Generating Normally Distributed Clusters by Means of a Self-organizing Growing Neural Network – An Application to Market Segmentation –

Reinhold Decker, Christian Holsing, and Sascha Lerke

***Abstract*—**This paper presents a new growing neural network for cluster analysis and market segmentation, which optimizes the size and structure of clusters by iteratively checking them for multivariate normality. We combine the recently published SGNN approach [8] with the basic principle underlying the Gaussian-means algorithm [13] and the Mardia test for multivariate normality [18, 19]. The new approach distinguishes from existing ones by its holistic design and its great autonomy regarding the clustering process as a whole. Its performance is demonstrated by means of synthetic 2D data and by real lifestyle survey data usable for market segmentation.

***Keywords*—**Artificial neural network, clustering, multivariate normality, market segmentation, self-organization

I. INTRODUCTION

CLUSTERING is one of the most manifold areas of data analysis and a topic of continuous research efforts, particularly due to its increasing importance for discovering significant patterns and characteristics in large databases [6]; its fields of application range from biology and medicine [12, 23] through chemistry and text mining [25, 3] to market basket analysis and market segmentation [9, 15]. The focus of this paper is on the latter field, but the neural network methodology discussed is also applicable to other clustering tasks.

Market segmentation is an essential part of marketing research [31] and stands for the strategy of dividing a heterogeneous market into homogeneous groups of customers with similar needs and meeting those needs with marketing efforts [1]. The heart of quantitative market segmentation is clustering. Due to the large diversity of individual needs [4] and the nearly omnipresent information flood in marketing communication, market segmentation is almost indispensable for successful marketing activities on saturated consumer markets [30, 31].

But the identification of the right or "natural" number of market segments or, more generally speaking, the number of groups in a dataset is one of the most difficult problems in cluster analysis [29]. Furthermore, marketing researchers often argue that adequate representations of market segments by means of easy-to-interpret cluster profiles (called centroids in the following) can ease the understanding of the inherent customer needs and preferences [28].

Dr. Reinhold Decker, Professor of Marketing, Department of Business Administration and Economics, Bielefeld University, P.O. Box 100131, 33615 Bielefeld, Germany (phone: +49-521-1066913; fax: +49-521-1066456; e-mail: rdecker@wiwi.uni-bielefeld.de).
Christian Holsing, Department of Business Administration and Economics, Bielefeld University (e-mail: christian.holsing@uni-bielefeld.de).
Sascha Lerke, Department of Business Administration and Economics, Bielefeld University (e-mail: sascha.lerke@uni-bielefeld.de).

Once the centroids are found, the closest customer profile (called medoid) can be considered as the cluster representative.

In market segmentation, this interpretation is particularly plausible when the customers assigned to one cluster are normally distributed around the centroid, at least approximately. Looking at the representatives of a target market segment is usual in customer-centric marketing communication. Therefore, flexible clustering methods like neural networks, that enable the computation of centroids with a substantial interpretation, are gaining increasing importance in market segmentation.

Although the importance of correctly answering the question of how many clusters exist in a dataset is emphasized in the relevant literature, there are comparatively few papers dealing with this problem in the market segmentation context in depth (see, e.g., [30] for details). If the true, but unknown number of clusters is over-specified, market researchers may over-segment the market and may treat segments separately that, in fact, could be treated inclusively. Conversely, if the true number of clusters is under-specified, they may under-segment the market and may fail to identify distinct segments that should be processed separately [5].

Conventional segmentation methods such as factor analysis, multidimensional scaling and hierarchical cluster analysis often suffer from the fact that the user has to control the data processing to a large extent. In hierarchical cluster analysis, which is widely used in market segmentation, for instance, the selection of the distance measure (for describing the attribute-based similarity of consumers), the fusion method (for computing the distance between two clusters or segments), and the criterion for determining the number of clusters [26], strongly influence the segmentation result as a whole. For data sets with a largely unknown internal structure this may imply an extensive search for an acceptable solution, particularly regarding identifiability, stability and actionability [32]. Partitioning clustering algorithms such as the widely used *k*-means or *k*-mediods [6], on the other hand, necessitate a pre-specification of the number of clusters or market segments (k) to be considered. Similar problems may arise from applying factor analysis as well, e.g. regarding the number of factors to be extracted or assumed, the communality estimation and the factor rotation. The higher the dimensionality of the data, the more difficult this task becomes.

