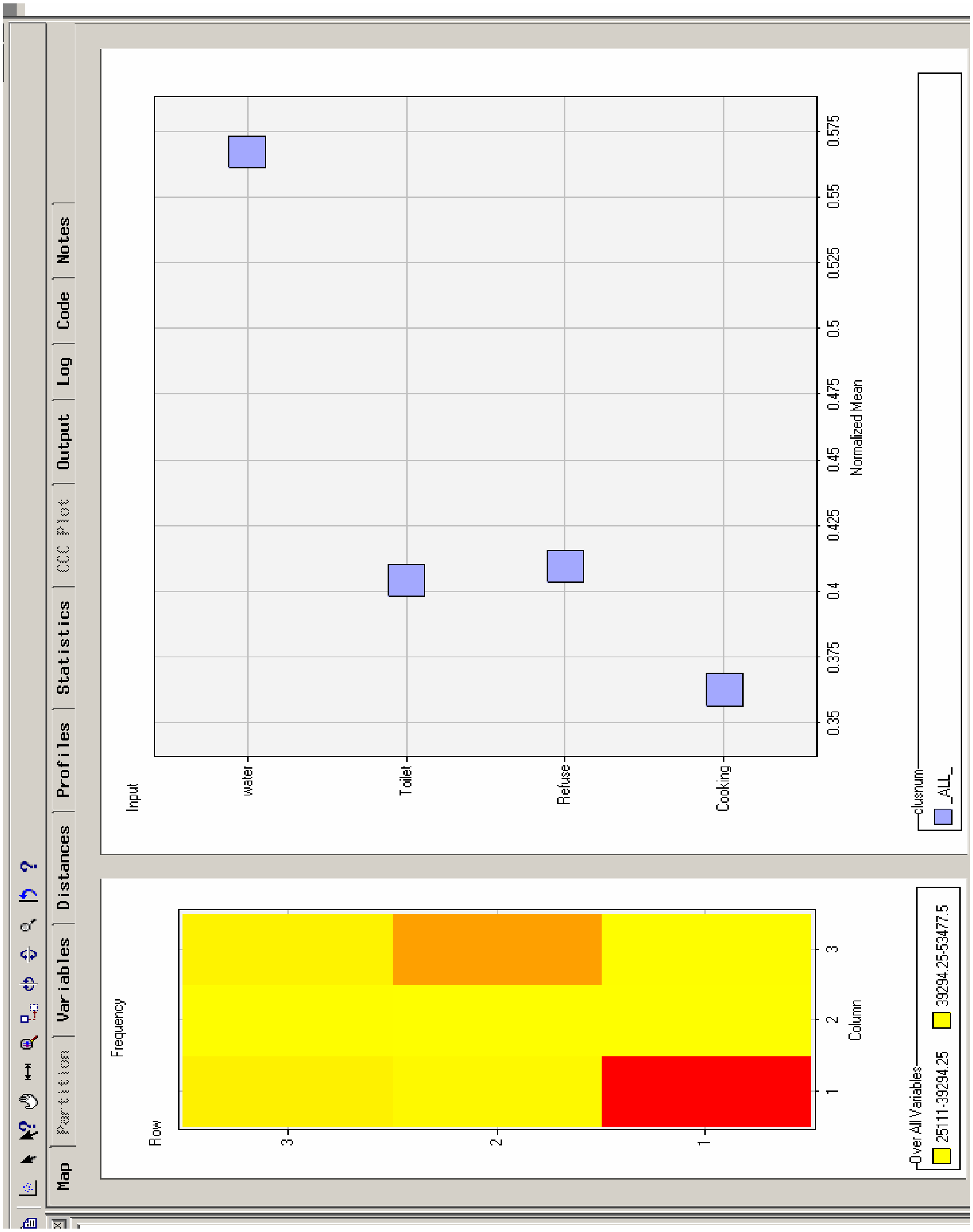
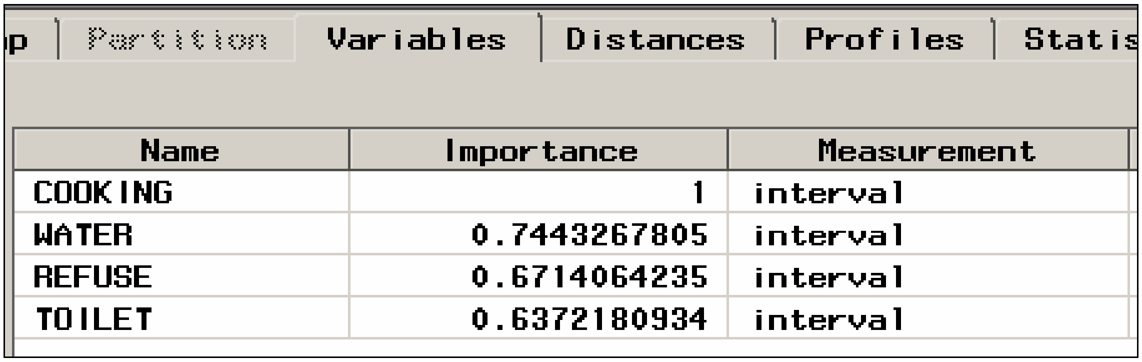


Figure 4.3.3 shows the Map tab of the SOM/Kohonen node results browser with the topological mapping on the left and the plot of the input means for the four attributes on the right. The row and column coordinates of the topological map in figure 4.3.3 correspond to the cluster numbers, for example, the coordinates for cluster 1 are Row 1, Column 1, and the coordinates for cluster number 9: Row 3, Column 3. The clusters in the map are colour coded by the frequency counts over all the input variables. The colours in the map legend correspond to the frequency count in the clusters. It can be clearly seen that cluster 1 has the highest frequency and cluster 6 has the second highest frequency.

The grid plot to the right of the tab displays a plot of the input means for the four attributes that are used in the analysis over all the clusters. The overall input means for each attribute are represented by the small squares in the plot. They are normalized to fall within a range of 0 to 1.

Figure 4.3.4: SOM/Kohonen node: Variables tab



Name	Importance	Measurement
COOKING	1	interval
WATER	0.7443267805	interval
REFUSE	0.6714064235	interval
TOILET	0.6372180934	interval

Figure 4.3.4 lists the four input attributes that were used in the SOM/Kohonen node to perform the Kohonen self organizing map analysis. For each attribute, an importance value is computed as a value between 0 and 1 to represent the relative importance of the given attribute to the formation of the clusters. Attributes that have the largest contribution to the cluster profile have importance values closer to 1.

In this analysis the attribute “energy source for cooking” has the highest importance value of 1. The other attributes also have fairly high importance values implying that they have also contributed to the cluster formation.

Table 4.3.1: SOM/Kohonen node: Statistics tab

	Segment	Frequency	water	Refuse	Cooking	Toilet	Distance
no deprivation	1	252043	0.000	0.001	0.001	0.002	0.002
very little deprivation	7	127488	0.473	0.002	0.002	0.011	0.473
little deprivation	4	76209	0.383	0.003	0.598	0.021	0.710
below average deprivation	5	25111	0.452	0.006	0.027	0.760	0.885
average deprivation	2	37236	0.345	0.824	0.021	0.098	0.899
above average deprivation	8	51495	0.620	0.011	0.575	0.765	1.141
extreme deprivation	3	46063	0.665	0.832	0.771	0.143	1.323
very extreme deprivation	9	121396	0.678	0.839	0.286	0.727	1.332
maximum deprivation	6	168707	0.748	0.852	0.926	0.810	1.673

Table 4.3.1 displays information about each cluster obtained from the statistics tab of the result browser in a tabular format. The cluster numbers and frequency (number of households) of each cluster are given in columns two and three. For each cluster the mean of the input attribute is also given. The last column in table 4.3.1 is the Euclidean distance calculated from the cluster means of each attribute to the centre of origin. The clusters were then ranked where the cluster with the smallest Euclidean distance is categorized as the cluster with households that were the best off and the cluster with the largest Euclidean distance regarded as the cluster with households that are worst off in terms of deprivation of basic services.

Households that have a cluster mean of zero for any attribute experience zero deprivation in that attribute. The cluster means of all the attributes in cluster 1 are virtually zero, thus the cluster households are described in the first column of table 4.3.1 as experiencing zero deprivation. The maximum possible Euclidean distance measure is 2, when the cluster means for all the attributes are equal to one. Cluster 6 has a Euclidean distance measure of 1.673 and all its households are described as experiencing maximum deprivation. Table 4.3.1 shows the multidimensional measure of deprivation. Households in cluster 1 experience zero deprivation. Households in cluster 6 experience maximum deprivation; this is the union measure of poverty where the households experience deprivation in all attributes. The remaining seven clusters experience the union measure of poverty, deprivation in at least one attribute. The self organizing map technique splits the union measure of poverty into seven grades or shades.

