The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data

Andreas Rauber, Dieter Merkl, and Michael Dittenbach
Department of Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstr. 9-11 / 188, A-1040 Vienna, Austria
e-mail: {andi, dieter, mbach}@ifs.tuwien.ac.at

Abstract

The *Self-Organizing Map* is a very popular unsupervised neural network model for the analysis of high-dimensional input data as in data mining applications. However, at least two limitations have to be noted, which are related, on the one hand, to the static architecture of this model, as well as, on the other hand, to the limited capabilities for the representation of hierarchical relations of the data. With our novel *Growing Hierarchical Self-Organizing Map* presented in this paper we address both limitations. The *Growing Hierarchical Self-Organizing Map* is an artificial neural network model with hierarchical architecture composed of independent growing self-organizing maps. The motivation was to provide a model that adapts its architecture during its unsupervised training process according to the particular requirements of the input data. Furthermore, by providing a global orientation of the independently growing maps in the individual layers of the hierarchy, navigation across branches is facilitated. The benefits of this novel neural network are first, a problem-dependent architecture, and second, the intuitive representation of hierarchical relations in the data. This is especially appealing in explorative data mining applications, allowing the inherent structure of the data to unfold in a highly intuitive fashion.

Keywords: Self-Organizing Map (SOM), Data Mining, Hierarchical Clustering, Exploratory Data Analysis, Pattern Recognition.

1 Introduction

Data mining, or more generally, pattern recognition and knowledge acquisition, heavily depend on suitable unsupervised learning methods. The purpose of these methods is to develop an optimal partitioning, i.e. clustering, of the data set to be analyzed. Cluster analysis is the organization of a collection of patterns, which are usually represented as vectors of measurements or points in a multidimensional space, into clusters based on similarity. Intuitively, patterns within a valid cluster are more similar to each other than to a pattern belonging to a different cluster [1]. In other words, the objective of unsupervised learning methods in data mining applications is to identify groupings in an unlabeled set of data vectors that share semantic similarities. This helps the user to build a cognitive model of the data, thus fostering the detection of the inherent structure and the interrelationship of data. However, in many applications little to no prior information about underlying models for the data is available. In such a situation clustering provides a particularly appropriate approach to the analysis of data.

The *Self-Organizing Map* (*SOM*) is being widely used as a tool for mapping high-dimensional data into a two-dimensional representation space [2]. This mapping retains the relationship between input data as faithfully as possible, thus describing a topology-preserving representation of input similarities in terms of distances in the output space. It is then possible to visually identify clusters on the map. The main advantage of such a mapping is the ease by which a user gains an idea regarding the structure of the data by analyzing the map.

However, some difficulties in *SOM* utilization remained largely untouched, even though a large number of research papers on applications of the *SOM* were presented over the years [3]. First, the *SOM* uses a fixed network architecture in terms of number and arrangement of neural processing elements, which has to be defined prior to training. Obviously, in case of largely unknown input data characteristics, it remains far from trivial to determine the network architecture that provides satisfying results. Thus, it certainly is worth considering neural network models that determine the number and arrangement of units during their unsupervised training process. We refer to [4, 5, 6] for recently proposed models that are based on the *SOM*, yet allow for adaptation of the network architecture during training.

Second, hierarchical relations between the input data are not mirrored in a straight-forward fashion. Such relations are rather shown within the same representation space and are thus hard to identify. Hierarchical relations, however, may be observed in a wide spectrum of application domains. Thus, their proper identification remains a highly important data mining task that cannot be addressed conveniently within the framework of the *SOM*.

To resolve both limitations of the *SOM* in a uniform fashion we propose a novel artificial neural network architecture in this paper, namely the *Growing Hierarchical Self-Organizing Map* (*GHSOM*). This model has a hierarchical architecture, where *SOM*-like neural networks with adaptive architecture build the various layers of the hierarchy. The size of these *SOM*-like neural networks as well as the depth of the hierarchy of the *GHSOM* is determined during its unsupervised training process according to the structure of the data.

The hierarchical structuring imposed on the data results in a separation of clusters mapped onto different branches. While this, in principle, is a desirable characteristic helping to understand the cluster structure of the data, it may lead to misinterpretations when large clusters are mapped onto and expanded from two neighboring, yet different units. Similar input data might thus be rather arbitrarily separated into different branches of the hierarchy. By choosing the initial orientation of deeper layers according to their respective higher-layer maps we can maintain the already learned similarities between input data during the creation of the hierarchical structure of the *GHSOM*. As a consequence, the negative effects of generating strictly disjoint clusters are eliminated because neighboring maps in deeper layers of the hierarchy show similar characteristics at their respective borders.

We show the usefulness of the *Growing Hierarchical Self-Organizing Map* with an application in document archive organization. Document archives represent a convenient application scenario because they are, by their very nature, represented as high-dimensional data. In particular, we show the results from two experiments. The first one is based on the *TIME Magazine* collection. This collection comprises 420 articles from the *TIME Magazine*, covering a variety of topics ranging from international politics to social gossip. The second experiment is based on a much larger document collection of more than 10,000 articles from the daily Austrian newspaper *Der Standard*.

The remainder of the paper is structured as follows: Section 2 provides an introduction to the architecture and training process of the *Self-Organizing Map*, followed by a review of related architectures in Section 3. The *GHSOM* architecture is introduced and presented in detail in Section 4. Two different data sets are used to demonstrate the characteristics and capabilities of the *GHSOM* model in Section 5, starting with the smaller *TIME Magazine* collection in Section 5.1, followed by the more extensive collection of articles from the newspaper *Der Standard* in Section 5.2. Some remarks conclude the paper in Section 6.

2 The Self-Organizing Map

The *Self-Organizing Map* (*SOM*), as proposed in [2] and described thoroughly in [7, 8, 9] is one of the most distinguished artificial neural network models adhering to the unsupervised learning paradigm. The *SOM* is a general unsupervised tool for the ordering of high-dimensional data in such a way that similar items are grouped spatially close to one another.

The range of applications where the *SOM* has been utilized successfully is impressive, see [3] for a fairly recent bibliography. The model also has a strong tradition in the text mining area where a number of research groups described work based on the *Self-Organizing Map* [10, 11, 12, 13, 14].

The *SOM* consists of a number of neural processing elements, i.e. units that are arranged according to some topology, the most common choice of which is marked by a two-dimensional rectangular or hexagonal grid. Each of the units i is further assigned a model vector m_i, $m_i \in \Re^n$. It is important to note that these model vectors have the same dimensionality as the input patterns.

The training process of *SOMs* may be described in terms of input pattern presentation and model vector adaptation. Each training iteration t starts with the random selection of one input pattern x, $x \in \Re^n$. This pattern is presented to the *SOM* and each unit determines its activation. Usually, the Euclidean distance between input pattern and model vector is used to calculate a unit's activation. In this case, the unit having the model vector with the smallest Euclidean distance to the input pattern is referred to as the *winner*. We will use the index c for denoting the *winner*, cf. Expression (1).

$$c(t) = \arg\min_i \{||x(t) - m_i(t)||\} \qquad (1)$$

Finally, the model vector of the *winner* as well as model vectors of units in the vicinity of the *winner* are adapted. This adaptation is implemented as a gradual reduction of the difference between corresponding components of the input pattern and the model vector, as shown in Expression (2). Note that we make use of discrete time notation with t denoting the current training iteration.