The GNNCN (Growing Neural Network with integrated testing for Cluster-wise Normality) algorithm to be introduced in the next section determines the quality of the detected clusters by iteratively testing them for multivariate normality. Since

the user has to preset only one parameter, our approach does not make heavy demands on the user. Therefore, it can also be applied successfully by those who are not experienced in neural network-based clustering.

The remainder of the paper is organized as follows. In section II, we provide a brief synopsis of the methodological developments relevant for the new algorithm and outline key elements of the GNNCN approach. Its performance is demonstrated in section III by means of a synthetic 2D dataset with normally distributed clusters. In section IV the usefulness of the new approach in market segmentation is shown by analyzing real data collected from a German household panel. The paper concludes with a brief discussion and an outlook on future research.

II. METHODOLOGY

A. Basic algorithms and previous work

The probably most popular neural network model for clustering using unsupervised learning techniques is the Self-Organizing Feature Map (SOFM). This algorithm enables nonlinear projections from high-dimensional data manifolds onto two-dimensional grids, while largely preserving the topology of the data [16]. It is easy to use and available in many software packages. But in its basic form SOFM suffers from the fact that its application requires the prior determination of the network's size and dimensionality. As a consequence thereof, the topological structure and the capacity of the neural network is fixed in terms of the predefined number of nodes ("neurons") included [10, 20], which makes SOFM less suitable for market segmentation tasks, where the number of segments is typically unknown in advance. Therefore, we exclude this approach from our further considerations.

The Neural Gas Network (NGN) introduced by [21] is an interesting alternative to SOFM regarding the topological structure of the network. In NGN the nodes are adapted to the data without any fixed topological arrangement within the network. This "free movement" of the nodes in the data space is often compared to the movement of gas molecules in a real space. In contrast to Kohonen's SOFM, the NGN algorithm does not require any prior knowledge about the topological structure of the network, takes a smaller number of learning steps to converge, and its dynamics can be characterized by a global cost function [2].

Further developments of the NGN algorithm are, e.g., the Growing Neural Gas Network (GNGN) by [11] and the Self-controlled Growing Neural Network (SGNN) by [8]. Both algorithms are able to dynamically insert and remove nodes during the network adaptation process. The basic idea is to create a new node when the input data to be learned suggests an enlargement of the neural network in order to better represent the data set as a whole. Nodes with low "utility" concerning the goodness of data representation are removed during the adaptation process. Both algorithms differ, among other things, in the way the insertion of new nodes is controlled. The GNGN algorithm repeatedly inserts a new node after a prespecified number of adaptation steps. In contrast, SGNN adds a new node when the fit of the node, which best matches the current input data, is lower than a dynamically adapted threshold. We later refer to this by the term "activity". SGNN bears resemblance to the Grow When Required Network by [20], particularly regarding its structural design. But since the number of parameters, that have to be preset by the user in advance, is less with SGNN, it is more autonomous in view of the adaptation process.

The GNNCN algorithm to be introduced in the next section is based on the learning principles of SGNN and it is designed to support the identification of normally distributed clusters or market segments by taking advantage of the ideas underlying the cluster-wise normality check discussed in the Gaussian-means algorithm by [13] and the well-known Mardia test of multivariate normality [18, 19]. Gaussian-means clustering starts with a small number of clusters and successively expands this number until the input data assigned to each cluster looks Gaussian. The procedure used for testing normality is based on the Anderson-Darling statistics and measures the distortion of the data belonging to one cluster.

B. The GNNCN approach

Let B be a set of nodes ("neurons") u_h with $h \in \{1,\ldots,H = |B|\}$ and C the set of edges connecting these nodes. Both sets together determine the topological structure of an artificial neural network. Each node u_h is represented by a weight vector $\vec{\eta}_h = (\eta_{h1},\ldots,\eta_{hk},\ldots,\eta_{hK})$, where subscript $k \in \{1,\ldots,K\}$ refers to the object attributes used for clustering and subscript h denotes the different clusters. This is in line with the assumption that a given number of objects J can be partitioned into H groups with $H < J$ but unknown at the beginning of the clustering process. Each object $j \in \{1,\ldots,J\}$ is represented by an input vector $\vec{x}_j = (x_{j1},\ldots,x_{jk},\ldots,x_{jK})$ that includes the attributes considered. Accordingly, x_{jk} denotes the value of attribute k of object j. With these definitions a description of the new approach can be given:

Initialization step:

i. Create two non-connected nodes u_1 and u_2 with randomly generated positive weight vectors $\vec{\eta}_1$ and $\vec{\eta}_2$. Set $B = \{u_1, u_2\}$ and $C = \emptyset$.

ii. Initialize some internal control variables:
$y_1 = y_2 = 0$: With the firing counter y_h we take into account how often u_h has been the best matching node in the network adaptation process.
$w_1 = w_2 = 1$: The w-variables refer to the current training requirement of the respective nodes.
$v_{Thres} = 1$: The network growth process is controlled by a threshold for the activity (see above) of nodes.