Figure 4.3.5: SOM Node: Distance Tab

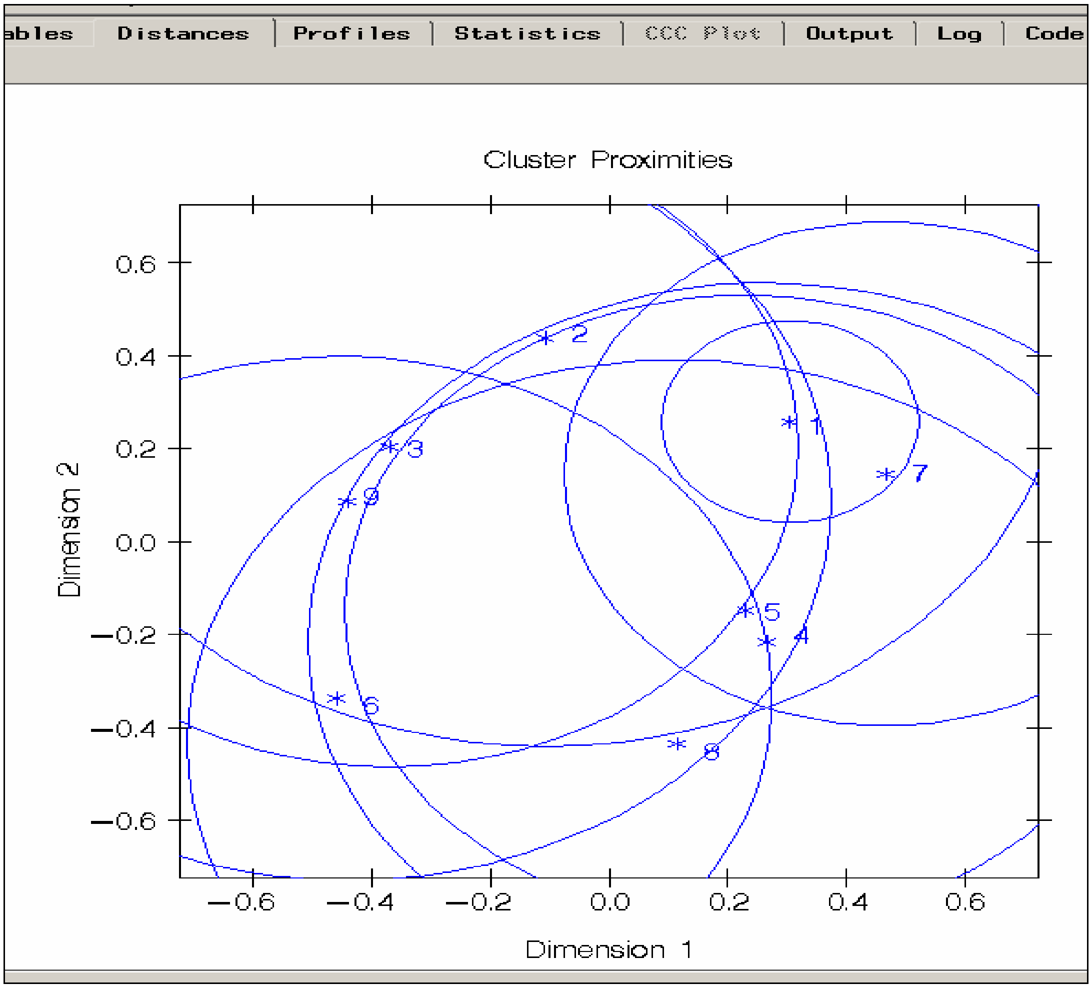
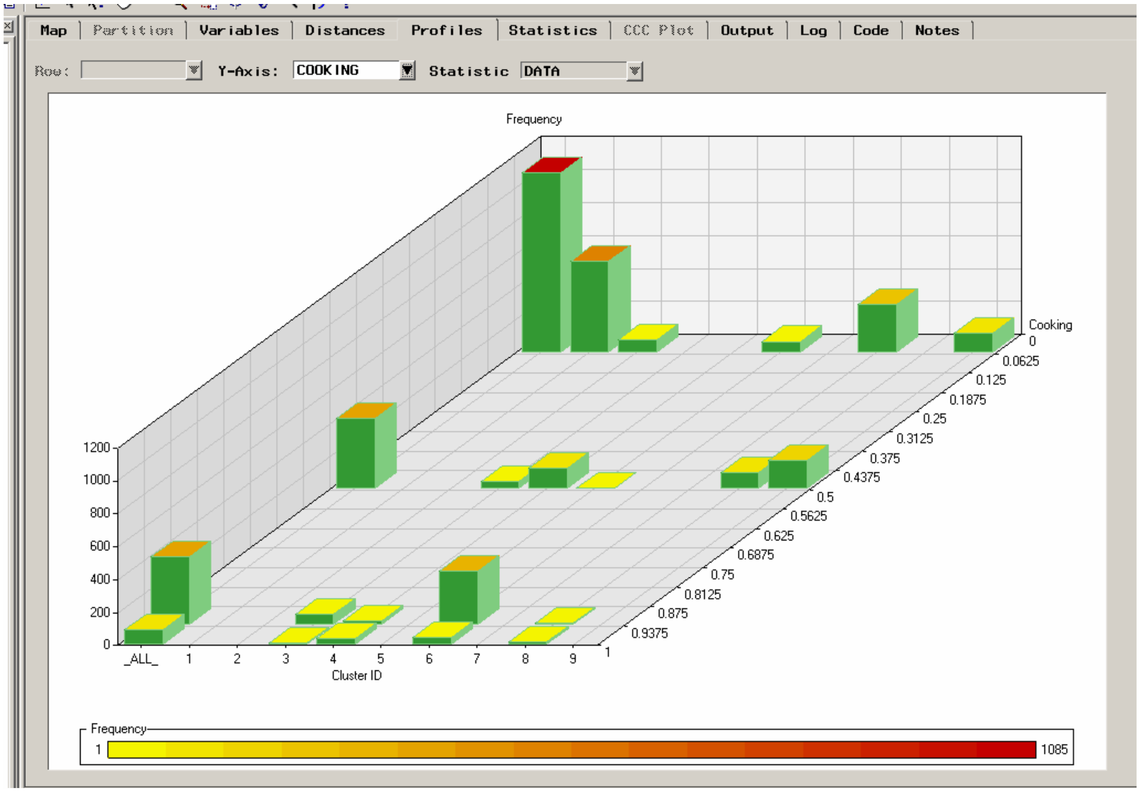


Figure 4.3.5 shows the graphical representation of the size of each cluster and the relationship among the clusters. The axis in figure 4.3.5 is determined from multidimensional scaling analysis. The cluster centres are represented by asterisks and the circles represent the cluster radii. If there is only one household in a cluster then this

household is displayed as an asterisk. The radius of each cluster depends on the most distant case in that cluster. Cluster 1 has the highest frequency of households, 252 043 households and the smallest circle. The small radius of cluster 1 suggests that the distance between the households within the cluster is small. The radii in figure 4.3.5 might appear to indicate that the clusters overlap, but the analysis assigns each household to only one cluster.

Figure 4.3.6: SOM/Kohonen node: Profile tab for cooking



Row:
Y-Axis: COOKING
Statistic DATA

Figure 4.3.6 displays a three dimensional bar chart for the interval input attributes “energy source for cooking”. The three dimensional bar chart displays the interval input attribute, cooking, on the Y-axis, the cluster number on the X-axis and the frequency within each cluster on the Z-axis. The frequencies are low since a sample of the training data set is used to construct the bar chart.

It can be seen that households in cluster 1 experience zero deprivation, while households in cluster 6 experience the most deprivation with respect to “energy source for cooking”. The bars for clusters 3, 4 and 8 show that they comprise some households that experience total deprivation with respect to “cooking” and other households that experience some deprivation. There are no households in these clusters that experience no deprivation with respect to “cooking”.

Figure 4.3.7 SOM/Kohonen node: Map tab

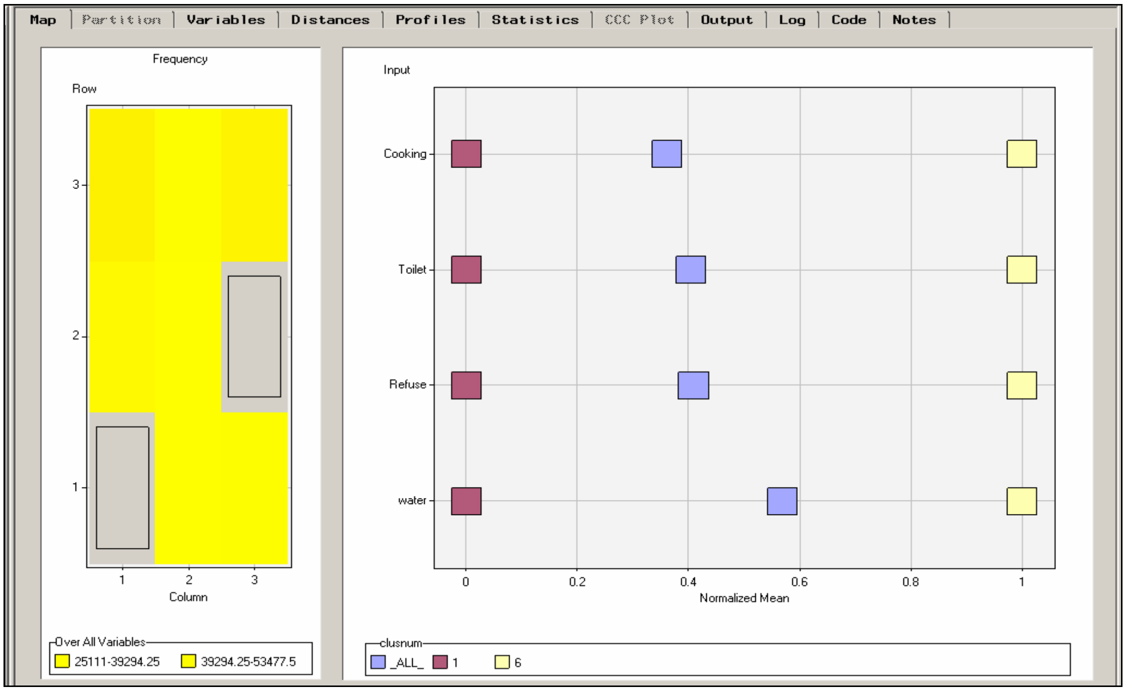
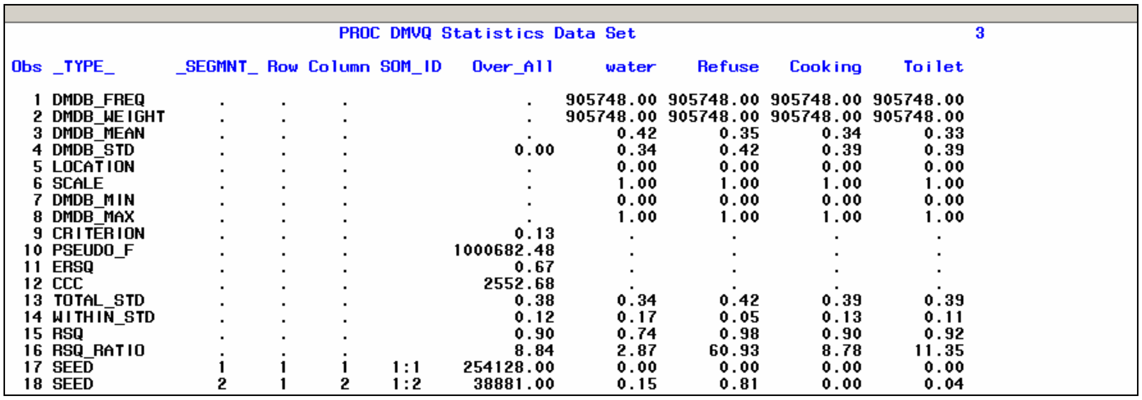


Figure 4.3.7 is the Map Tab results for the Kohonen self organizing map, comparing the input means for cluster 1 and cluster 6 with the overall input means. In the topological mapping on the left of figure 4.3.7 segment 1 (row 1, column 1) and segment 6 (row 2, column 3) are highlighted.

The input plot on the right in figure 4.3.7 shows the input means of cluster 1, cluster 6 and the overall input means. The plot ranks the attributes based on how spread out the input means are for the selected clusters relative to the overall input means. The input with the greatest spread, attribute “energy source for cooking”, is listed first and the input with the smallest spread, attribute “access to water”, is listed last.

For cluster 1 the input means for all the attributes are shown as zero. The input means are normalized to have a range of zero to one. This means that all the households in cluster 1 are best off with respect to deprivation of basic services for the four attributes. For cluster 6 the input means for all the attributes are 1. This means that all the households in cluster 6 are the worst off with respect to deprivation of basic services for the four attributes.