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)] \qquad (2)$$

Geometrically speaking, the model vectors of the adapted units are moved a bit towards the input pattern. The amount of model vector movement is guided by a learning rate α, decreasing in time. The number of units that are affected by adaptation as well as the strength of adaptation depending on a unit's distance from the *winner* is determined by a neighborhood function h_{ci}. This number of units also decreases in time such that towards the end of the training process only the *winner* is adapted. Typically, the neighborhood function is a unimodal function which is symmetric around the location of the winner and monotonically decreasing with increasing distance from the winner. Often a Gaussian is used as a neighborhood function as given in Expression (3).

$$h_{ci}(t) = \exp\left(-\frac{||r_c - r_i||^2}{2 \cdot \delta(t)^2}\right) \qquad (3)$$

In this expression $||r_c - r_i||^2$ denotes the distance between units c and i within the output space, with r_i representing the two-dimensional location vector of unit i within the grid. The time-varying parameter δ guides the reduction of the neighborhood kernel during training. It is common practice that this neighborhood kernel is selected large enough to cover a wide area of the output space in the beginning of learning. The spatial width of the kernel is reduced gradually during training such that towards the end of the process just the *winner* is adapted.

3 Related Architectures

During the last years, a number of modifications have been suggested to enhance the usefulness of the *Self-Organizing Map* for data mining applications. In particular the identification of inter- and intra-cluster similarity has been addressed. Approaches such as the *U-Matrix* [15], *Adaptive Coordinates*, and *Cluster Connections* [16] represent techniques that put emphasis on detection and visualization of cluster structures in *SOMs*. These techniques analyze the distances between neighboring units or mirror the effect of the model vector adaptation in the two-dimensional output space. Similar cluster information can be obtained with our *LabelSOM* method described in [17, 18]. Using the *LabelSOM* method the characteristics of the various units are described in terms of shared features among the input patterns mapped onto a particular unit. Grouping units that have the same characteristics allows to identify clusters within the output space of the *SOM*. Other modifications of the architecture address the problems arising from the mapping onto a lattice structure, providing a smooth manifold as output space, such as the *Self-Organizing Field* [19], or the *Generative Topographic Mapping (GTM)* [20]. However, none of the methods identified above facilitates the detection of hierarchical structure inherent in the data or adapt the size of the network.

The *Hierarchical Feature Map* [21] tries to uncover the hierarchical structure of data by modifying the *SOM* architecture. Instead of training a flat *SOM*, a balanced hierarchical structure of *SOMs* is trained. Data mapped onto one single unit is represented at a further level of detail in the lower-level map assigned to this unit. However, this model merely represents the data in a hierarchical way, rather than really reflecting the hierarchical structure of the data. This is because the architecture of the network has to be defined in advance, i.e. the number of layers and the sizes of the maps at each layer are fixed prior to network training. This leads to the definition of a balanced tree which is used to represent the data. Desirable is, however, a network architecture defined by the peculiarities of the input data.

In [22, 23] a *Tree-Structured SOM*, a hierarchical modification of the *SOM*, is introduced. The focus in this model, however, is rather on computational speedup during *winner* selection by using a tree-based organization of the units. This model does not provide a hierarchical decomposition of the input space, the input patterns are again organized in one single flat *SOM*.

The shortcoming of having to define the size of the *SOM* in advance has been addressed by a number of different models. Consider, for example the *Incremental Grid Growing* [4], *Growing Grid* [5], *Growing SOM* [6], and the *Hypercubical SOM* [24]. The first, i.e. *Incremental Grid Growing*, allows the addition of new units at the boundary of the map. Furthermore, connections between units of the map may be established and removed according to some threshold settings based on the similarity of their respective model vectors. This may result in several separated, irregularly shaped map structures. Being quite similar in spirit, the *Growing SOM* uses a spread factor to control the growth process of the map. Using manual intervention, the supervisor can decide to train separate *SOMs* for specific units in order to obtain a more detailed representation. This obviously results in manually created hierarchies, an approach that is basically possible with each variant of the *SOM*.

The *Growing Grid*, on the other hand, adds rows and columns of units during the training process, starting with a *SOM* of initially 2×2 units. The decision where to insert new units is governed by the computation of some measure for each unit, e.g. the winner counter. As an extension, the *Hypercubical SOM* allows for a growth process of the *SOM* into more than two dimensions, thus providing improved data representation while forsaking visual interpretability.

It can be stated, however, that the main focus of each of these adaptive variants of the *Self-Organizing Map* lies with an equal distribution of the input patterns accross the map by adding new units in the neighborhood of units that represent an unproportionally high number of input data. They thus do not primarily reflect the concept of representation at a certain level of detail, which is expressed in terms of the overall quantization error rather than in the number of input data mapped onto specific areas. Moreover, neither of these adaptive models takes the inherently hierarchical structure of data into account.

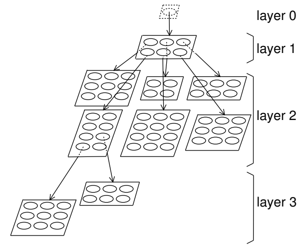


Figure 1: **Trained *GHSOM*:** The *GHSOM* evolves to a structure of *SOMs* reflecting the hierarchical structure of the input data.

4 The Growing Hierarchical Self-Organizing Map

4.1 The principles

While the *SOM* has proven to be a very suitable tool for detecting structure in high-dimensional data and organizing it accordingly on a two-dimensional output space, some shortcomings have to be mentioned. These include its inability to capture the inherent hierarchical structure of data. Furthermore, the size of the map has to be determined in advance when proper insight into the characteristics of a data distribution might not be available. These drawbacks have been addressed separately in several modified architectures of the *SOM* as outlined in Section 3. However, none of these approaches provides an architecture which fully adapts to the characteristics of the input data. To overcome the limitations of both fix-sized and non-hierarchically adaptive architectures we developed the *Growing Hierarchical Self-Organizing Map* (*GHSOM*), which dynamically fits its multi-layered architecture according to the structure of the data [25].

The *GHSOM* has a hierarchical structure of multiple layers, where each layer consists of several independent growing *Self-Organizing Maps*. Starting from a top-level map, each map, similar to the *Growing Grid* model, grows in size to represent a collection of data at a specific level of detail. After a certain improvement regarding the granularity of data representation is reached, the units are analyzed to see whether they represent the data at a specific minimum level of granularity. Those units that represent too diverse input data are expanded to form a new small growing *SOM* at a subsequent layer, where the respective data shall be represented in more detail. These new maps again grow in size until a specified improvement of the quality of data representation is reached. Units representing an already rather homogeneous set of data, on the other hand, will not require any further expansion into subsequent layers. The resulting *GHSOM* thus is fully adaptive to reflect, by its very architecture, the hierarchical structure inherent in the data, allocating more space for the representation of inhomogeneous areas in the input space.

A graphical representation of a *GHSOM* is given in Figure 1. The map in layer 1 consists of 3×2 units and provides a rather rough organization of the main clusters in the input data. The six independent maps in the second layer offer a more detailed view of the data. The input data for one map is the subset which has been mapped onto the corresponding unit in the upper layer. Two units from one of the second-layer maps have further been expanded into third-layer maps to provide sufficiently granular input data representation. It has to be noted that the maps have different sizes according to the structure of the data, which relieves us from the burden of predefining the structure of the architecture. The layer 0 serves as a representation of the complete data set and is necessary for the control of the growth process.

4.2 Training algorithm

4.2.1 Initial setup and global network control

The principle of the *GHSOM* architecture is its adaptation to the training data. The quality of this adaptation is measured in terms of the deviation between a unit's model vector and the input vectors represented by this particular unit. Basically, two different strategies can be used for the control of the growth process, using either the mean quantization error of a unit (which is commonly used as a quality measure for data representation with *SOMs*), or the absolute value, i.e. the quantization error of a unit.