$l = 0$: The adaptation step counter l controls the intensity of network adaptation between two tests of normality.
$m = 1$: The iteration counter m refers to individual repetitions of the test procedure.

iii. Define a training rate α, which controls the whole network adaptation process and simultaneously serves as the significance level for the cluster-wise tests of multivariate normality. The training rate is the only control parameter to be entered by the user externally. As a rule, $0.001 \le \alpha \le 0.1$ should apply.

Network adaptation step:

1. Select randomly one input vector $\vec{x}_j = (x_{j1}, \ldots, x_{jk}, \ldots, x_{jK})$ from the database and compute the Euclidean distances between input $\vec{x}_j$ and all nodes u_h or rather weight vectors $\vec{\eta}_h$:
$$dist(\vec{x}_j, \vec{\eta}_h) = \left\| \vec{x}_j - \vec{\eta}_h \right\| = \sqrt{\sum_{k=1}^{K} (x_{jk} - \eta_{hk})^2} \quad \forall u_h \in B$$
2. Determine both the best and the second best matching node $u_{h_{Best}}$ and $u_{h_{Second}} \in B$ in terms of their Euclidean distance to $\vec{x}_j$ and update the set of edges by $C = C \cup \{(h_{Best}, h_{Second})\}$ if $u_{h_{Best}}$ and $u_{h_{Second}}$ are not connected. Initialize the current age of the edge connecting $u_{h_{Best}}$ and $u_{h_{Second}}$ by $a_{h_{Best}h_{Second}} = 0$.
3. Compute the activity of the best matching node according to $v_{h_{Best}} = \exp(-dist(\vec{x}_j, \vec{\eta}_{h_{Best}}))$.
   Note: The extent to which the best matching node $u_{h_{Best}}$ or rather weight vector $\vec{\eta}_{h_{Best}}$ fits the current input vector is called its activity. The smaller the distance is, the higher the activity is [20].
4. Compute thresholds for both the activity $v_{h_{Best}}$ and the training requirement $w_{h_{Best}}$ of the best matching node:
   If $v_{h_{Best}} < v_{Thres}$ then $v_{Thres} = \frac{l \cdot v_{Thres} + v_{h_{Best}}}{l+1}$
   $w_{Thres} = \alpha^{\sqrt{|B|}} \cdot \frac{\ln(l+1)}{|B|}$
   Note: The first term compares the activity of the current winning node with the overall average (threshold). Depending on this comparison threshold v_{Thres} is adapted successively to control the node creation process. The second term defines the threshold of the training requirement. A large l together with a small number of nodes $|B|$, for example, indicates that these nodes have already been trained to a considerable degree. On the other hand, a small value of α causes the algorithm to strongly adapt the available weight vectors before a new node is added to the network.
5. If $v_{h_{Best}} \ge v_{Thres}$ or $w_{h_{Best}} \ge w_{Thres}$ then go to substep 6, else create a new node $u_{h_{New}}$: $\vec{\eta}_{h_{New}} = \frac{1}{2} \cdot (\vec{\eta}_{h_{Best}} + \vec{x}_j)$; update the set of edges, initialize the control variables $y_{h_{New}}$ and $w_{h_{New}}$ and go to substep 7.
6. Adapt the best matching node $u_{h_{Best}}$: $\vec{\eta}_{h_{Best}} = \vec{\eta}_{h_{Best}} + \Delta\vec{\eta}_{h_{Best}}$
   with $\Delta\vec{\eta}_{h_{Best}} = (1-\alpha) \cdot \sqrt{w_{h_{Best}}} \cdot (\vec{x}_j - \vec{\eta}_{h_{Best}})$
7. Increase the age $a_{h_{Best}h_i}$ of all edges (h_{Best}, h_i) emanating from the best matching node as well as the respective firing counter $y_{h_{Best}}$ by 1 and update the training requirement of the best matching node according to $w_{h_{Best}} = (y_{h_{Best}} + 1)^{-1}$.
8. Delete all edges $(h_i, h_{i'}) \in C$ with $a_{h_i h_{i'}} > \sqrt[3]{J \cdot K}$ as well as all nodes $u_{h_i} \in B$ with no connection to any other node $u_{h_{i' \ne i}}$, and set $l = l + 1$.
9. If $l \le \text{int}(10^5 \cdot \sqrt{K})$ then return to substep 1, else start tests of cluster-wise multivariate normality:
   i. For each h test whether the input vectors assigned to node u_h are normally distributed and increase the iteration counter m for the test procedure by 1.
   ii. If multivariate normality applies to all clusters or $m > \text{int}(\ln(J \cdot K))$ then STOP with $H = |B|$ else set $l = 1$ and return to substep 1.
   Note: The upper bound for m ensures the termination of the algorithm if multivariate normality can not been achieved for all clusters.