Figure 4.3.8: SOM/Kohonen node: Output statistics



```
                               PROC DMVQ Statistics Data Set                               3

Obs _TYPE_       _SEGMNT_ Row Column SOM_ID    Over_All     water    Refuse   Cooking    Toilet

  1 DMDB_FREQ           .   .      .                  . 905748.00 905748.00 905748.00 905748.00
  2 DMDB_WEIGHT         .   .      .                  . 905748.00 905748.00 905748.00 905748.00
  3 DMDB_MEAN           .   .      .                  .      0.42      0.35      0.34      0.33
  4 DMDB_STD            .   .      .               0.00      0.34      0.42      0.39      0.39
  5 LOCATION            .   .      .                  .      0.00      0.00      0.00      0.00
  6 SCALE               .   .      .                  .      1.00      1.00      1.00      1.00
  7 DMDB_MIN            .   .      .                  .      0.00      0.00      0.00      0.00
  8 DMDB_MAX            .   .      .                  .      1.00      1.00      1.00      1.00
  9 CRITERION           .   .      .               0.13         .         .         .         .
 10 PSEUDO_F            .   .      .         1000682.48         .         .         .         .
 11 ERSQ                .   .      .               0.67         .         .         .         .
 12 CCC                 .   .      .            2552.68         .         .         .         .
 13 TOTAL_STD           .   .      .               0.38      0.34      0.42      0.39      0.39
 14 WITHIN_STD          .   .      .               0.12      0.17      0.05      0.13      0.11
 15 RSQ                 .   .      .               0.90      0.74      0.98      0.90      0.92
 16 RSQ_RATIO           .   .      .               8.84      2.87     60.93      8.78     11.35
 17 SEED                1   1      1    1:1   254128.00      0.00      0.00      0.00      0.00
 18 SEED                2   1      2    1:2    38881.00      0.15      0.81      0.00      0.04
```

Figure 4.3.8 displays the output obtained after running the SAS DMVQ procedure. A table of the following statistics for each attribute is created:

- Total standard deviation
- Pooled standard deviation
- R square
- R square Ratio
- Pseudo f statistic

In this analysis the overall R Square is 0.90 with a pseudo F statistics value of 1 000 682.

4.4 BATCH SELF-ORGANIZING MAPS

As in the case of the k-means clustering, self organization can also be performed as a deterministic procedure. A deterministic self organizing map has been proposed by Kohonen (2001) as the Batch map. In this procedure each map node is mapped to a weighted average of the fixed data points, based on the current winner assignment. This important learning rule is named "Batch map", which is based on fixed point iteration, and is significantly faster in terms of computation time.

4.4.1 Methodology

The Batch map principle is use to define learning as a succession of certain generalized conditional averages over subsets of selected strings. These averages over the strings are computed as generalized medians of the strings.

Let

S be a fundamental set of some items $x(i)$

and

$d[x(i), x(j)]$ be some distance measure between $x(i), x(j) \in$ S.

The set median m over S shall minimize the expression

$$D = \sum_{x(i) \in S} d[x(i), m] \qquad (4.16)$$

The reason for naming m the median is that it is relatively easy to show that the usual median of real numbers is defined by equation (4.16) whenever the distance measure satisfied the following:

$$d[x(i), x(j)] = |x(i) - x(j)| \qquad (4.17)$$

In the case above it was assumed that m belongs to the fundamental set S, however it is possible to find a hypothetical item m such that D attains its absolute minimum value. In contrast to the set median the term generalized median is used to denote the value of *m* that gives the absolute minimum value for *D* as it was shown earlier that the convergence limits during the learning process were

$$m_i\ (t+1) = m_i\ (t) + h_{ci}(t)\ [x(t) - m_i\ (t)] \qquad (4.18)$$

It is now useful to understand what the convergence limits m_i^* represent. Assume that the convergence to some ordered state is true, then the expected values of

$m_i\ (t+1)$ and $m_i\ (t)$ must be equal.

In other words in the stationery state

$$E\{h_{ci}(x-m_i^*)\} = 0 \qquad \text{for all values of } i$$

In the simplest case $h_{ci}(t)$ was defined as follows:

$$h_{ci} = \begin{cases} 1 & \text{if i belongs to some topo log ical neighbourhood set } N_c \\ 0 & \text{otherwise} \end{cases}$$

The convergence limit m_i^* can be defined as follows:

$$m_i^* = \frac{\int_{Vi} xp(x)d(x)}{\int_{Vi} p(x)d(x)} \qquad (4.19)$$

where

> V_i is the set of those values in the integrands that are able to update vector m_i, in other words the winner node c for each $x \in V_i$ must belong to the neighbourhood set N_i of cell *i.*

The iterative process in which a number of samples of x is first classified into the respective V_i regions and the updating of the m_i^* is made iteratively as defined by equation (4.19), can be expressed in the following steps (Kohonen 2001).

Firstly the training samples are assumed to be available when the learning begins. The learning steps can be defined as follows:

Step 1: For the initial reference vectors, take the first K training samples, where K is the number of reference vectors.

Step 2: For each map unit i, collect a list of copies of all those training samples x whose nearest reference vector belongs to unit i.

Step 3: Take for each new reference vector the mean over the union of the lists in N_i.

Step 4: Repeat step 2 and step 3 until convergence or the maximum iterations.

If a general neighbourhood function h_{ji} is used and $\overline{x}_j$ is the mean of the x(t) in the Voronoi set V_j , then it shall be weighted by the number n_j of samples V_j and the neighbourhood function.

The following equation is obtained:

$$m_i^* = \frac{\sum_j n_j h_{ji} \overline{x}_j}{\sum_j n_j h_{ji}} \qquad (4.20)$$

where the sum over j is taken for all units of the self organizing map, or if h_{ji} is truncated over the neighbourhood set N_i in which it is defined.

For cases in which no weighting in the neighbourhood is used, equation (4.20) becomes

$$m_i^* = \frac{\sum_{j \in N_i} n_j \bar{x}_j}{\sum_{j \in N_i} n_j} \tag{4.21}$$

The above algorithm is very effective if the initial values of the reference vectors are already roughly ordered, even if they might not yet approximate the distribution of the samples.

It should also be noticed that the algorithm contains no learning rate parameter; therefore it has no convergence problems and yields stable asymptotic values for m_i other than the Kohonen self organizing map.

Better convergence may be achieved by specifying, in addition to Kohonen training, one or both of the Batch training options for Nadaraya-Watson smoothing or local-linear smoothing. Batch training often converges but sometimes does not. Any combination of the Kohonen, Nadaraya-Watson, and local-linear training may be specified but always applied in that order.

The self organizing map works by smoothing the seeds in a manner similar to kernel estimation methods, but the smoothing is done in neighbourhoods in the grid space rather than in the input space (Mulier and Cherkassky 1995). This can be seen in a Batch algorithm for self-organizing map which is similar to Forgy's algorithm for Batch k-means, but incorporates an extra smoothing process:

Read the data, assign each case to the nearest seed using the Euclidean distance measure, and at the same time track the mean and the number of cases for each cluster.

Do a nonparametric regression using $K^s(j,n)$ as a kernel function, with the grid points as inputs, the cluster means as target values, and the number of cases in each cluster as a case weight.

Replace each seed with the output of the nonparametric regression function evaluated at its grid point.

4.4.2 Analysis

In this section the Batch self organizing map technique is applied to the 10% sample data from the Republic of South Africa 2001 Census. In the sample there are 905 748 households and four attributes were selected to measure the dimension of poverty: access to basic services.

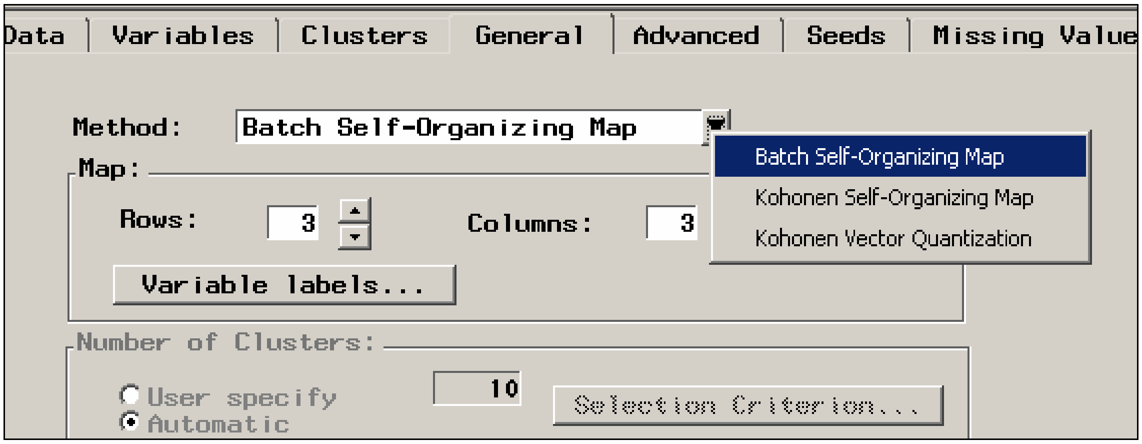
The four attributes used in the analysis are the following:

- access to water,
- energy source for cooking,
- toilet facilities and
- refuse removal.

The analysis is conducted using SAS Enterprise miner's SOM/Kohonen node with the membership function proposed by Cheli and Lemmi (1995) applied to the four attributes. The data set used in this calculation is the same that was used in section 4.3.2 in the Kohonen self organizing map analysis.

In the SOM/Kohonen node general tab as shown in figure 4.4.1, the method selected is the Batch self organizing map. The number of columns and the number of rows in the map need to be selected before the analysis can be run. There are no restrictions on the number of rows and the number of columns. The number of columns does not have to be the same as the number of rows. In this application the number of rows is set to three and the number of columns is set to three. The number of clusters is dimmed when the Batch self organizing map is selected. In this calculation the mapping is made onto a grid, where the number of rows and the number of columns need to be determined before the analysis is run.