More formally, the mean quantization error of a unit i is calculated according to Expression 4 as the mean Euclidean distance between its model vector m_i and the $n_\mathcal{C}$ input vectors x_j that are elements of the set of input vectors $\mathcal{C}_i$ mapped onto this unit i:

$$mqe_i = \frac{1}{n_\mathcal{C}} \cdot \sum_{x_j \in \mathcal{C}_i} ||m_i - x_j||, \qquad n_\mathcal{C} = |\mathcal{C}_i|, \mathcal{C}_i \neq \emptyset \tag{4}$$

The starting point for the *GHSOM* training process is the calculation of a mean quantization error mqe_0 of the unit forming the layer 0 map as provided in Expression 5. With $n_\mathcal{I}$ we refer to the number of all input vectors x of the input data set $\mathcal{I}$ and m_0 denotes the mean of the input data.

$$mqe_0 = \frac{1}{n_\mathcal{I}} \cdot \sum_{x_i \in \mathcal{I}} ||m_0 - x_i||, \qquad n_\mathcal{I} = |\mathcal{I}| \tag{5}$$

The *mqe* measures the dissimilarity of all input data mapped onto a particular unit and will be used to control the growth process of the neural network. Specifically, the minimum quality of data representation of each unit will be specified as a fraction, indicated by a parameter τ_2, of mqe_0.

In other words, all units must represent their respective subsets of data at a mean quantization error smaller than a fraction τ_2 of mqe_0, i.e. satisfy the global termination criterion specified in Expression 6:

$$mqe_i < \tau_2 \cdot mqe_0 \tag{6}$$

For all units not satisfying this condition, a more detailed data representation is required, leading to the addition of further units to provide more map space for data representation.

Alternatively, the quantization error of the unit as given in Expression 7, denoted by qe, may be used instead of the mean quantization error, resulting in a global termination criterion as given in Expression 8.

$$qe_i = \sum_{x_j \in \mathcal{C}_i} ||m_i - x_j|| \tag{7}$$

$$qe_i < \tau_2 \cdot qe_0 \tag{8}$$

While using the *mqe* of the data distribution as a global quality measure for the *GHSOM* training process may be more intuitive, using the *qe* follows more closely the principle characteristic of *Self-Organizing Maps* of providing more map space for more densely populated regions of the input space, also referred to as *magnification factor*. Thus, using Expression 8 rather than Expression 6 as a global stopping criterion produces maps that more intuitively reflect the characteristics of data distributions by capturing finer differences in more densely populated clusters. This effect is specifically important when the resulting maps shall be used as explorative interfaces to data sets, as it is frequently the case for *SOM*-like architectures. We will thus, for the remainder of this paper, use the quantization error *qe* as a basis for *GHSOM* training.

Following the decision whether to use *mqe* or *qe* for the global control of the training process, the *GHSOM* architecture is initialized by creating a new growing *SOM* beneath the layer 0 map. The initial size of this first-layer map is set to 2×2 units, with its model vectors being initialized to random values.

4.2.2 Training and growth process of a growing SOM

A newly created map is trained according to the standard *SOM* training procedure as described in

Section 2. After a fixed number λ of training iterations the qes of all units as provided in Expression 7 are analyzed. A high qe indicates that an inhomogeneous part of the input space containing dissimilar data, or at least a rather large set of input data from a more homogenous part of the input space is represented by this unit. Therefore, new units are needed to provide more space for appropriate data representation. The unit with the highest qe is thus selected and is denoted as the *error unit*. We will refer to the error unit as e. Next, the most dissimilar neighboring unit d in terms of input space distance is selected. This is done by comparing the model vectors of all neighboring units with the model vector of the error unit e. A new row or column of units is inserted between the error unit e and its most dissimilar neighbor d. The model vectors of the new units are initialized as the average of their corresponding neighbors.

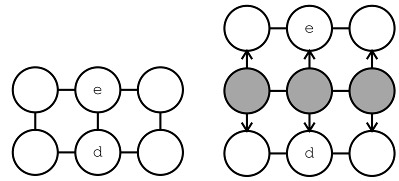
Figure 2 shows a graphical representation of the insertion process of our realization of a growing *SOM*, with the newly inserted units being depicted as shaded circles. The arrows point to the respective neighboring units used for model vector initialization.

More formally, the growth process of a growing *SOM* can be described as follows. Let $\mathcal{C}_i$ be the subset of vectors x_j of the input data that is mapped onto unit i, i.e. $\mathcal{C}_i \subseteq \mathcal{I}$; and m_i the model vector of unit i. Then, the error unit e is determined as the unit with the maximum qe as provided in Expression 9:

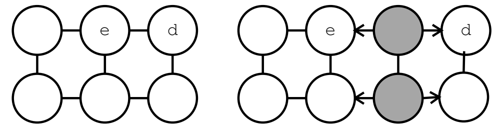
$$e = \arg\max_i \left(\sum_{x_j \in \mathcal{C}_i} ||m_i - x_j|| \right), \qquad n_{\mathcal{C}} = |\mathcal{C}_i|, \mathcal{C}_i \neq \emptyset \tag{9}$$

Please note, that the mean quantization error may be used instead of the quantization error, resulting in a *GHSOM* architecture focusing on overall homogeneity of data representation, rather than capturing higher degrees of detail for more densly populated areas of the data space.

Following the selection of the error unit, its most dissimilar neighbor d is determined as listed in Expression 10, where $\mathcal{N}_e$ is the set of neighboring units of the error unit e:



(a) Insertion of a Row



(b) Insertion of a Column

Figure 2: **Insertion of Units:** A row (a) or a column (b) of units (shaded gray) is inserted in between *error unit* e and the neighboring unit d with the largest distance between its model vector and the model vector of e in the Euclidean space.

$$d = \arg\max_i \left(||m_e - m_i|| \right), \qquad m_i \in \mathcal{N}_e \tag{10}$$

A row or column of units is inserted between d and e. To obtain a smooth positioning of the newly added units in the input space, their model vectors are initialized as the means of their respective neighbors. After insertion the learning rate and neighborhood range are reset to their original values, and training continues in a *SOM*-like fashion for the next λ iterations.

This training process of single growing *SOM* is highly similar to the *Growing Grid* model [5]. The difference so far is that we use a decreasing learning rate and a decreasing neighborhood range instead of fixed values.

4.2.3 Termination of growth process

As more units are added to the growing *SOM* their qes decrease as each of the units represents a smaller, more concise subset of the input space. Basically, the training process could continue until all units satisfy the global stopping criterion, i.e. they represent their respective subset of the data at a granularity lower than a certain fraction of the initial standard deviation of the data, resulting in a large *SOM* representing all data at the required granularity in one layer.

Yet, in order to reveal the hierarchical structure present in the data, each map shall only explain a portion of data similarity. The growth process thus continues only until the map's mean quantization error, referred to as MQE in capital letters, reaches a certain fraction τ_1 of the qe_u of the corresponding unit u in the upper layer (i.e. the unit constituting the layer 0 map for the first-layer map). The MQE of a map is computed as the mean of all units' quantization errors qe_i (cf. Expression 7) of the subset $\mathcal{U}$ of the maps' units onto which data is mapped:

$$MQE_m = \frac{1}{n_{\mathcal{U}}} \cdot \sum_{i \in \mathcal{U}} qe_i, \qquad n_{\mathcal{U}} = |\mathcal{U}| \tag{11}$$

In general terms, the stopping criterion for the growth of a single map m is defined as:

$$MQE_m < \tau_1 \cdot qe_u \tag{12}$$

where qe_u is the quantization error of the corresponding unit u in the upper layer. Obviously, the smaller the parameter τ_1 is chosen the larger the resulting map will be, explaining its data at a higher granularity. For larger τ_1, more detailed data representation will be delegated to additional maps further down the hierarchy. The parameter τ_1 thus serves as the control parameter for the depth/shallowness of the resulting hierarchical *GH-SOM* architecture.