Test of cluster-wise multivariate normality:

According to [18, 19, 27] we use the multivariate sample kurtosis and the multivariate sample skewness in substep 9 to check whether the input vectors assigned to cluster h are normally distributed. The test statistic for the sample kurtosis is defined as follows:

$$b_h^{kurt} = \frac{d_h^{kurt} - K \cdot (K+2)}{\sqrt{8 \cdot K \cdot (K+2) / |N_h|}} \qquad \forall h = 1, \ldots, H \quad \text{with}$$

$$d_h^{kurt} = \frac{1}{|N_h|} \cdot \sum_{n \in N_h} \left[(\vec{x}_n - \vec{\eta}_h) \cdot S_h^{-1} \cdot (\vec{x}_n - \vec{\eta}_h)' \right]^2$$, where S_h denotes the cluster-specific sample covariance matrix and N_h is the set of input vectors assigned to node u_h. The above test statistic is asymptotically standard normally distributed with probability value $p_h^{kurt} = 2 \cdot (1 - P_N(|b_h^{kurt}|))$. Analogously, the test statistic for the sample skewness is:

$$b_h^{skew} = |N_h| \cdot d_h^{skew} / 6 \qquad \forall h = 1,\ldots,H \qquad \text{with}$$

$$d_h^{skew} = \frac{1}{|N_h|^2} \cdot \sum_{n \in N_h} \sum_{n' \in N_h} \left[(\vec{x}_n - \vec{\eta}_h) \cdot S_h^{-1} \cdot (\vec{x}_{n'} - \vec{\eta}_h)' \right]^3 .$$

This test statistic is asymptotically chi-square distributed with $p_h^{skew} = 1 - P_{\chi^2(K\cdot(K+1)\cdot(K+2)/6)}\left(b_h^{skew}\right)$.

The probability values p_h^{kurt} and p_h^{skew} can be used to check the inner cluster distribution. The null hypothesis of multivariate normality is rejected for cluster h when their values are smaller than significance level α. The special design of the presented approach explains its name GNNCN (growing neural network with integrated testing for cluster-wise normality). Figure 1 summarizes the general structure of GNNCN (the related substep numbers are given in brackets).

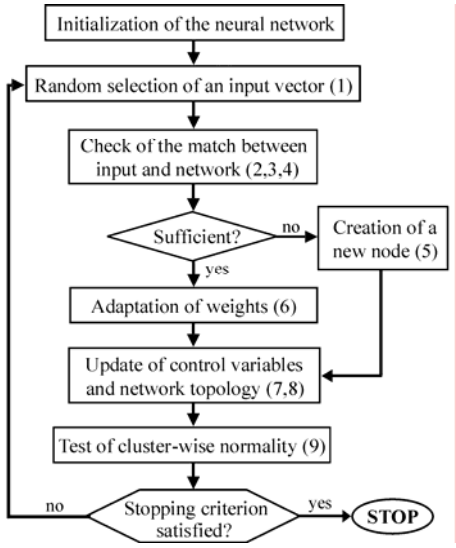

Fig. 1 General structure of the GNNCN algorithm

III. COMPARISON WITH EXISTING ALGORITHMS

The performance and adaptability of the new approach was analyzed by means of a 2D synthetic data set, which is similar to that used by [13]. It contains 7×1000 bivariate normally distributed data points scattered over seven separate clouds. For benchmarking our own approach we used the NGN and the GNGN algorithm sketched above. There are two main reasons for this selection. Firstly, as already outlined, both algorithms belong to the same class of clustering methods and can be regarded as early precursors of GNNCN. Secondly, both NGN and GNGN already proved to be powerful tools for the optimal partitioning of data in a comprehensive comparative study by [7] including *k*-means and growing *k*-means. The authors argue that the incremental character of GNGN makes this algorithm independent of the initial partition and gives the possibility to verify the "natural" number of clusters in data. In most cases NGN provides better partitions than those obtained by *k*-means in this study.