Figure 4.4.1: The SOM/Kohonen node: General tab



Method: Batch Self-Organizing Map

Map:

Rows: 3 Columns: 3

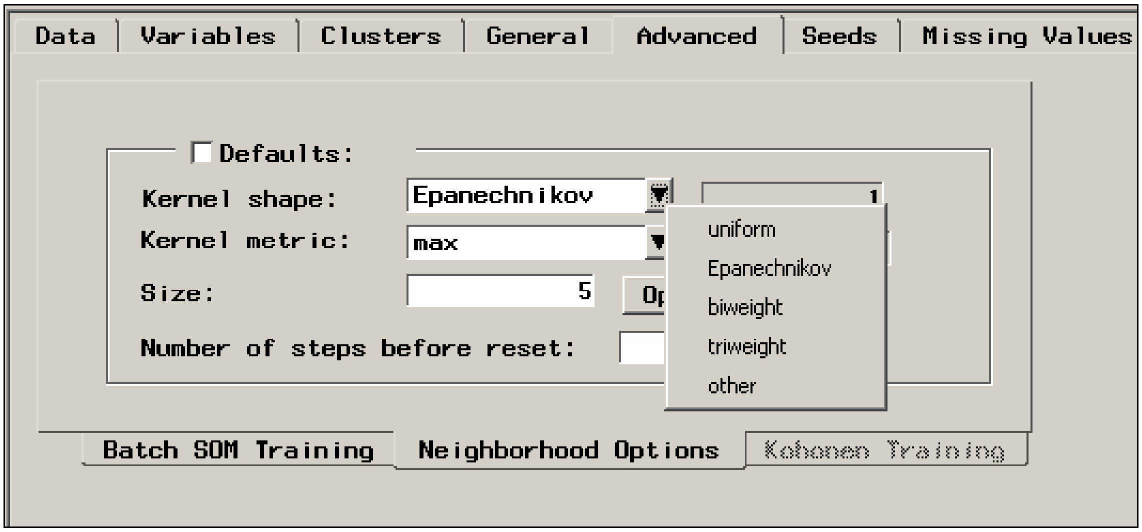
Number of Clusters:

○ User specify 10

● Automatic

Figure 4.4.2 shows the kernel shape options neighbourhood options sub tab of the advanced tab in the SOM/Kohonen node . In the kernel shape the default selection is Epanechnikov which has a value of 1. The uniform option has a value of 0. For the bi-weight the value is 2 and a value of 3 applies to the tri-weight. The other option allows the user to set a non negative value.

Figure 4.4.2: The SOM/Kohonen node: Advanced tab



☐ Defaults:

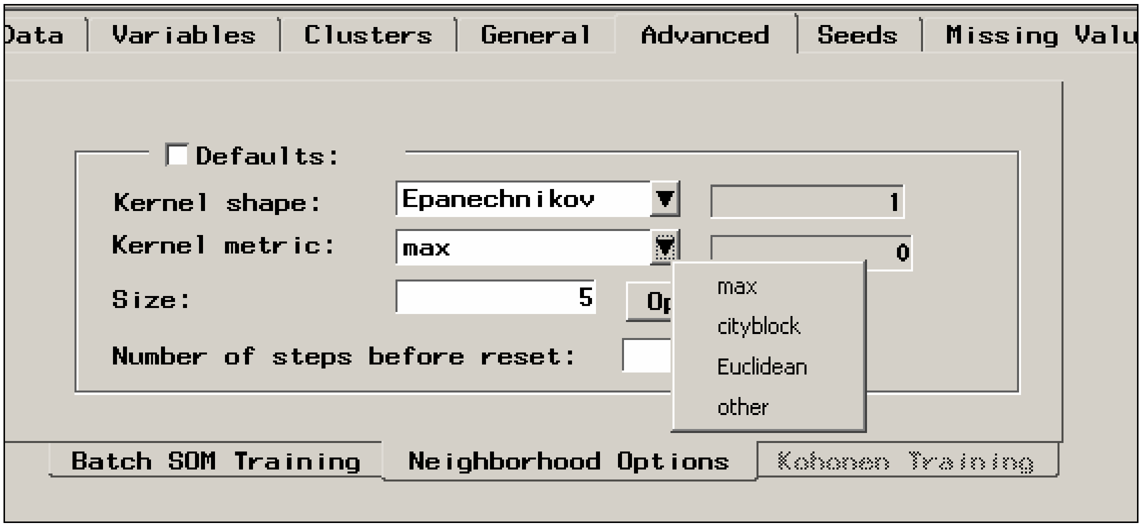
Kernel shape: Epanechnikov 1

Kernel metric: max

Size: 5

Number of steps before reset:

Figure 4.4.3: The SOM/Kohonen node: Advanced tab



□ Defaults:

Kernel shape: Epanechnikov ▼ 1

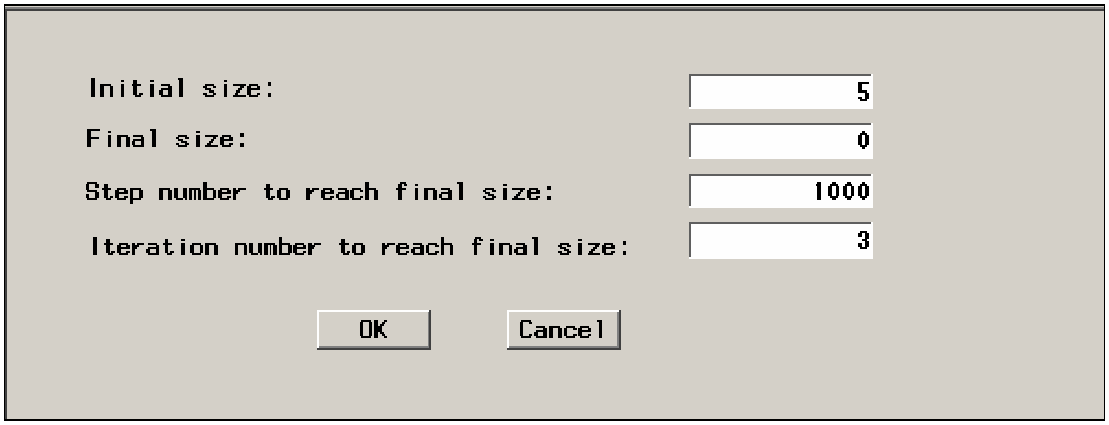
Kernel metric: max ▼ 0

Size: 5

Number of steps before reset:

Figure 4.4.3 shows the kernel metric options neighbourhood options sub tab of the advanced tab in the SOM/Kohonen node. The default selection for the kernel metric is max with a value of 0. The other metrics available are city block (value is 1), Euclidean (value is 2) and the other (a non negative value is supplied).

Figure 4.4.4: The SOM/Kohonen node: Neighbourhood size options



Initial size: 5

Final size: 0

Step number to reach final size: 1000

Iteration number to reach final size: 3

Figure 4.4.4 shows the size options of the neighbourhood options sub tab of the advanced tab in the SOM/Kohonen node. The neighbourhood size must be greater than or equal to zero.

Using the options button the initial size can be set using the following:

$$\text{Default size} = \text{Max}\left[5, \max\frac{(\text{Rows}, \text{Columns})}{2}\right].$$

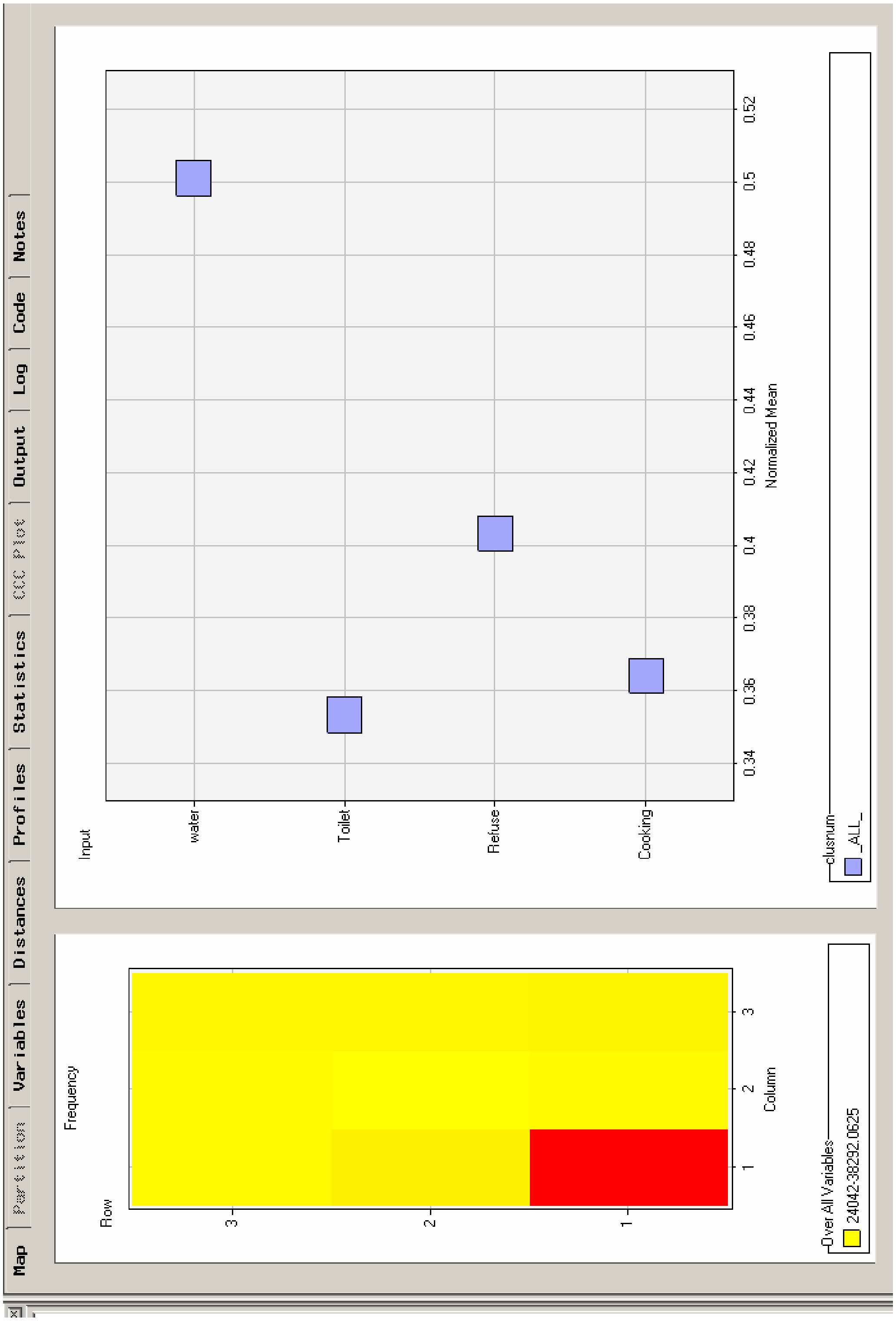
The final size is 0 and the number of steps to reach the final size is 1000, with the number of iterations to reach the final size set to 3.

The SOM/Kohonen node is run for the Batch self organizing map analysis with the above-mentioned settings and the following results are obtained:

- The Map Tab contains a topological mapping of all the input attributes to the clusters and a plot of the input means for all the attributes that were used in the analysis.
- The Variables Tab lists all the input attributes that are used in the Batch self organizing map analysis.
- The Statistics tab displays information about each segment in a tabular format.
- The Distance Tab provides a graphical representation of the categorical attributes and interval attributes for each segment.
- The Output Tab displays the output that is generated from running the underlying SAS DMVQ procedure.

Figure 4.4.5 shows the Map tab of the Batch self organizing map results with the topological mapping on the left and the plot of the input means for the four attributes on the right. The row and column coordinates in the topological map correspond to the cluster numbers, for example , the coordinates for cluster number 2 are row 1, column 2, and the coordinates for cluster number 7 are row 3, column 1.