In case of the first-layer map the stopping criterion for the training process is $MQE_1 < \tau_1 \cdot qe_0$.

4.2.4 Hierarchical growth with global map orientation

When the training of a map is finished according to the criterion specified in Expression 12, every unit has to be checked for fulfillment of the global stopping criterion given in Expression 8. Units representing a set of too diverse input vectors, are expanded to form a new map at a subsequent layer of the hierarchy. Units satisfying the global stopping criterion require no further expansion.

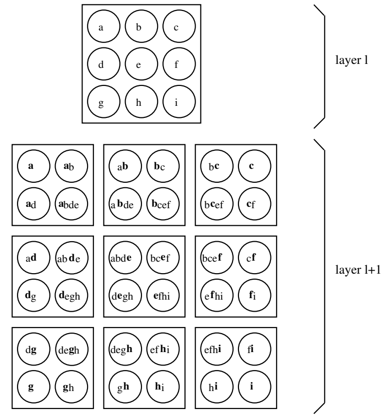


Figure 3: **Initialization of Units:** The corner units of newly created maps are initialized to preserve the orientation of their parents' map.

Similar to the procedure chosen for the creation of the layer 1 map, originating from the single-unit map at layer 0, a new map of initially 2×2 units is created. While, again, random initialization can be chosen for model vector orientation, this usually will distort the global topology of neighboring maps. This is because the orientation of data on this map, i.e. which subclusters are located on which area of the map, is determined by the self-organizing process following the random orientation of the map in data space. Navigation across maps within the same layer of the hierarchy would thus be prevented, leading to serious disadvantages of the organization of disjoint clusters in a hierarchical manner, especially when larger clusters should be split. To provide a global orientation of the individual maps in the various layers of the hierarchy, their orientation must conform to the orientation of the data distribution on their parent's map. This can be achieved by creating a coherent initialization of the units of a newly created map [26].

Let unit p be expanded to form a new 2×2 map in the subsequent layer of the hierarchy. This map's four model vectors $s1$ to $s4$ are initialized to mirror the orientation of neighboring units of its parent p. Figure 3 provides an illustration of the initialization of new maps and the influence of the parent unit's neighbors. Geometrically speaking, the model vectors of the four corner units are moved in data space towards the directions of their respective parent's neighbors by a certain fraction. This initial orientation of the map is preserved during the training process. While new units may be inserted in between, the four corner units will still be most similar to the respective corner units of the maps in neighboring branches. The exact amount by which these corner units are moved in the respective directions does not influence the characteristic of the topology-preserving initialization. We can thus choose to set them to the mean of the parent and its neighbors in the respective directions, e.g. setting $e1 = (e + a + b + d)/4$. In the simplest case, the neighbors' model vectors may be used directly as initial corner positions of the new maps in data space.

The input vectors to train the newly added map are the ones mapped onto the unit which has just been expanded, i.e. the subset of the data space mapped onto its parent. This map will again continue to grow following the procedures detailed in Section 4.2.2. The whole process is repeated for the subsequent layers until the global stopping criterion given in Expression 8 is met by all leaf units.

4.2.5 Analysis of GHSOM characteristics

The *SOM* offers itself for the analysis of large, high-dimensional data collections, allowing many shortcuts to accelerate the training process, see [10] for a concise treatment. Additionally, due to its hierarchical structuring and partitioning of the input data, the *GHSOM* provides improved scalability. At the transition from one layer to the next, the number of input vectors used for training a particular map decreases to the subset of vectors mapped onto the respective upper-layer unit.

Furthermore, each map in the hierarchy explains a particular set of characteristics of its input data. Subsequently, some input vector components, i.e. features of the data set, can be expected to be almost identical for all input vectors mapped onto a specific unit. These features can be ignored at the transition to a subsequent layer of the hierarchy, allowing to shorten the input vectors. Especially for very high-dimensional and sparse data sets this effect allows for a significant reduction of computational effort.

The training and growth process of the *GHSOM* is entirely data driven, requiring no prior knowledge or estimates for parameter specification. The hierarchical structure of data can be represented in different forms, favoring either (a) lower hierarchies with rather detailed refinements presented at each subsequent layer, or (b) deeper hierarchies, which provide a stricter separation of the various sub-clusters by assigning separate maps. Parameter τ_1 is used to control this trade-off between shallow or deep hierarchies.

In the first case we will prefer larger maps in each layer, which explain larger portions of the data in their flat representation, yet providing less hierarchical structuring. As an extreme example we might consider the *Growing Grid*, which grows in size explaining the complete structure of the data in one single flat map. It ignores all hierarchical information and tries, at best, to preserve it in the mapping of various clusters on the flat structure. On the other hand we might consider setting τ_1 rather large, which requires only limited growth of individual maps, resulting in a deeper hierarchical structure of small maps focusing on the hierarchical structure. Basically, the total number of units at the lowest-level maps may be expected to be similar in both cases as this is the number of neural processing units necessary for representing the data at the required level of granularity.

In principle, the choice of τ_1 may seem crucial, as it might result in a rather arbitrary separation of, a larger cluster with homogeneous data distribution into two or more sub-clusters in different branches. However, due to the global orientation provided by the initialization of the model vectors of new maps, navigation across maps in the same layer of the hierarchy is facilitated, offsetting the damage of cluster separation. Furthermore, by analyzing the distances in input space of model vectors on neighboring map boundaries, the similarity of neighboring maps can be detected and indicated.

For exploratory data analysis, a homogeneous distribution of data samples across the map space is desired, allowing to capture finer differences between clusters in more densely populated areas of the data space. Thus, using the quantization error of a unit, rather than its mean quantization error, has shown to produce more favorable results. The global stopping criterion, be it an absolute value or a fraction specified by parameter τ_2, directly influences the overall size of the resulting *GHSOM*, i.e. the number of units available for data space representation.

It should further be noted, that the training process usually does not lead to a balanced hierarchy in terms of all branches having the same depth. This is one of the main advantages of the *GHSOM* over the *Hierarchical Feature Map* [21], because the structure of the hierarchy adapts itself according to the requirements of the input space. Therefore, areas in the input space that require more units for appropriate data representation create deeper branches than others.

5 Experiments

For the following experiments we use an information retrieval application as a testbed for the *Growing Hierarchical Self-Organizing Map*. In a nutshell, topical clusters shall be detected in a collection of free-form documents with documents covering similar topics to be grouped together. This application domain represents an ideal and challenging scenario for clustering algorithms, as typically very high-dimensional feature spaces are involved. Furthermore, the data can be considered highly noisy as a result from the indexing process that is used to approximately capture the content of a particular document.

Two different experimental settings are presented, focusing on different characteristics of the *GHSOM*. In Section 5.1 the classic *TIME Magazine* collection is used to compare the characteristics and features of the hierarchical structuring of a data collection with respect to a *SOM*. In Section 5.2, we analyze the characteristics of shallow and deep hierarchies depending on various settings of parameter τ_1, and present the benefits of the preservation of the global orientation in the *GHSOM* hierarchy. For these experiments we use a larger collection of news articles from the Austrian newspaper *Der Standard*. All experiments presented in this paper as well as an implementation of the presented *GHSOM* architecture are available at the *SOMLib* project homepage[1] for interactive evaluation.

5.1 Experiment 1: TIME Magazine

5.1.1 Data representation

In a first step, the documents have to be mapped into some representation language in order to enable further analysis. This process is termed indexing in the information retrieval literature. A number of different strategies have been suggested over the years of information retrieval research. Still one of the most common representation techniques is single term full-text indexing, where the text of the documents is accessed and the various words forming the document are extracted. These words can be reduced to their word stem yielding the terms used to represent the documents. The resulting set of terms is further cleared from stop-words, i.e. words that appear either too often or too rarely within the document collection and thus have only little influence on the discrimination between different documents and would just unnecessarily increase the computational load during classification.