Figure 2 shows the best results of each of the above-mentioned algorithms. At first sight NGN (a), GNGN (b) and GNNCN (c1) led to almost the same goodness of data partitioning. Each node of the respective networks represents exactly 1000 data points. To complete the picture Mardia's kurtosis and skewness are given for GNNCN (c2). According to these measures all data points are normally distributed around their cluster centroids for $\alpha = 0.01$, the only parameter to be preset. This almost perfect solution was achieved after 989947 iterations. Therefore, each GNNCN weight vector (centroid) can be seen as the "representative" of the cluster it belongs to. This aspect will play an important role in market segmentation later on. In contrast to GNNCN the present implementations of NGN and GNGN require the preset of eight respectively seven parameters. In the case of NGN this also includes the number of clusters H, whereas GNGN determines the optimal number of clusters more autonomously. Since the internal structure of the data to be analyzed is often unknown in practice, the determination of adequate parameters for NGN and GNGN can imply time-consuming trial and error passes.

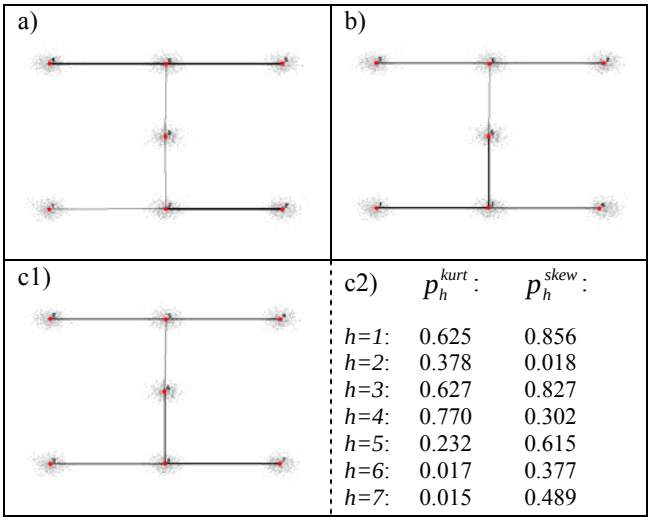

Fig. 2 Results of NGN (a), GNGN (b) and GNNCN (c1: clusters; c2: cluster-wise *p*-values of the Mardia tests)

In Table 1 some performance measures are given for each algorithm. The well-known Quantization Error *QE* [16] and the so-called Maximum Distance Error *MDE* [14] can be used to assess the accuracy of data representation by the neural networks. The latter refers to the worst matching of an input vector with its associated weight vector and therefore appraises the balance of pattern representation. Both measures should be as small as possible. Their formal definitions are as follows:

$$QE = \sum_{j=1}^{J} dist(\vec{x}_j, \vec{\eta}_{m_j}) \qquad \text{with } m_j = \arg \min_{h \in \{1,\ldots,H\}} dist(\vec{x}_j, \vec{\eta}_h)$$

$$\text{and } MDE = \max_h \left\{ \max_{j|m_j=h} dist(\vec{x}_j, \vec{\eta}_{m_j}) \right\}.$$

TABLE I
PERFORMANCE VALUES QE AND MDE AND NUMBERS OF ADAPTATION STEPS REQUIRED FOR REALIZING THE BEST SOLUTION

Algorithm	Adaptation steps	Quantization Error (QE)	Maximum Distance Error (MDE)
NGN	500000	17374.26	8.94
GNGN	1000000	17582.02	8.98
GNNCN	989947	17375.66	8.94

As already evident in Figure 2 all algorithms performed in an absolutely satisfactory manner. The "non-growing" NGN achieved its optimum with the lowest number of adaptation steps. GNGN and GNNCN performed quite similar. However, the latter solved the given clustering task with a minimum of a priori knowledge which significantly eases its application in practice.

IV. APPLICATION TO LIFESTYLE SURVEY DATA

In this section the GNNCN algorithm is used to determine a consumer typology from real survey data. The data was provided by the German ZUMA Institute [24]. It comprises attitudes and opinions with regard to daily nutrition and consumption behavior as well as socio-demographic characteristics of 1000 randomly selected households. The attitudes and opinions were measured on the basis of 61 5-point Likert-scaled statements. A Likert-scale [17] is an itemized rating scale that can be used to measure the level at which respondents agree or disagree with a given statement (with 1: "I definitely disagree.", … 5: "I definitely agree."). It is often used to measure attitudes, preferences, and subjective reactions, since it is easy to construct and easy to understand by the respondents. Typical items are "*I am fond of company.*" and "*I watch my figure when eating and drinking.*". The scale is assumed to be equidistant and therefore the data are treated as metric.