Figure 4.4.5: SOM/Kohonen node: Map tab

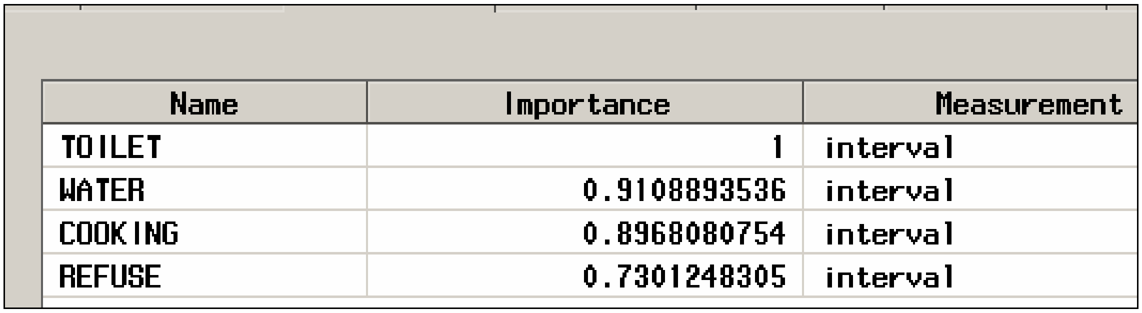


The clusters in the topological map are colour coded by the frequency counts over all the input attributes. The colours in the map legend correspond to the frequency count in the clusters. In this analysis, cluster number 1 has the highest frequency and this segment is the darkest coloured in the topological map.

The grid plot on the right of the topological map in figure 4.4.5 displays a plot of the input means for the four attributes that are used in the analysis over all the clusters. The overall input means for each attribute are represented by the small squares in the plot. All the input means are normalized to fall within a range of 0 to 1.

The attributes in the grid plot are arranged from the attribute with the largest input means on the top. In this case the attribute "access to water" has the highest normalized input mean and is listed first. The attribute "energy source for cooking" has the smallest normalized input mean and is listed last.

Figure 4.4.6: SOM/Kohonen node: Variables tab



Name	Importance	Measurement
TOILET	1	interval
WATER	0.9108893536	interval
COOKING	0.8968080754	interval
REFUSE	0.7301248305	interval

In figure 4.4.6 the four input attributes that were used in the SOM/Kohonen node to perform the Batch self organizing map analysis are listed. For each attribute, an importance value is computed as a value between 0 and 1 to represent the relative importance of the given attribute to the formation of the clusters.

Attributes that have the largest contribution to the cluster profile have importance values closer to 1. In this analysis the attribute "toilet facilities" has the highest importance value of 1. The attributes “access to water” and “energy source for cooking” has importance values very close to 1, suggesting that they have also contributed to the cluster formation.

Table 4.4.1: SOM/Kohonen node: Statistics tab

	Segment	Frequency	water	Refuse	Cooking	Toilet	Distance
no deprivation	1	252 043	0.00	0.00	0.00	0.00	0.00
very little deprivation	4	127 488	0.47	0.00	0.00	0.01	0.47
little deprivation	7	76 209	0.38	0.00	0.60	0.02	0.71
below average deprivation	5	24 042	0.47	0.00	0.00	0.76	0.90
average deprivation	2	53 562	0.44	0.83	0.15	0.11	0.95
above average deprivation	8	52 564	0.61	0.01	0.57	0.77	1.13
extreme deprivation	3	114 838	0.66	0.84	0.28	0.72	1.32
very extreme deprivation	6	102 569	0.65	0.83	0.92	0.50	1.49
maximum deprivation	9	102 433	0.84	0.87	0.90	0.93	1.77

Table 4.4.1 displays information about each cluster obtained from the statistics tab of the results browser in a tabular format. The segment number and the frequency (number of households) of each cluster are given in columns two and three. For each segment the mean of the input attribute is also given. The last column in table 4.4.1 is the Euclidean distance measure calculated from the segment means of each attribute to the centre of origin. The segments were then ranked according to the Euclidean distance. The segment with the smallest Euclidean distance is categorized as the segment with households that were the best off and the cluster with the largest Euclidean distance regarded as the segment with households that are worst off in terms of deprivation of basic services.

Households that have a segment mean of zero for any attribute experience zero deprivation in that attribute. The segment means of all the attributes in segment 1 are very close to zero. In table 4.4.1 the first column describes the segments and segment 1 is described as households experiencing zero deprivation. The maximum possible Euclidean distance measure is 2, (i.e. when the segment means for all the attributes are equal to one), segment 9 has an Euclidean distance measure of 1.771 and all its households are described as experiencing maximum deprivation in basic services. Table 4.4.1 shows the multidimensional measure of deprivation from households experiencing no deprivation to households experiencing maximum deprivation. There are 252 043 households in segment 1 that experience no deprivation of basic services. Segment 9 has 102 433 households that experience maximum deprivation of basic services, this can be described as the union measure of poverty where the households experience deprivation in all attributes. The middle seven segments experience the union measure of poverty, i.e. deprivation in at least one attribute. Segments in the first column of the grid experience less deprivation than segments in the last column.

Figure 4.4.7: SOM/Kohonen node: Distance tab

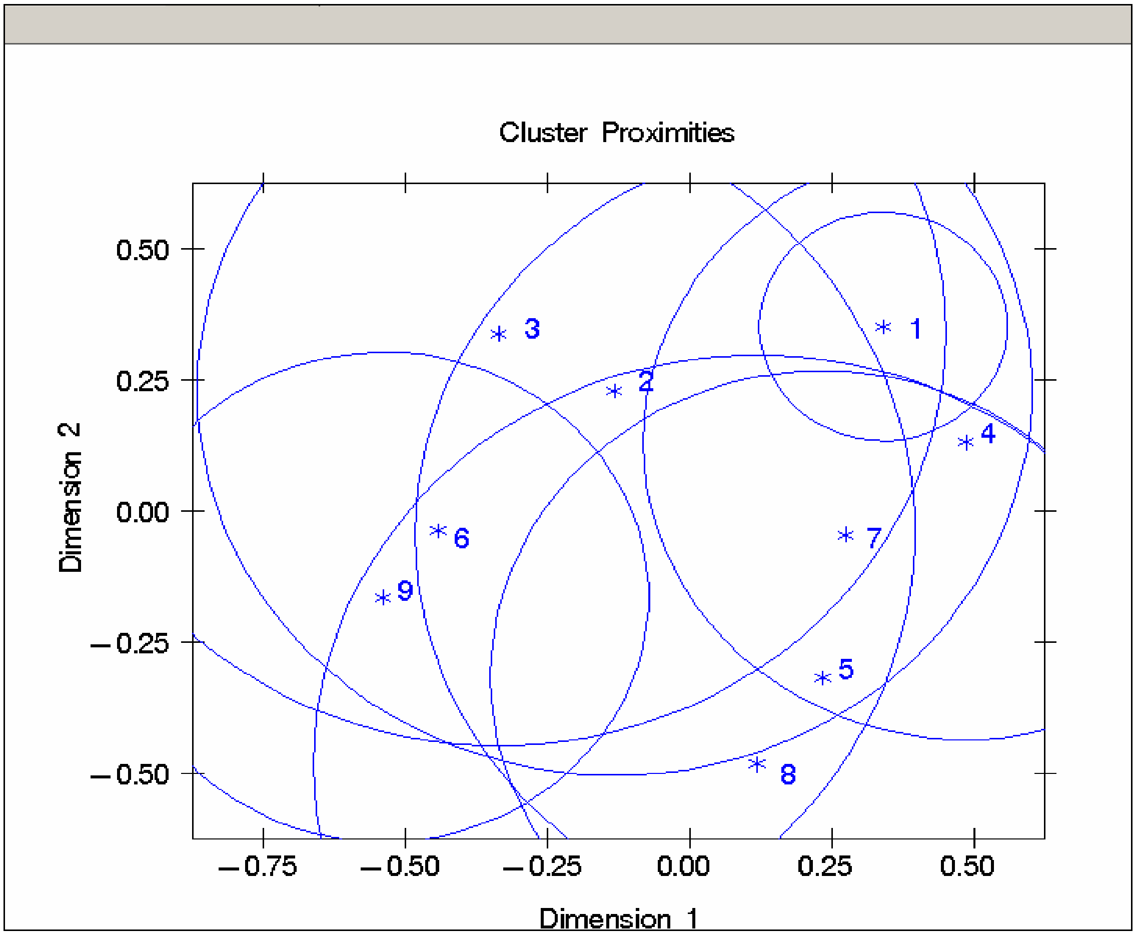


Figure 4.4.7 shows the graphical representation of the size of each segment and the relationship among the segments. The axis in figure 4.4.7 is determined from multi-dimensional scaling analysis. The segment centres are represented by asterisks and the circles represent the cluster radii. If there is only one household in a segment then this household is displayed as an asterisk. The radius of each segment is dependent on the most distant case in that segment. Segment 1 has the highest frequency of households, 252 043 households and the smallest circle. This suggests that the distance between households in segment 1 is small. The radii in figure 4.4.7 might appear to indicate that the segments overlap but the self organizing map algorithm assigns each household to only one segment.