In the vector-space model of information retrieval the documents contained in a collection are represented by means of feature vectors x of the form $x = [\xi_1, \xi_2, \ldots, \xi_n]^T$. In such a representation, the i, $1 \leq i \leq n$, correspond to the index terms extracted from the documents as described above. The specific value of ξ_i corresponds to the importance of index term i in describing the content of the particular document at hand. One might find a lot of strategies to prescribe the importance of an index term for a particular document [27]. Without loss

[1] `http://www.ifs.tuwien.ac.at/~andi/somlib`

of generality, we may assume that this importance is represented as a scalar in the range of $[0,1]$ where *zero* means that this particular index term is absolutely unimportant to describe the document. Any deviation from *zero* towards *one* is proportional to the increased importance of the index term at hand. In such a vector-space model, the similarity between two documents corresponds to the similarity of their vector representations [28].

The *TIME* Magazine article collection consists of 420 articles from the *TIME* magazine of the 1960's, covering a broad range of topics from political issues to social gossip. The indexing process identified 5923 content terms, i.e. terms used for document representation, by omitting words that appear in more than 90% or less than 1% of the documents. The terms are roughly stemmed and weighted according to a *tf* $\times$ *idf*, i.e. term frequency times inverse document frequency, weighting scheme [29]. This weighting scheme assigns high values to terms that appear frequently within one document yet rarely within the overall document collection, i.e. to terms that are considered important in describing the contents of a document. Following the feature extraction process we end up with 420 vectors describing the documents in the 5923-dimensional feature space. These vectors are further used for neural network training.

5.1.2 A SOM of the TIME Magazine Collection

Figure 4 shows a *Self-Organizing Map* trained with the *TIME* Magazine data set. It consists of 10×15 units represented as table cells with a number of articles being mapped onto each individual unit. The articles mapped onto the same or neighboring units are considered to be similar to each other in terms of the topic they deal with. Due to space considerations we cannot present all the articles in the collection. We thus selected a number of units for detailed discussion.

We find that the *SOM* has succeeded in creating a topology preserving representation of the topical clusters of articles. For example, in the lower left

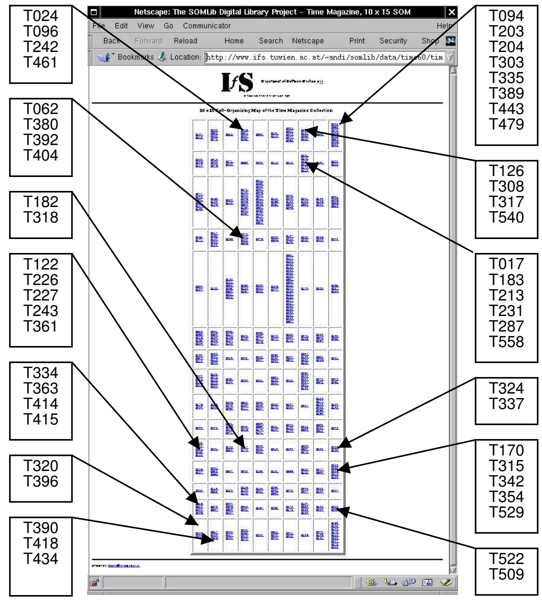


Figure 4: **Flat SOM:** 10×15 map of the *Time Magazine* collection

corner we find a group of units representing articles on the conflict in Vietnam. To name but a few, we find articles *T320, T369* on unit (1/15)[2], or *T390, T418, T434* on the neighboring unit (2/15) dealing with the government crack-down on Buddhist monks, next to a number of articles on units (4/15), (5/15) and neighboring ones, covering the fighting and suffering during the Vietnam War.

A cluster of documents covering affairs in the Middle-East is located in the lower right corner of the map around unit (10/15), next to a cluster on the Profumo–Keeler affair, a political scandal in Great Britain in the 1960's, on and around units (10/11) and (10/12). Above this area, on units (10/6) and neighboring ones we find articles on elections in Italy and possible coalitions, next to two units (10/3) and (10/4) covering elections in India. Similarly, all other units on the map can be identified to represent a topical cluster of news articles. For a more detailed discussion of the articles and topic clusters found on this map, we refer to [30].

While we find the *SOM* to provide a good topologically ordered representation of the various topics

[2]We use the notion (x/y) to refer to the unit located in column x and row y of the map, starting with (1/1) in the upper left corner

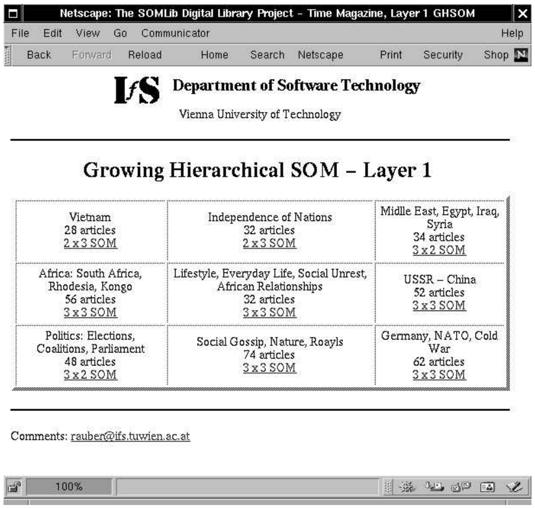


Figure 5: **GHSOM Layer 1** of the TIME Magazine article collection *GHSOM*

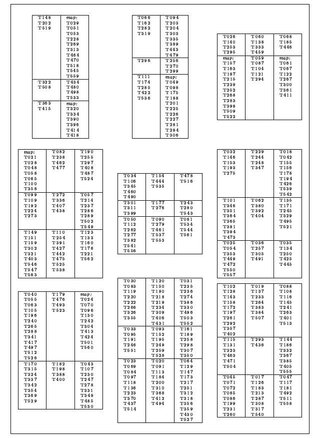


Figure 6: **GHSOM Layer 2:** 1 SOM per unit of layer 1 map

found in the article collection, no information about topical hierarchies can be identified from the resulting flat map. Apart from this we find the size of the map to be quite large with respect to the number of topics identified. This is mainly because the size of the map has to be determined in advance, before any information about the number of topical clusters is available.

5.1.3 A hierarchical archive of the TIME Magazine collection

Based on the unit representing the mean of all data points at layer 0, the *GHSOM* training algorithm starts with a 2×2 *SOM* at the first layer. The training process for this map continues with additional units being added until the quantization error drops below a certain percentage of the overall quantization error of the unit at layer 0. The resulting first-layer map is depicted in Figure 5. The map has grown twice, adding one row and one column respectively, resulting in 3×3 units representing 9 major topics in the document collection.

For convenience we list the topics of the various units, rather than the individual articles in the figure. For example, we find unit (1/1) to represent all articles related to the situation in Vietnam, whereas Middle-East topics are covered on unit (3/1), or articles related to elections and other political topics on unit (1/3) in the lower left corner, to name but a few.

Based on this first separation of the most dominant topical clusters in the article collection, further maps were automatically trained to represent the various topics in more detail. This results in nine individual maps on the second layer, each representing the data of the respective higher-layer unit in more detail. Some of the units on these second-layer maps were further expanded as distinct *SOMs* in the third layer.