If we apply the GNNCN algorithm to this data with training rate $\alpha = 0.01$, we receive an 11-cluster solution with performance values *QE* = 8647.18 and *MDE* = 14.41. The number of consumers represented by the weight vectors $\vec{\eta}_1 = (\eta_{1,1},...,\eta_{1,65})$, … $\vec{\eta}_{11} = (\eta_{11,1},...,\eta_{11,65})$ ranges between 35 and 128. Each cluster or market segment can be represented by its centroid (or the related medoid). Two of these market segments will now be interpreted for illustration purposes.Cluster 10 is represented by the weight vector $\vec{\eta}_{10}$ depicted in Figure 3 and comprises 11 % of the consumers surveyed. By referring to item-specific weights the members of this cluster can be characterized as follows: They do not buy many new products that are unknown to other consumers (low value of $\eta_{10,5}$) and they do not own new products before their acquaintances (low value of $\eta_{10,16}$). They like to stay at home and have a strong preference for hearty plain fare (high values of $\eta_{10,10}$ and $\eta_{10,27}$). At the same time, they strictly reject vegetarian food and do not often use grains for cooking (low values of $\eta_{10,41}$ and $\eta_{10,55}$). The latter corresponds with the low interest of these consumers in preparing unusual or exotic dishes (low values of $\eta_{10,38}$ and $\eta_{10,50}$). Regarding their socio-demographics the members of cluster 10 typically have a below average education (rather low value of $\eta_{10,64}$; measured on a 9-point scale) and are characterized by their distinct price consciousness (rather high value of $\eta_{10,65}$; measured on a 4-point scale). Due to their particular profile the members of cluster 10 could be labeled as "the conservatives".

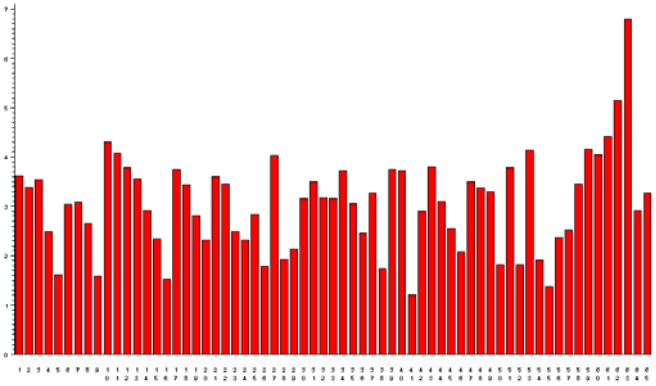

Fig. 3 Bar chart of cluster 10's item weights

Figure 4 shows the item weights defining centroid $\vec{\eta}_8$, which represents the attitudes and opinions of 7 % of the consumers. In contrast to cluster 10 the members of market segment 8 are characterized by a rather "undecided" or homogeneous profile. This finds its expression, for example, in the following virtual contradictions: The respective consumers are fond of company (high value of $\eta_{8,3}$), but at the same time they like to stay at home (high value of $\eta_{8,10}$). They watch their figure when eating and drinking and pay attention to a low-fat diet (high values of $\eta_{8,22}$ and $\eta_{8,34}$), but also love hearty and solid food (rather high value of $\eta_{8,43}$). The indecisiveness of these consumers regarding their individual lifestyle also becomes apparent by the rather high values of $\eta_{8,53}$ ("cooking well-tried recipes") and $\eta_{8,54}$ ("being very demanding when eating and drinking"). The members of cluster 8 have a medium education ($\eta_{8,64}$) and are less price conscious ($\eta_{8,65}$) than those of cluster 10, for instance.