Figure 4.4.8: SOM/Kohonen node: Map Tab

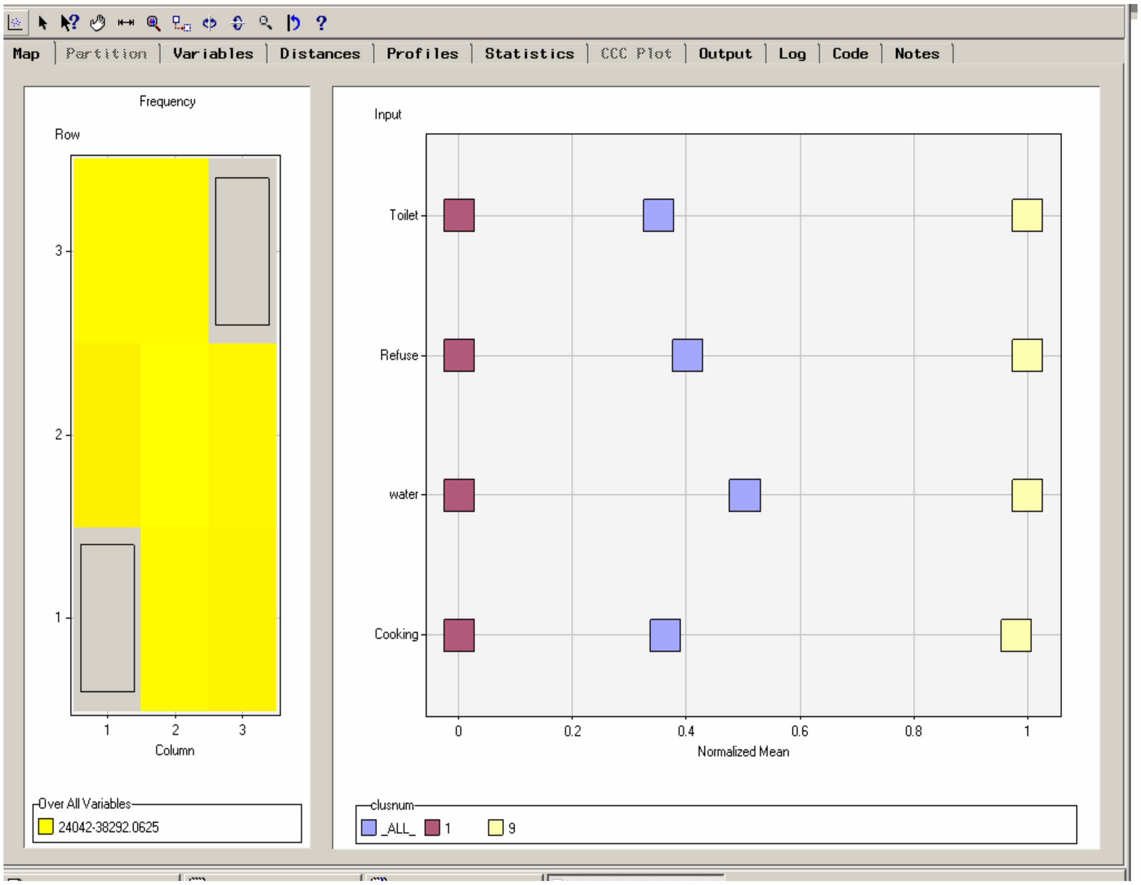


Figure 4.4.8 is the Map tab results for the Batch self organizing map, comparing the input means for segment 1 and segment 9 with the overall input means. In the topological mapping on the left of figure 4.4.8 segment 1 (row 1, column 1) and segment 9 (row 3, column 3) are highlighted.

The input plot on the right in figure 4.4.8 shows the input means of segment 1, segment 9 and the overall input means. The plot ranks the attributes based on how spread out the input means are for the selected segments relative to the overall input means. The input means with the greatest spread, attribute "toilet facilities" is listed first and the input with the smallest spread, attribute "energy source for cooking", is listed last.

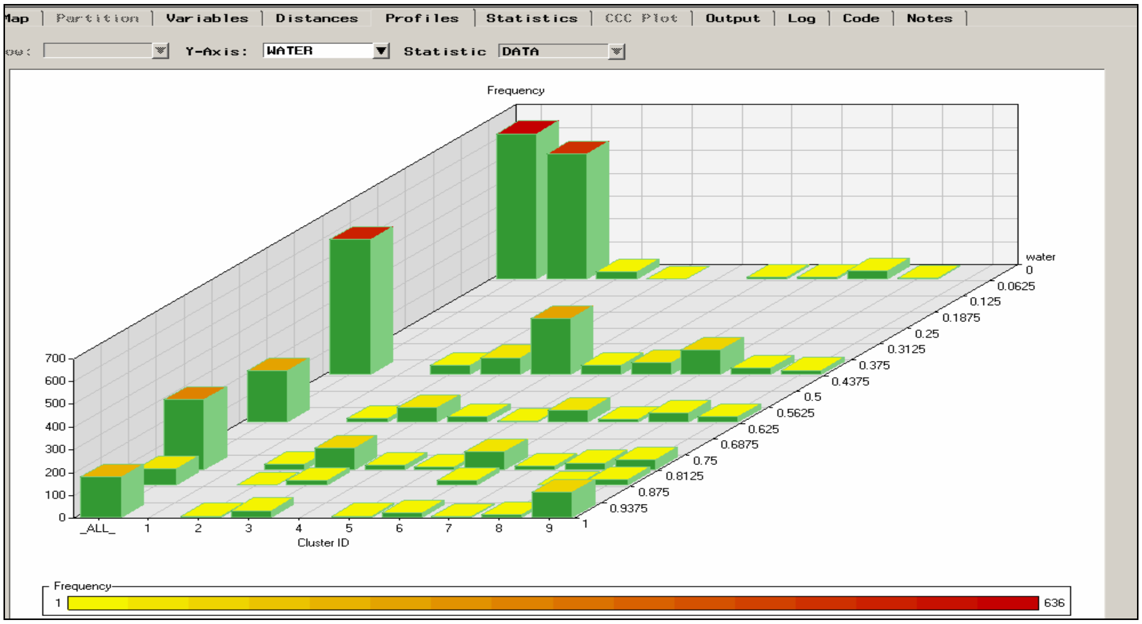
For segment 1, the input means for all the attributes are shown as zero. The input means are normalized to have a range of zero to one. This means that all the households in segment 1 are best off with respect to deprivation of basic services for the four attributes.

For segment 9 the input means for all attributes are 1. This means that all the households in segment 9 are worse off with respect to deprivation of basic services for the four attributes.

Figure 4.4.9 displays a three dimensional bar chart for the interval attribute "access to water". The three dimensional bar chart displays the interval input attribute "access to water" on the T-axis, the segment number on the X-axis and the frequency within each segment on the Z-axis. The frequencies are calculated on the training data set and not on the entire data set.

It can be seen that households in segment 1 experience zero deprivation, i.e. they all have flush toilets while the majority of households in segment 9 experience severe deprivation, that is, they have no toilet facilities.

Figure 4.4.9: SOM/Kohonen node: Profiles tab for water



ow:
Y-Axis: WATER
Statistic DATA

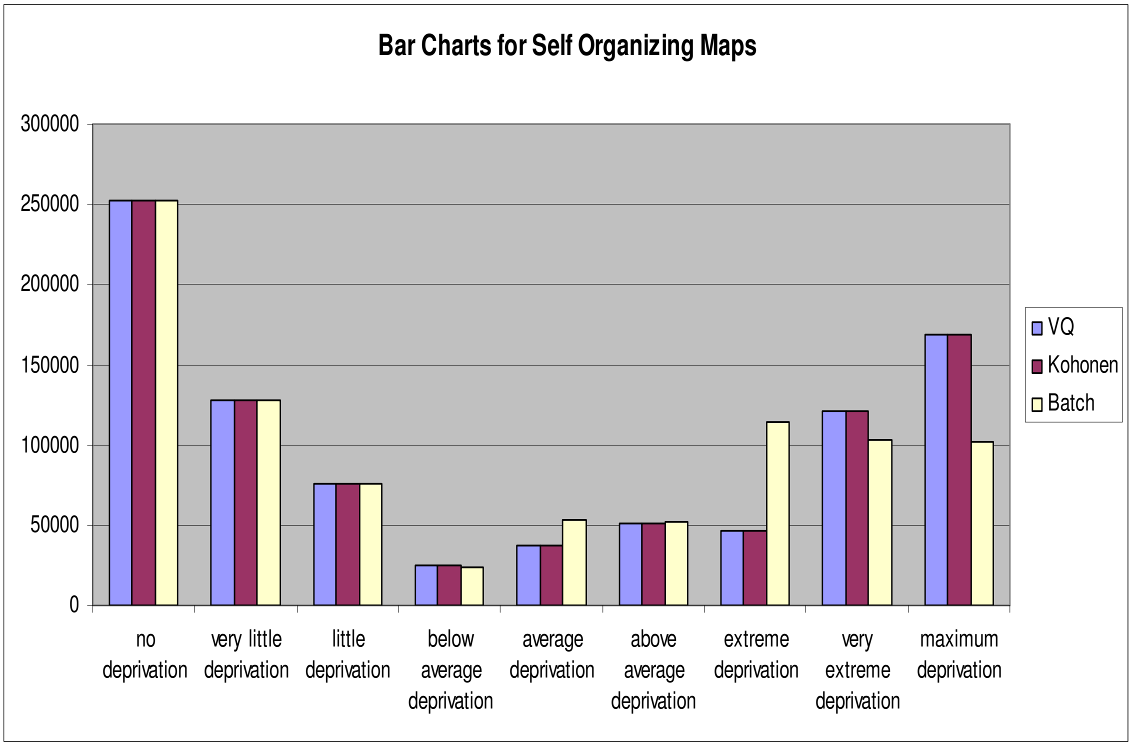
Segment 4 comprises of households that are in between, there are no households with flush toilets and no households with any toilet facilities. All the houses in this segment have alternative toilet facilities to flush toilets. This graphical representation clearly shows the multidimensional nature of poverty. There are many households that fall in between households that experience no deprivation and households that experience maximum deprivation.

4.5 CONCLUSION

In this chapter the Kohonen self organizing map node of SAS Enterprise miner was applied to the Republic of South Africa census sample data. For each method nine clusters or segments were created. Figure 4.5.1 shows the frequencies of each cluster/segment in a bar chart. The frequency of clusters created by the Kohonen vector quantization is the same as the Kohonen self organizing map. All three methods identified the same households as experiencing zero deprivation, very little deprivation, little deprivation and average deprivation. The differences emerge in the worst off clusters. The Batch self organizing map identifies fewer households in the maximum deprivation.

The final segments obtained for the Batch self organizing map are analysed further in this chapter. Each of the 905 748 households are categorised according to a segment created in the Batch self organizing map analysis. This section shows how the results can be used in poverty alleviation programs and policy decisions.

Figure 4.5.1 Bar chart for 9 clusters



In figure 4.5.2 the different shades of deprivation for the dimension “access to basic services” are plotted for each province. It can be seen that 62% of households in Western Cape experience no deprivation in basic services, while only 6% of households in Northern Province experience no deprivation in basic services.

The multidimensional measure of poverty created in this analysis can be clearly seen in figure 4.5.2. Poverty measurement can not be classified only as poor or not poor. For the provinces Mpumulanga, Eastern Cape, North West and Northern Province it can clearly be seen that many households experience the union definition of poverty. They experience deprivation in some attributes. This type of analysis allows for the monitoring of poverty among households.