The resulting second-layer maps are depicted in Figure 6. Please note that the maps on the second layer have grown to different sizes according to the structure of the data. In particular, we find small 3×2 maps representing the articles of unit (3/1) and (1/3) of the first-layer map and up to 3×3 maps for the units (1/2), (2/2), (3/2), (2/3) and (3/3). Taking a more detailed look at the first map of the second layer representing unit (1/1) on the first layer we find it to give a clearer representation of articles covering the situation in Vietnam. Units

(1/1) and (2/1) on this map represent articles on the fighting during the Vietnam War, whereas the remaining units represent articles on the internal conflict between the catholic government and Buddhist monks. At this layer, the two units (2/1) and (2/3) have further been expanded to form separate maps with 3×3 units each at the third layer.

To give another example of the hierarchical structures identified during the *Growing Hierarchical Self-Organizing Map* training process, we take a look at the 3×2 map representing the articles of unit (1/3) of the first-layer map. All of these articles were found to deal with political matters on the first layer. This common topic is now displayed in more detail at the resulting second-layer map. For example, we find unit (3/1) to represent articles on the elections in India. This unit is expanded to form a 3×4 map in the third layer. Next to these, on unit (3/2) we find articles covering the elections and discussions about political coalitions between Socialists and Christian Democrats in Italy. The remaining units on this map deal with different issues related to the Profumo-Keeler scandal in Great Britain, covering the political hearings in parliament, background information on this scandal and the persons involved, as well as related issues pertaining to the elections in Great Britain.

As a last example, consider the 3×3 map representing articles of unit (3/3) of the first layer. In the first layer we find this unit to cover articles related to east-west relationships, mainly dealing with post-war Germany, the relationships between Germany and the Soviet Union and the NATO. In the second-layer map we find these topics to be separated clearer, with units (1/1) to (3/1) covering mainly Germany-related articles, whereas the other two topics are represented by the remaining four units. In this case, no further expansions onto third-layer maps were necessary, as all articles are already represented in sufficient detail on the units of the second-layer map.

5.1.4 Comparison of SOM and GHSOM representation

When comparing the *GHSOM* with a *SOM* we can identify the locations of the articles on the nine second-layer maps on a corresponding 10×15 *SOM*. This allows us to view the hierarchical structure of the data on the flat map. We find that, for example, the cluster on Vietnam simply forms one larger coherent cluster on the flat map in the lower left corner of the map covering the rectangle spanned by the units (1/14) and (5/15). The same applies to the cluster of Middle-East affairs, which is represented by the map of unit (3/1) in the *Growing Hierarchical Self-Organizing Map*. This cluster is mainly located in the lower right corner of the *SOM*. The cluster of political affairs, represented by unit (1/3) on the first layer of the *GHSOM* and explained in more detail on its subsequent layers, is spread across the right hand side of the *SOM*, covering more or less all units on columns 9 and 10 and between rows 3 and 12. Note, that this common topic of political issues is not easily discernible from the overall map representation in the *SOM*, where exactly this hierarchical information is lost. The subdivision of this cluster on political matters becomes further evident when we consider the second-layer classification of this topic area, where the various sub-topics are clearly separated, covering Indian elections, Italian coalitions and the British Profumo-Keeler scandal.

Another interesting hierarchical structure not evident from the *SOM* is represented by the East-West relationships cluster on unit (3/3) of the *GHSOM*. When identifying the areas on the *SOM* that are represented by this branch in the *GHSOM*, we find that it covers two areas. This is, on the one hand, a group of seven units in the upper left corner of the *SOM* representing the Germany-related articles, whereas a second area in the upper right area of the *SOM* covers the NATO-related articles in this cluster. The relationship between these two sub-clusters is lost in the large *SOM*. This may be because of the size of the *SOM*, where the the overall organization of the map needs to be determined during the very first training steps when the neighborhood range of the learning function still covers a large area of the *SOM*. A similar situation can be identified for several smaller clusters, which are scattered across different areas on the *SOM*, but nicely combined in the first layer of the *Growing Hierarchical Self-Organizing Map* and further analyzed and separated as independent sub-clusters on subsequent layers.

Yet another interesting feature of the *GHSOM* we want to emphasize is the overall reduction in map size. During analysis we found the second layer of the *GHSOM* to represent the data at about the same level of topical detail as the corresponding *SOM*. However, the number of units of all individual second-layer maps combined is only 69 as opposed to 150 units in the 10×15 *SOM*. With the *GHSOM* model, this number of units is determined automatically, and only the necessary number of units is created for each level of detail representation required by the respective layer. Furthermore, not all branches are grown to the same depth of the hierarchy. As can be seen from Figure 6, only some of the units are further expanded in a third-layer map. With the resulting maps at all layers of the hierarchy being rather small, activation calculation and winner evaluation of the *GHSOM* is by orders of magnitude faster than in the *SOM* model. Apart from the speed-up gained by the reduced network size, orientation for the user is highly improved as compared to the rather huge maps which cannot be easily comprehended as a whole.

5.2 Experiment 2: Two Hierarchies of Newspaper Articles from "Der Standard"

5.2.1 Data representation

In the second experiment we will take a closer look at the influence of parameter τ_1, providing a trade-off between shallow and deep hierarchies, as well as the topology-preserving orientation of the various maps in different branches of the hierarchy. For these we use a larger collection of 11,627 articles from the Austrian newspaper *Der Standard* covering the second quarter of 1999. To be used for map training, a vector-space representation of the single documents is created by full-text indexing. Instead of defining language or content specific stop word lists, we rather discard terms that appear in more than 813 (7%) or in less than 65 articles (0.56%). We end up with a vector dimensionality of 3,799 unique terms. The 11,627 articles thus are represented by automatically extracted 3,799-dimensional feature vectors of word histograms weighted by a $tf \times idf$ weighting scheme and normalized to unit length. These feature vectors are used to train two *GHSOM*s. The mqe_0 of the layer 0 map evaluates to 12,180.3, serving as the basis for the global stopping criterion.

5.2.2 Deep Hierarchy

Training the *GHSOM* with parameters $\tau_1 = 0.07$ and $\tau_2 = 0.0035$ results in a rather deep hierarchical structure of up to 13 layers. Since it is impossible to present the complete topic hierarchy of three months of news articles, we will concentrate on some sample topical sections. The first-layer map depicted in Figure 7(a) has grown to a size of 4×4 units, all of which are expanded at subsequent layers. Among the well separated main topical branches we find sports, culture, radio- and TV programs, the political situation on the Balkan, national politics, business, or weather reports, to name but a few. These topics are clearly identifiable by the automatically extracted keywords using the *LabelSOM* technique [17, 18], such as *weather*, *sun*, *reach*, *degrees* for the section on weather reports[3]. The branch of articles covering the political situation on the Balkan is located in the upper left corner of the top-layer map labeled with *Balkan*, *Slobodan Milosevic*, *Serbs*, *Albanians*, *UNO*, *Refugees*, and others.

We find the branch on national politics in the lower right corner of this map listing the three largest political parties of Austria as well as two key

[3]We provide English translations for the original German labels.

politicians as labels. This unit has been expanded to form a 4×4 map in the second layer as shown in Figure 7(b). The upper left area of this map is dominated by articles related to the Freedom Party, whereas, for example, articles focusing on the Social Democrats are located in the lower left corner. Other dominant clusters on this map are neutrality, or the elections to the European parliament, with one unit carrying specifically the five political parties as well as the term *election* as labels. Two units of this second-layer map are further expanded in a third layer, such as, for example, the unit in the lower right corner representing articles related to the coalition of the People's Party and the Social Democrats. These articles are represented in more detail by a 3×4 map in the third layer.