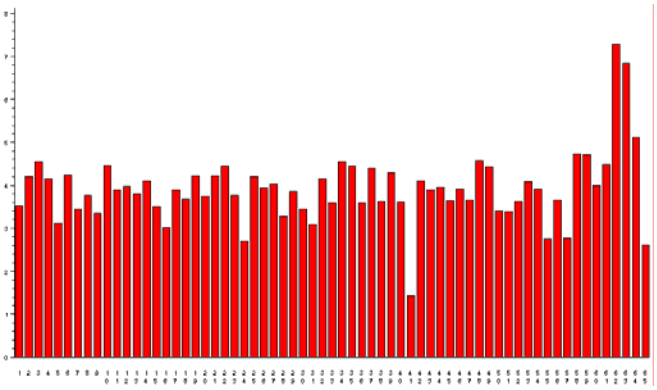

Fig. 4 Bar chart of cluster 8's item weights

The remaining clusters can be interpreted in a similar way. By this means GNNCN-based market segmentations provide an excellent basis for developing marketing strategies, since the resulting consumer profiles can be used to develop segment-centered communication strategies. Beyond the explication of consumption and nutrition styles for marketing communication the presented approach also yields two additional benefits: Firstly, by means of the cluster assignments we are able to provide criteria for compiling informative customer lists for direct marketing activities. Secondly, by linking the prototypical profiles (centroids or medoids) to media usage typologies, media planning and budgeting can be supported as well. The "undecided" consumers (according to cluster 8), for instance, may be a promising target for new customer recruitment activities in direct marketing or sales promotion.

V. CONCLUSIONS AND OUTLOOK

We introduced a new growing neural network approach for unsupervised clustering called GNNCN with a special focus on market segmentation, a field of market research, which will become more and more important in future and will be used to perform new tasks [32]. A performance test based on synthetic 2D data has shown that none of the compared clustering algorithms was significantly superior to the others regarding the goodness of data representation. However, by making minimal demands on the user with respect to its parameterization GNNCN proves to be a powerful tool, particularly in classification tasks, where prior knowledge about the internal structure of the data is missing. The GNNCN algorithm tends to generate market segments with a maximum number of normally distributed clusters, which may ease the interpretation of the segments and help to identify the "natural" number of clusters in a dataset. The strongly data-driven adaptation process contributes to avoiding both over- and under-segmentation and therefore may significantly unburden the market researcher regarding the necessity of later strategy adjustments.

Future research will focus on a further optimization of GNNCN regarding the computing time when both the dimensionality and the heterogeneity of the data are high. This, for instance, becomes relevant when applying GNNCN to market basket or brand image analysis, which are further promising fields of application for unsupervised clustering in marketing research [8]. Another outstanding task is the verification of alternative measures for testing for multivariate normality. Corresponding options are discussed in [22].

REFERENCES

[1] Assael, H. (1993): Marketing - Principles and Strategy, 2nd ed., Dryden Press, Fort Worth.

[2] Atukorale, A.; Suganthan, N. (2000): Hierarchical Overlapped Neural Gas Network with Application to Pattern Classification, Neurocomputing, Vol. 35, 165-176.

[3] Berry, M. (Ed.) (2003): Survey of Text Mining: Clustering, Classification, and Retrieval, Springer, New York.

[4] Bock, T.; Uncles, M. (2002): A Taxonomy of Differences Between Consumers for Market Segmentation, International Journal of Research in Marketing, Vol. 19, 215-224.

[5] Boone, D.S.; Roehm, M. (2002): Evaluating the Appropriateness of Market Segmentation Solutions Using Artificial Neural Networks and the Membership Clustering Criterion, Marketing Letters, Vol. 13, No. 4, 317-333.

[6] Chu, S.-C.; Roddick, J.F.; Pan, J.-S. (2002): An Incremental Multi-Centroid, Multi-run Sampling Scheme for *k*-Medoids-based Algorithms, Knowledge Discovery and Management Laboratory, Technical Report KDM-02-003, Flinders University of South Australia.

[7] Daszykowksi, M.; Walczak, B.; Massart, D.L. (2002): On the Optimal Partitioning of Data with k-means, Growing k-means, Neural Gas, and Growing Neural Gas, Journal of Chemical Information and Computer Sciences, Vol. 42, 1378-1389.

[8] Decker, R. (2005): Market Basket Analysis by Means of a Growing Neural Network, International Review of Retail, Distribution and Consumer Research, Vol. 15, No. 2, 151-169.

[9] Decker, R.; Monien, K. (2003): Market Basket Analysis with Neural Gas Networks and Self-organizing Maps, Journal of Targeting, Measurement and Analysis for Marketing, Vol. 11, No. 4, 373-386.

[10] Dittenbach, M.; Rauber, A.; Merkl, D. (2002): Uncovering Hierarchical Structure in Data Using the Growing Hierarchical Self-organizing Map, Neurocomputing, Vol. 48, 199-216.

[11] Fritzke, B. (1995): A Growing Neural Gas Network Learns Topologies, in: Tesauro, G.; Touretzky, D.S.; Leen, T.K. (Eds.): Advances in Neural Information Processing Systems 7, MIT Press, 625-632.