Figure 4.5.2: Bar chart for provinces

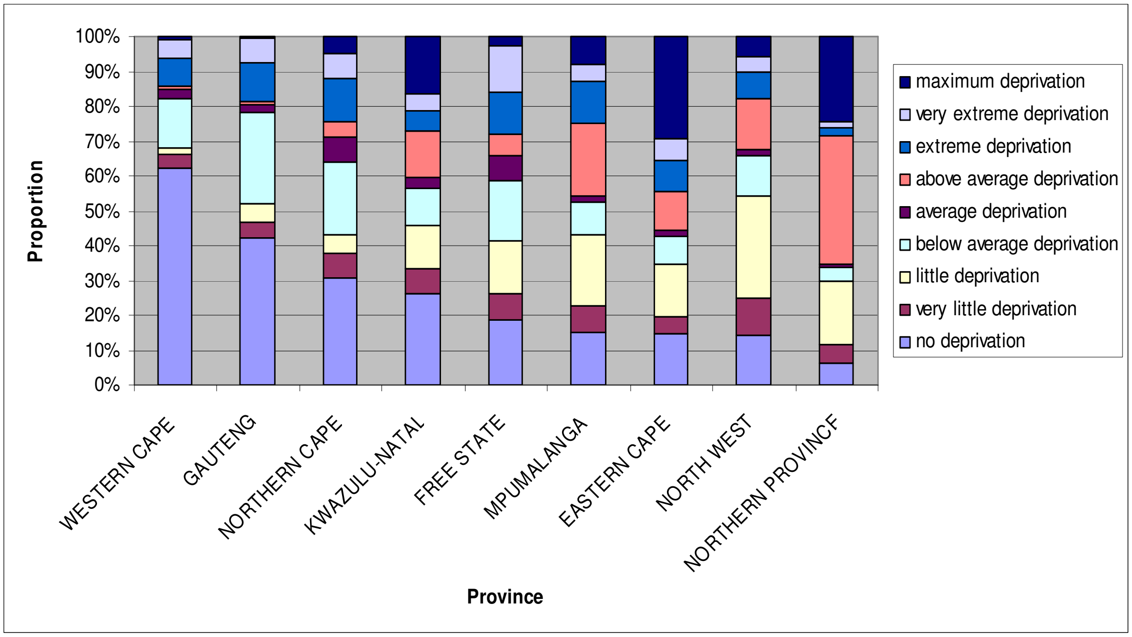


Table 4.5.1 shows the proportion of households within each province that experience deprivation. The provinces are ranked according to the highest proportion of households that experience no deprivation of basic services.

This result is useful to measure the impact of a poverty alleviation program on a province or municipality. The table is calculated before the relief measures and then calculated again after a period of time and the proportion in each category is compared. This monitoring tool can measure the effectiveness of the poverty relief measure.

Table 4.5.1: Deprivation across the 9 provinces

Province	WC	GP	NC	KZ	FS	MP	EC	NW	NP
no deprivation	0.62	0.42	0.31	0.26	0.19	0.15	0.14	0.14	0.06
very little deprivation	0.04	0.04	0.07	0.07	0.07	0.08	0.05	0.10	0.05
little deprivation	0.02	0.05	0.05	0.12	0.15	0.21	0.15	0.30	0.18
below average deprivation	0.14	0.26	0.20	0.11	0.18	0.09	0.08	0.11	0.04
average deprivation	0.03	0.03	0.07	0.03	0.07	0.02	0.02	0.02	0.01
above average deprivation	0.01	0.01	0.04	0.13	0.06	0.21	0.11	0.14	0.37
extreme deprivation	0.08	0.11	0.12	0.06	0.12	0.12	0.09	0.08	0.02
very extreme deprivation	0.05	0.07	0.07	0.05	0.13	0.05	0.06	0.05	0.02
maximum deprivation	0.01	0.01	0.05	0.17	0.03	0.08	0.29	0.06	0.25

In figure 4.5.3 the different shades of deprivation are plotted for the four race groups in South Africa. One can clearly see the disparity between race groups in terms of access to basic services. The Indian community in South Africa is very small and mostly concentrated in a few cities. A large number of households in the rural areas are made up of the African community, many living without access to basic services.

Figure 4.5.3: Bar chart for race groups

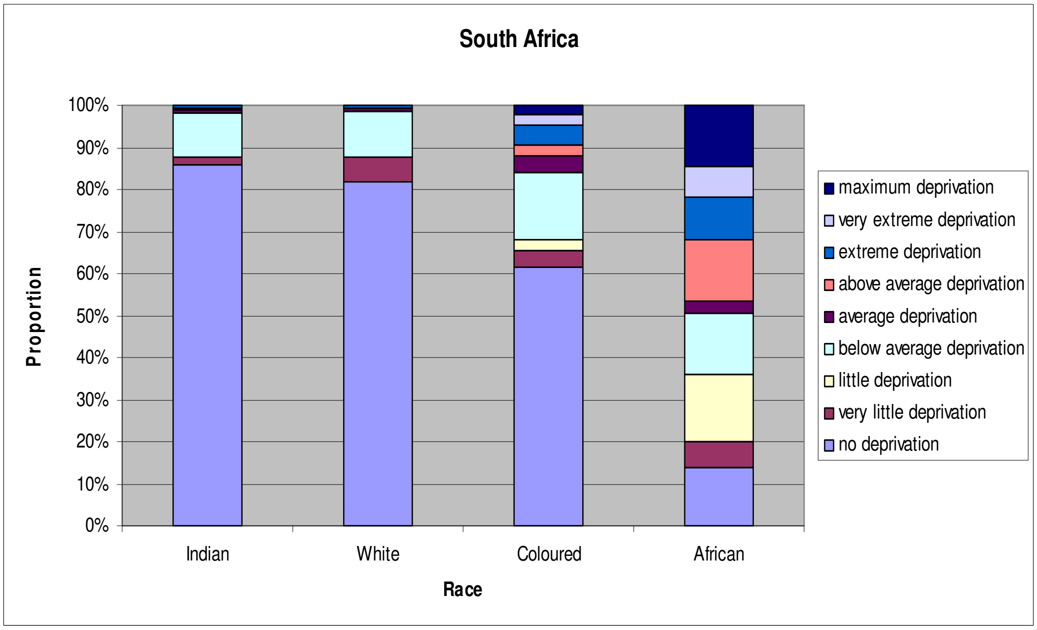


Figure 4.5.4: Bar chart for Africans across the 9 provinces

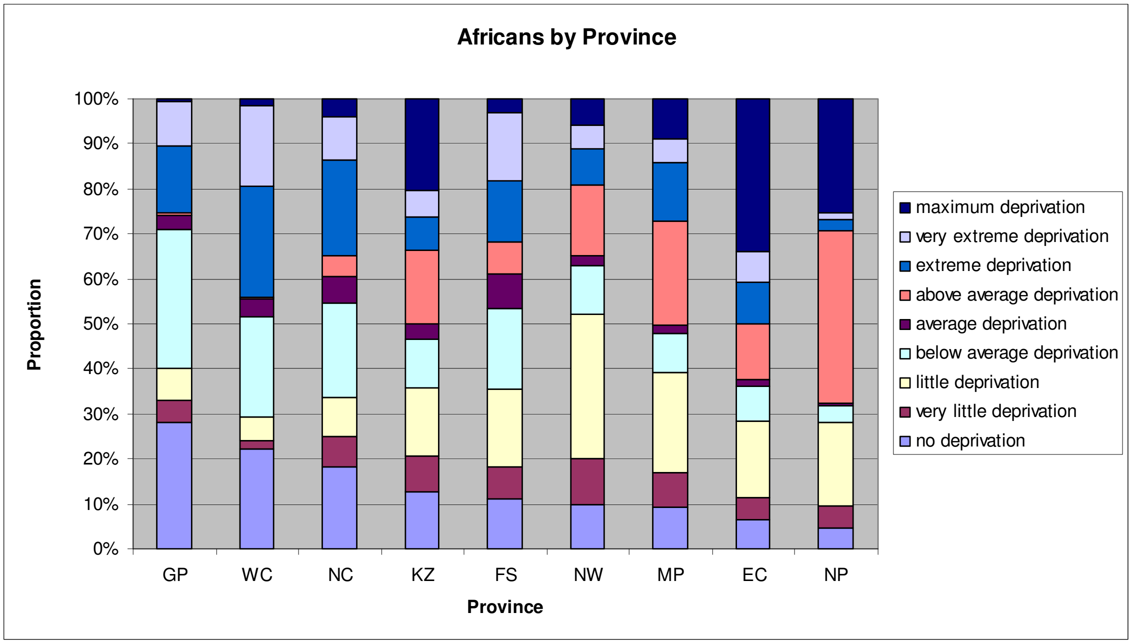


Figure 4.5.4 shows the bar chart for the African race group across the nine provinces. The different shades of deprivation of basic services can clearly be seen. In the Eastern Cape and KwaZulu Natal a large proportion of households experience maximum deprivation in respect to basic services. African households in Gauteng experience a higher proportion of no deprivation than any other province.

In figure 4.5.5 the bar chart is plotted for selected magisterial districts. Households in Roodepoort and Mitchell's Plain experience no deprivation or very little deprivation, while households in Flagstaff experience maximum deprivation or extreme deprivation. The multidimensional measure of poverty can be used to monitor the effectiveness of a poverty relief program.

Figure 4.5.5: Bar chart for magisterial districts

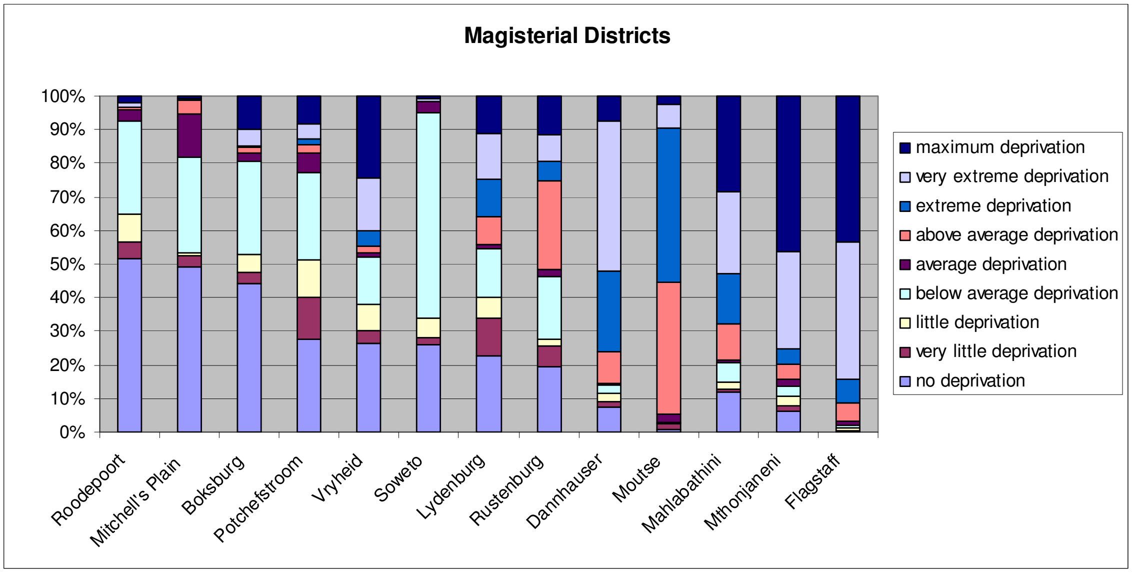


Table 4.5.2 shows the proportion of households in the selected magisterial districts that was used to plot the bar charts in figure 4.5.5.

The columns in table 4.5.2 are numbered from 1 to 9 to represent no deprivation to maximum deprivation respectively.