5.2.3 Shallow Hierarchy

To show the effects of different parameter settings we trained a second *GHSOM* with τ_1 set to half of the previous value ($\tau_1 = 0.035$), while τ_2, i.e. the absolute granularity of data representation, remained unchanged. This leads to a more shallow hierarchical structure of only up to seven layers, with the first-layer map evolving to a size of 7×4 units. Again, we find the most dominant branches to be, for example, sports, located in the upper right corner of the map, national politics in the lower right corner, Internet-related articles on the left hand side of the map, to name but a few. However, because of the large size of the resulting first-layer map, a fine-grained representation of the data is already provided at this layer. This results in some larger clusters to be represented by two neighboring units already at the first layer, rather than being split up at a lower layer of the hierarchy. For example, we find the cluster on national politics to be represented by two neighboring units. One of these, on position (6/4), covers solely articles related to the Freedom Party and its political leader Jörg Haider, representing one of the most dominant political topics in Austria for some time now, resulting in an accordingly large number of news articles covering this topic. The neighboring unit to the right, i.e. located in the lower right corner on position (7/4), covers other aspects of national politics, with one of the main topics being the elections to the European parliament. Figure 8 shows these two second-layer maps.

However, we also find articles related to the Freedom Party on this second branch covering the more general national politics, reporting on their role and campaigns for the elections to the European parliament. As might be expected these are closely related to the other articles on the Freedom Party, which are located in the neighboring branch to the left. Obviously, we would like them to be presented on the left hand side of this map, so as to allow the transition from one map to the next, with a continuous orientation of topics. Because of the initialization of the added maps during the training process, this continuous orientation is preserved, as can easily be seen from the automatically extracted labels provided in Figure 8. Continuing from the second-layer map of unit (6/4) to the right we reach the according second-layer map of unit (7/4), where we first find articles focusing on the Freedom Party, before moving on to the Social Democrats, the People's Party, the Green Party and the Liberal Party. As all units at layer two in these branches have a quantization error below 42.63, no unit is further expanded at a third layer.

We thus find the global orientation to be well preserved in this map. Even though the cluster of national politics is split into two dominant subclusters in the more shallow hierarchy, the articles are organized correctly on the two separate maps in the second layer of the hierarchy. This allows the user to continue his or her exploration across map boundaries. For this purpose, the labels of the upper layer's neighboring unit serves a general guideline as to which topic is covered by the neighboring map. In the deeper hierarchy, these two sub-clusters are represented within one single branch in the second layer, covering the upper and the lower area of the map, respectively.

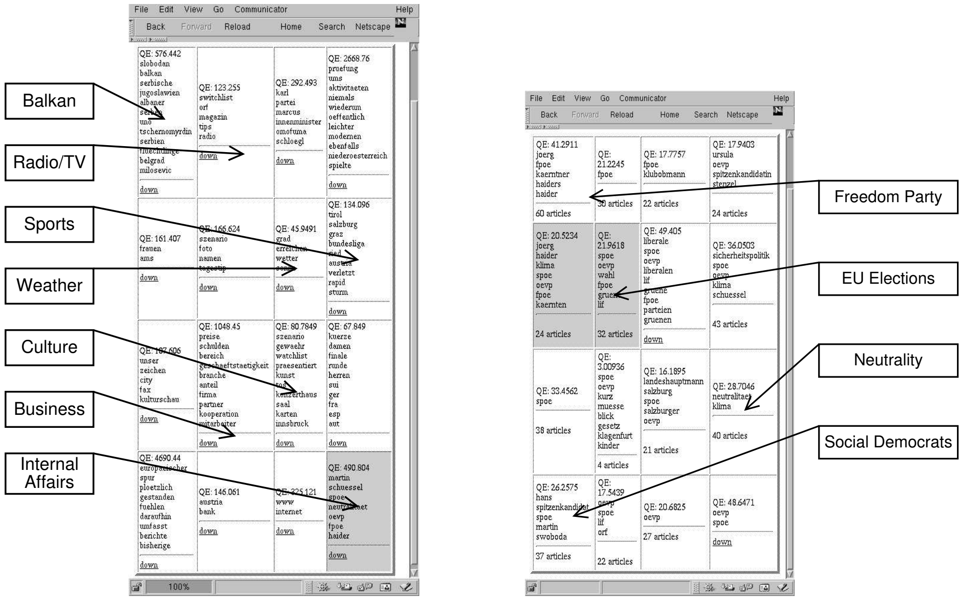


Figure 7: (a) Top-layer map with 4 × 4 units showing general topics, and (b) second-layer map with 4 × 4 units representing national politics.

6 Conclusions

In this paper, we have described the *Growing Hierarchical Self-Organizing Map* (*GHSOM*). The characteristic feature of this novel neural network model is its adaptive architecture which grows during its unsupervised training process to uncover the hierarchical structure of the analyzed data collection. In a nutshell, the *GHSOM* has a layered architecture composed of independent *SOM*-like neural networks. These networks determine their structure during the training process as well.

The major benefits of our *GHSOM* model compared with the standard *SOM* are the following. First, the overall training time is largely reduced since only the necessary number of units are developed to organize the data collection at a certain degree of detail. Second, the *GHSOM* uncovers the hierarchical structure of the data by its very architecture, thus allowing the user to understand and analyze large amounts of data in an explorative way. Third, with the various emerging maps at each layer of the hierarchy being rather small in size, it is much easier for the user to keep an overview of the various clusters. Last, but not least, by ensuring a consistent global orientation of the individual maps in the respective layers, the topological similarities of neighboring maps are preserved. Thus, navigation across map boundaries is facilitated, allowing the exploration of similar clusters that are represented by neighboring branches in the *GHSOM* structure.

We have shown the potential of the *GHSOM* with an information retrieval application, namely the topical clustering of free-form documents. This, by its very nature, is a challenging application for clustering algorithms because of the high-dimensional and noisy feature spaces. The results of the experiments clearly indicated that the *GHSOM* successfully identified the topical clusters of the document collections.

References

[1] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, September 1999,

[2] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biologi-*

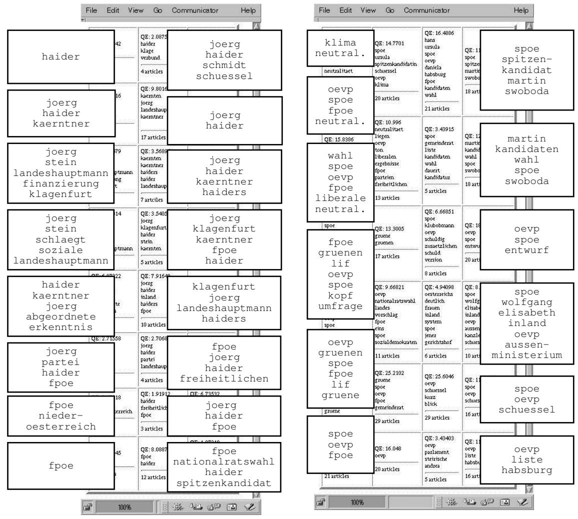


Figure 8: Two neighboring second-layer maps on national politics

cal Cybernetics, vol. 43, pp. 59–69, 1982.

[3] S. Kaski, J. Kangas, and T. Kohonen, “Bibliography of self-organizing map (SOM) papers 1981-1997,” *Neural Computing Surveys*, vol. 1, no. 3&4, pp. 1–176, 1998,

[4] J. Blackmore and R. Miikkulainen, “Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map,” in *Proceedings of the IEEE International Conference on Neural Networks (ICNN’93)*, San Francisco, CA, USA, 1993, vol. 1, pp. 450–455,

[5] B. Fritzke, “Growing Grid – A self-organizing network with constant neighborhood range and adaption strength,” *Neural Processing Letters*, vol. 2, no. 5, pp. 1 – 5, 1995,

[6] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan, “Dynamic self-organizing maps with controlled growth for knowledge discovery,” *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 601–614, May 2000.