[12] Glotsos, D.; Tohka, J.; Ravazoula, P.; Cavouras, D.; Nikiforidis, G. (2005): Automated Diagnosis of Brain Tumours Astrocytomas Using Probabilistic Neural Network Clustering and Support Vector Machines, International Journal of Neural Systems, Vol. 15, No. 1-2, 1-11.

[13] Hamerly, G.; Elkan, C. (2003): Learning the k in k-means, Advances in Neural Information Processing Systems, Vol. 17.

[14] Hoff, K.E.; Culver, T.; Keyser, J.; Lin, M.; Manocha, D. (1998): Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware, Department of Computer Science, University of North Carolina.

[15] Hruschka, H.; Natter, M. (1999): Comparing Performance of Feedforward Neural Nets and K-means for Cluster-based Market Segmentation, European Journal of Operational Research, Vol. 114, No. 2, 346-353.

[16] Kohonen, T. (2001): Self-Organizing Maps, 3rd ed., Springer, Berlin.

[17] Likert, R. (1932): A Technique for the Measurement of Attitudes, Archives of Psychology, Vol. 140, 1-55.

[18] Mardia, K.V. (1970): Measures of Multivariate Skewness and Kurtosis with Applications, Biometrika, Vol. 57, 519-530.

[19] Mardia, K.V. (1974): Applications of Some Measures of Multivariate Skewness and Kurtosis for Testing Normality and Robustness Studies, Sankhya: Indian Journal of Statistics, Vol. 36, Series B, 115-128.

[20] Marsland, S.; Shapiro, J.; Nehmzow, U. (2002): A Self-Organising Network that Grows When Required, Neural Networks, Vol. 15, No. 8-9, 1041-1058.

[21] Martinetz, T.; Schulten, K. (1991): A 'Neural Gas' Network Learns Topologies, in: Kohonen, T.; Maekisara, K.; Simula, O.; Kangas, J. (Eds.): Artificial Neural Networks, North Holland, Amsterdam, 397-402.

[22] Mecklin, C.J.; Mundfrom, D.J. (2004): An Appraisal and Bibliography of Tests for Multivariate Normality, International Statistical Review, Vol. 72, No. 1, 123-138.

[23] Ozbay, Y.; Ceylan, R.; Karlik, B. (2006): A Fuzzy Clustering Neural Network Architecture for Classification of ECG Arrhythmias, Computers in Biology & Medicine, Vol. 36, No. 4, 376-88.

[24] Papastefanou, G.; Schmidt, P.; Börsch-Supan, A.; Lüdtke, H.; Oltersdorf, U. (Eds.) (2001): Social Research with Consumer Panel-Data, GESIS, Mannheim.

[25] Questier, F.; Guo, Q.; Walczak, B.; Massart, D.L.; Boucon, C.; de Jong, S. (2002): The Neural-Gas Network for Classifying Analytical Data, Chemometrics and Intelligent Laboratory Systems, Vol. 61, No. 1-2, 105-121.

[26] Sander, J.; Qin, X.; Lu, Z.; Niu, N.; Kovarsky, A. (1999): Automatic Extraction of Clusters from Hierarchical Clustering Representations, Working Paper, Department of Computing Science, University of Alberta Edmonton.

[27] Schwager, S.J. (1985): Testing For Multivariate Normality, in: Kotz, S.; Johnson, N.L. (Eds.): Encyclopedia of Statistical Sciences, Vol. 6, John Wiley & Sons, New York, 122.

[28] Solomon, M.R.; Marshall, G.W.; Stuart, E.W. (2005): Marketing – Real People, Real Choices, 4th ed., Prentice Hall, Upper Saddle River.

[29] Sugar, C.A.; James, G.M. (2003): Finding the Number of Clusters in a Dataset: An Information-theoretic Approach, Journal of the American Statistical Society, Vol. 98, No. 463, 750-762.

[30] Wagner, R.; Scholz, S.W.; Decker, R. (2005): The Number of Clusters in Market Segmentation, in: Baier, D.; Decker, R.; Schmidt-Thieme, L. (Eds.): Data Analysis and Decision Support, Springer, Berlin, 166-176.
[31] Wedel, M.; Kamakura, W. (2000): Market Segmentation: Conceptual and Methodological Foundations, 2nd ed., Kluwer Academic Publishers, Dordrecht.
[32] Wirth, N.; Dietrich, H. (2005): Management of Segmentation Projects, Yearbook of Marketing and Consumer Research, Vol. 3, 22-36.