Table 4.5.2: Deprivation cross magisterial districts

	1	2	3	4	5	6	7	8	9
Roodepoort	0.52	0.05	0.08	0.28	0.03	0.01	0.00	0.01	0.02
Mitchell's Plain	0.49	0.03	0.01	0.29	0.13	0.04	0.00	0.00	0.01
Boksburg	0.44	0.03	0.06	0.27	0.03	0.02	0.01	0.05	0.10
Potchefstroom	0.28	0.13	0.11	0.26	0.06	0.02	0.02	0.04	0.08
Vryheid	0.27	0.04	0.08	0.14	0.01	0.02	0.05	0.16	0.24
Soweto	0.26	0.02	0.06	0.61	0.03	0.00	0.00	0.01	0.01
Lydenburg	0.23	0.11	0.07	0.14	0.01	0.08	0.11	0.14	0.11
Rustenburg	0.20	0.06	0.02	0.19	0.02	0.26	0.06	0.08	0.12
Dannhauser	0.08	0.02	0.02	0.02	0.00	0.10	0.24	0.44	0.07
Moutse	0.01	0.02	0.00	0.00	0.02	0.39	0.46	0.07	0.03
Mahlabathini	0.12	0.01	0.02	0.06	0.01	0.11	0.15	0.24	0.29
Mthonjaneni	0.06	0.02	0.03	0.03	0.02	0.04	0.05	0.29	0.46
Flagstaff	0.00	0.00	0.01	0.01	0.01	0.05	0.07	0.41	0.44

To compare the different methods discussed in this study, all the households in the magisterial district of Rustenburg were selected. In the Republic of South Africa 10% sample of Census 2001, there were 10 574 households for Rustenburg. Table 4.5.3 shows the 9 categories of the multi-dimensional measure of poverty for Rustenburg. The first column is the classification obtained using the Batch self organizing map. The second column is the results from the k-means cluster algorithm. The third column is the Kohonen vector quantization and the fourth column is the Kohonen self organizing map. The results from the Euclidean distance measure are shown in the last column.

The first comparison will be made between the Batch self organizing map and the Kohonen self organizing map. In both methods 1 738 households are classified in the "no deprivation" category. The question then arises: do the methods select the same households? To answer this question a two way contingency table is calculated.

Table 4.5.3: Magisterial district of Rustenburg: poverty categories

	Batch	Cluster	VQ	Kohonen	Euclidean
No deprivation	1 738	2 072	1 738	1 738	1 707
Very little deprivation	1 709	1 965	1 709	1 709	1 803
Little deprivation	958	619	958	958	1 221
Below average deprivation	122	222	126	126	774
Average deprivation	1 365	2 760	877	877	1 210
Above average deprivation	232	249	228	228	585
Extreme deprivation	2 987	641	658	658	1 485
Very extreme deprivation	636	823	3 646	3 646	1 126
Maximum deprivation	827	1 223	634	634	663
Total	10 574	10 574	10 574	10 574	10 574

Table 4.5.4 is the two way contingency for the 9 categories in the multi-dimensional measure of poverty for the Batch and Kohonen self organizing maps. In the first three categories both methods select exactly the same households. In the category "below average deprivation" 122 out of the 126 households are exactly the same. In the category "very extreme deprivation" the Kohonen self organizing maps method selects 3 646 households compared to the 636 households selected by the Batch method.

The Kohonen method tends to bunch many households in the extreme poverty categories. The Nadaraya-Watson and local-linear smoothing performed by the batch self organizing map method classifies houses more evenly in the extreme poverty categories.

Table 4.5.4: Cross tabulation: Kohonen and Batch self organizing maps

Kohonen self organizing map	Batch self organizing map									Total
	1	2	3	4	5	6	7	8	9	
No deprivation	1 738	0	0	0	0	0	0	0	0	1 738
Very little deprivation	0	1 709	0	0	0	0	0	0	0	1 709
Little deprivation	0	0	958	0	0	0	0	0	0	958
Below average deprivation	0	0	0	122	0	4	0	0	0	126
Average deprivation	0	0	0	0	877	0	0	0	0	877
Above average deprivation	0	0	0	0	0	228	0	0	0	228
Extreme deprivation	0	0	0	0	446	0	0	212	0	658
Very extreme deprivation	0	0	0	0	42	0	2 987	0	617	3 646
Maximum deprivation	0	0	0	0	0	0	0	424	210	634
Total	1 738	1 709	958	122	1 365	232	2 987	636	827	10 574

Out of the 10 574 households in Rustenburg, 55% were classified in the same categories of poverty by both methods. A further 40% of the households were classified within one category.

In the comparison between the k-means clustering and the Batch self organizing map, the two way contingency table was created as shown in table 4.5.5. The two methods select the same households in the first category of poverty. If one combines the first three categories of poverty, then 95% of the households are selected by both methods.

The difference arises in the middle categories. There are 1 365 households in the category "average deprivation" in the Batch method. The k-means cluster method categorises 675 of these households as "average deprivation", it categorises 327 as "zero deprivation", 83 as "extreme deprivation" and 194 as "very extreme deprivation".

Table 4.5.5: Cross tabulation: Batch self organizing map and k-means clustering

k-means cluster	Batch self organizing map									Total
	1	2	3	4	5	6	7	8	9	
No deprivation	1 707	0	30	7	327	0	0	1	0	2 072
Very little deprivation	0	1 570	393	0	2	0	0	0	0	1 965
Little deprivation	27	133	337	17	74	31	0	0	0	619
Below average deprivation	0	0	0	97	0	108	17	0	0	222
Average deprivation	0	0	0	0	675	0	1913	92	80	2 760
Above average deprivation	4	6	198	1	10	30	0	0	0	249
Extreme deprivation	0	0	0	0	83	23	152	321	62	641
Very extreme deprivation	0	0	0	0	194	1	391	219	18	823
Maximum deprivation	0	0	0	0	0	39	514	3	667	1 223
Total	1 738	1 709	958	122	1 365	232	2 987	636	827	10 574

A similar comparison is obtained between the Batch self organizing map and the Euclidean distance measure. In the category "extreme deprivation" the Euclidean distance measure classifies 1 485 households. Table 4.5.6 shows how these households are classified by the Batch self organizing map.

Table 4.5.6: Comparison of poverty category extreme deprivation

	Frequency	Percentage
No deprivation	0	0.00%
Very little deprivation	3	0.20%
Little deprivation	56	3.77%
Below average deprivation	3	0.20%
Average deprivation	223	15.02%
Above average deprivation	82	5.52%
Extreme deprivation	795	53.54%
Very extreme deprivation	265	17.85%
Maximum deprivation	58	3.91%
Total	1 485	100.00%

The Euclidean measure is a distance measure calculated from the origin to the household. The groupings of the categories are based on the length of the distance. All households on the arc created from the origin are grouped together; in this case the Euclidean distances between 1.3 and 1.5 are grouped in the category “extreme deprivation”. In spite of this spread, 53.54% of the households are correctly classified, while 17.85% of households are classified in the category above and 5.52% of the households are classified in the lower category.

CHAPTER FIVE

CONCLUSIONS

5.1 INTRODUCTION

The conclusions of this research are that poverty analysis and monitoring must be conducted on a multidimensional scale. Each attribute or dimension of poverty has grades and shades and should not be classified as poor or not poor. Poverty should not only be measured in monetary terms, non monetary aspects such as "access to basic services" are important. The multi-dimensional measure of poverty should not be aggregated to a single value but rather should be shown as shades or grades of deprivation.

Poverty is a phenomenon whose study is commonly oversimplified and its manifestation perceived as dichotomous, consequently its analysis is conventionally based merely over the splitting of the population into two groups: *poor* and *non-poor*, defined in relation to some chosen poverty line.

As an alternative to the conventional methodology, this thesis recognises poverty as a fuzzy set to which all members of the population belong in varying degrees. This method succeeds in avoiding the oversimplification in capturing the various degrees of poverty which affect different persons determined by the different individual's position in the income distribution.

Multivariate analysis seems to be the most proper choice if the aim is investigating poverty and deprivation of a given population.

The thesis attempts to assess the potential contribution of multi-dimensional analysis in terms of definition and measurement of poverty. Many studies have researched new approaches to provide poverty measures which account for multi-dimensionality. The fuzzy approach starts by selecting welfare indicators, choosing the membership function, aggregating the data in an index and weighting the variables.

The research developed alternative methods for aggregating the data without the need for weighting the variables. Many studies have condensed the multidimensional measure of poverty into a single index for purposes of comparison. The self organizing map algorithm avoids aggregation by plotting the vector of poverty indicators onto a two dimensional mapping grid.

This has reduced the need for the conceptual issue of how to counter multi-dimensional poverty. Many studies raise the question of multi-dimensional poverty as the accumulation of deprivation in various attributes (the intersection approach) or the failure to access one or more of the dimensions of poverty (the union approach). Instead of creating a single index, several shades or quantum of poverty are created in this research to accommodate both the union and intersection approach to poverty.

The number of segments developed provides a better view of the multi-dimensional aspects of poverty and deprivation and allows for an effective comparison of a poverty alleviation program on a group of households. The segments are created "before and after" for the households and a chi square test can measure the movement of households between segments, thus the effectiveness of the poverty program.

Households in the first segment experience zero poverty and households in the last segment experience maximum poverty (the intersection approach) and all the segments in between experience poverty in at least one dimension (union approach of poverty).

The distance measures provide for a ranking from the best off household to the worst off household in respect of selected dimensions of poverty.

Chapter 1 gives a definition on poverty with the literature study on poverty measurement and special attention paid to poverty studies on South Africa. The five approaches to poverty are introduced; the fuzzy set approach, the distance function approach, the information theory approach, the axiomatic derivations of multidimensional poverty indices and the Kohonen self organizing map.

Chapter 2 discusses the fuzzy set approach. The fuzzy membership function was applied to the Republic of South Africa Census data. A comparison of the nine provinces was made in respect of the head count ratio and the multi-dimensional measure using fuzzy membership.

Chapter 3 deals with the distance function approach. Fuzzy membership allows for categorical data to be analysed as continuous data, thus allowing for a ranking of each household according to a distance measure. The clustering technique was applied to created groups of households to demonstrate the union definition and the intersection definition of poverty. The clustering technique could not order the clusters in terms of deprivation

Chapter 4 considers the self organizing map. In this section three techniques were applied to the Republic of South Africa Census sample data. The Kohonen vector quantization also created clusters that could not be ordered in respect of deprivation. The Kohonen self organizing map created segments that could be ordered. Segment 1 comprised of the least deprived households in respect of basic services. This analysis could not order the segments accurately and also tended to group the worst off households together. The Batch self organizing map uses Nadaraya-Watson smoothing and local linear smoothing to create segments that are ordered. In a grid of 3 rows and 3 columns the first segment comprises of households that are least deprived and the last segment comprises of households that experience maximum deprivation.

The results from the batch self organizing map are further analysed to show how the multidimensional measure of poverty can effectively be used as a monitoring tool for poverty alleviation.

Finally, the methods described in this thesis will provide a viable poverty monitoring mechanism for developing countries. The multi-dimensional approach for measuring poverty is far more realistic than the traditional ones based on a single indicator of resources. This research will allow countries to measure and monitor poverty in a multi-dimensional manner by grouping together many dimensions and attributes of poverty.