[7] T. Kohonen, *Self-Organization and Associative Memory*, Springer Verlag, Berlin, Germany, 3 edition, 1989.

[8] J.A. Kangas, T. Kohonen, and J.T. Laaksonen, “Variants of self-organizing maps,” *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 93–99, March 1990,

[9] T. Kohonen, *Self-organizing maps*, Springer-Verlag, Berlin, 1995.

[10] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela, “Self-organization of a massive document collection,” *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 574–585, May 2000,

[11] X. Lin, “A self-organizing semantic map for information retrieval,” in *Proceedings of the 14. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR91)*, Chicago, IL, October 13 - 16 1991, pp. 262–269, ACM,

[12] D. Merkl, “Text classification with self-organizing maps: Some lessons learned,” *Neurocomputing*, vol. 21, no. 1–3, pp. 61–77, 1998.

[13] D. Merkl and A. Rauber, “Document classification with unsupervised neural networks,” in *Soft Computing in Information Retrieval*, F. Crestani and G. Pasi, Eds., pp. 102–121. Physica Verlag, 2000,

[14] D.G. Roussinov and H. Chen, “Information navigation on the web by clustering and summarizing query results,” *Information Process-*

ing and Management, vol. 37, pp. 789 – 816, 2001.

[15] A. Ultsch, "Self-organizing neural networks for visualization and classification," in *Information and Classification. Concepts, Methods and Application*, O. Opitz, B. Lausen, and R. Klar, Eds., Studies in Classification, Data Analysis, and Knowledge Organization, pp. 307–313. Springer, Dortmund, Germany, April 1-3 1992.

[16] D. Merkl and A. Rauber, "Alternative ways for cluster visualization in self-organizing maps," in *Proceedings of the Workshop on Self-Organizing Maps (WSOM97)*, T. Kohonen, Ed., Espoo, Finland, June 4. - 6. 1997, Helsinki University of Technology, pp. 106–111, HUT,

[17] A. Rauber, "LabelSOM: On the labeling of self-organizing maps," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'99)*, Washington, DC, July 10. - 16. 1999,

[18] D. Merkl and A. Rauber, "Automatic labeling of self-organizing maps for information retrieval," in *Proceedings of the 6. International Conference on Neural Information Processing (ICONIP99)*, Perth, Australia, November 16. - 20. 1999,

[19] S. Santini, "The self-organizing field," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1415–1423, November 1996.

[20] C.M. Bishop, M. Svensen, and C.K.I. Williams, "GTM: The generative topographic mapping," *Neural Computation*, vol. 10, no. 1, pp. 215–235, 1998,

[21] R. Miikkulainen, "Script recognition with hierarchical feature maps," *Connection Science*, vol. 2, pp. 83 – 101, 1990.

[22] P. Koikkalainen and E. Oja, "Self-organizing hierarchical feature maps," in *Proceedings of the International Joint Conference on Neural Networks*, San Diego, CA, 1990, vol. 2, pp. 279 – 284.

[23] P. Koikkalainen, "Fast deterministic self-organizing maps," in *Proceedings of the International Conference on Artificial Neural Networks*, Paris, France, 1995, vol. 2, pp. 63 – 68.

[24] H.-U. Bauer and T. Villmann, "Growing a hypercubical output space in a self-organizing feature map," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 226 – 233, 1997.

[25] M. Dittenbach, D. Merkl, and A. Rauber, "The growing hierarchical self-organizing map," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, S. Amari, C. L. Giles, M. Gori, and V. Puri, Eds., Como, Italy, July 24-27 2000, vol. VI, pp. 15 – 19, IEEE Computer Society,

[26] M. Dittenbach, A. Rauber, and D. Merkl, "Recent advances with the growing hierarchical self-organizing map," in *Proceedings of the 3rd Workshop on Self-Organizing Maps*, N. Allinson, H. Yin, L. Allinson, and J. Slack, Eds., Lincoln, England, June 13-15 2001, Advances in Self-Organizing Maps, pp. 140–145, Springer,

[27] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[28] H.R. Turtle and W.B. Croft, "A comparison of text retrieval models," *Computer Journal*, vol. 35, no. 3, pp. 279–290, 1992.

[29] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, Reading, MA, 1989.

[30] A. Rauber and D. Merkl, "Using self-organizing maps to organize document collections and to characterize subject matters: How to make a map tell the news of the world,"

in *Proceedings of the 10. International Conference on Database and Expert Systems Applications (DEXA99)*, T. Bench-Capon, G. Soda, and A.M. Tjoa, Eds., Florence, Italy, September 1. - 3. 1999, number LNCS 1677 in Lecture Notes in Computer Science, pp. 302–311, Springer,

Biographies

- **Andreas Rauber**, (S'93-M'01) received the M.Sc. and Ph.D. degrees in computer science from the Vienna University of Technology, Vienna, Austria, in 1997 and 2000, respectively.

  Since 1997, he has been a member of the Academic Faculty at the Department of Software Technology at the Vienna University of Technology. In 2002 he held an ERCIM Research Fellowship at the Italian National Research Council (CNR) in Pisa, Italy. He is currently an ERCIM Research Fellow at INRIA, Paris, France. He has published more than 30 papers in refereed journals and international conferences. His current research interests include neural computation, digital libraries, and information visualization.

  Dr. Rauber received the OeGAI award of the Austrian Society for Artificial Intelligence in 1998.

- **Dieter Merkl** received the Diploma and Doctoral degrees in social and economic sciences from the University of Vienna, Vienna, Austria, in 1989 and 1995, respectively.

  He is Associate Professor with the Department of Software Technology, Vienna University of Technology, Vienna, Austria. From 1990 to 1994 he held a research position at the University of Vienna. Since 1995, he has been a member of the Academic Faculty at Vienna University of Technology. During 1997, he was visiting research fellow with the Department of Computer Science, Royal Melbourne Institute of Technology, Melbourne, Australia. He has published more than 80 articles in refereed journals and international conferences. His current research interests include neural computation, information retrieval, and software engineering.

  Dr. Merkl is an affiliate member of the IEEE Computer Society.

- **Michael Dittenbach** received the Diploma degree in computer science from the Vienna University of Technology, Vienna, Austria.

  He is currently a Research Assistant with the E-Commerce Competence Center EC3, Adaptive Multilingual Interfaces Grooup, and a Junior Researcher with the Department of Software Technology, Vienna University of Technology. He has published several papers at international conferences. Currently, his main research interests include natural language processing, cross-language information retrieval, text mining, digital libraries, and neurocomputing.

Erratum

This article appeared in the IEEE Transactions on Neural Networks, Vol. 13, No. 6, pp. 1331-1341, November 2002.

Inadvertently, two corrections were not included in the final release, resulting in an erroneous version of our article being published. Since both the print as well as the on-line version currently contain these errors, we herewith provide the correct version of the article based on a pre-print, together with pointers to the two errors found in the article, which are:

1. **Section IV-B, page 1334, column 2, sentence immediately preceeding expression 9**:
   The erroneous version reads: "*Then, the error unit e is determined as the unit with the mqe as follows:*"
   It should rather read: "*Then, the error unit e*

is determined as the unit with the maximum qe as follows:"

Comment: "*mqe*" stands for the "*mean quantization error*" or "*mean qe*" in short. However, the error unit is the one exhibiting the "*maximum*" quantization error, rather than the mean. Obviously, in an eager quest to use the shorter abbreviations rather than outspelled longer versions, it was decided to use the abbreviation defined for the mean qe.

2. **Section IV-B, page 1335, column 2, reference to Figure 3**:

   Only half of *Figure 3* was printed in the page proofs. Rather than correcting, Figure 3 was omitted in the final version, resulting in Figure 2 on page 1335 being followed by Figure 4 on page 1337.