Technical Report OeFAI-TR-2003-09, extended version published in
Intelligent Data Analysis Journal.

A New Approach to Hierarchical Clustering and Structuring of Data with Self-Organizing Maps

Elias Pampalk[1,*], Gerhard Widmer[1,2], and Alvin Chan[3]

[1]Austrian Research Institute for Artificial Intelligence (OeFAI)
Schottengasse 3, A-1010 Vienna, Austria
{elias, gerhard}@oefai.at

[2] Department of Medical Cybernetics and Artificial Intelligence
University of Vienna

[3]DSO National Laboratories
20 Science Park Drive, Singapore 118230
ctuckwai@dso.org.sg

Abstract

The *Self-Organizing Map* (SOM) is a powerful tool for exploratory data analysis which has been employed in a wide range of data mining applications. We present a novel approach to reveal the inherent hierarchical structure of data using multiple SOMs together with heuristics which optimize the stability. In particular, we address shortcomings of the *Growing Hierarchical Self-Organizing Map* (GHSOM) regarding the decision which areas in the hierarchical structure need to be represented by a finer granularity and which areas do not. We introduce the *Tension and Mapping Ratio* extension to exploit specific characteristics of the SOM based on the topology preservation. As a main result, in contrast to the GHSOM, the inherent hierarchical structure of the data is revealed without requiring the user to define a threshold parameter which controls the map sizes of the individual SOMs. We evaluate our approach using data from real-world data mining projects in the music domain.

Keywords
Exploratory Data Analysis, Growing Hierarchical Self-Organizing Maps, Tension and Mapping Ratio

[*] Corresponding Author
Tel.: +43-1-5336112-21
Fax: +43-1-5336112-77

1 Introduction

With the increasing amount of data available digitally the potential of intelligent data analysis techniques has risen. The field of applications covers simple data surveys as well as sophisticated summaries for decision support. One important aspect of data analysis is to understand the inherent structure and organization of the data. Furthermore, algorithms which reveal the data structure are valuable tools in the automatized organization of digital libraries containing, for example, text-documents or pieces of music. One prominent algorithm with such capabilities is the *Self-Organizing Map* (SOM) [16] which is frequently employed in the field of data exploration.

The SOM maps high-dimensional data to a 2-dimensional map display in such a way that similar data items are located close to each other on the map. Two of the main advantages of the SOM are the computational efficiency and the intuitive way the results are presented to the user. However, the basic SOM lacks the ability to extract the hierarchical structure of the data. This has been addressed for example in [31] where the results of the SOM were used as input for hierarchical agglomerative clustering algorithms. Other alternatives include, for example, the hierarchical SOMs [18, 19, 24]. A top-layer SOM organizes the data into a few categories which are further refined on SOMs in sub-layers. However, the user is required to predefine the granularity of the individual SOMs as well as the overall depth of the structure.

The *Growing Hierarchical Self-Organizing Map* (GHSOM) [6] addresses this problem through a flexible structure which grows to fit the data. A quality measure based on the variance of the data together with threshold parameters are used to decide which granularity is appropriate for a specific SOM, and which areas of the SOM are promising candidates for further hierarchical expansion.

In this paper we identify shortcomings of the GHSOM using illustrative examples and present a new extension to overcome these. The *Tension and Mapping Ratio* (TMR) extension is based on two new measures which analyze the topological and cluster stability of the SOM. The result is an algorithm with improved decisions regarding the necessary granularity to represent the inherent hierarchical structure of the data and requires the user to adjust fewer parameters.

In the following two sections the SOM and the GHSOM algorithms are reviewed. In Section 4 the TMR extension is introduced. In Section 5 we apply the GHSOM-TMR to two real-world datasets both from the music domain and evaluate the results. Conclusions are drawn in Section 6.

2 Self-Organizing Maps

The Self-Organizing Map (SOM), as proposed in [13] and described thoroughly in [16], is one of the most distinguished unsupervised artificial neural network models. The SOM maps high-dimensional input data to a usually 2-dimensional map display in such a way that similar data items are located close to each other on the map. The SOM consists of an ordered set of units i, which are often arranged on a rectangular grid. Each unit i is assigned a model vector $\mathbf{m}_i$ in the high-dimensional data space. A data item $\mathbf{x}_k$ is mapped to the best matching map unit c_k with the smallest distance between $\mathbf{m}_i$ and $\mathbf{x}_k$ defined by

$$c_k = \arg\min_i \|\mathbf{x}_k - \mathbf{m}_i\| . \tag{1}$$

Two basic training methods exist to adapt the model vectors to fit the data. The SOM algorithm in its original form is an online algorithm which takes one data item per iteration as input and makes small adaptations sequentially. On the other hand, the batch-SOM [14] algorithm takes all data items as input per iteration and thus makes bigger adaptations. Here we will review the batch-SOM algorithm. A detailed discussion of the online algorithm can be found in [16].

The model vectors are adapted iteratively in two steps. In the first step c_k is calculated for all $\mathbf{x}_k$ according to Equation 1. In the second step each model vector is adapted to better fit the data it represents. To ensure that each unit i represents similar data items as its neighbors, the model vector $\mathbf{m}_i$ is adapted not only according to the assigned data items but also with respect to those assigned to the units in the neighborhood. The neighborhood relationship between two units i and j is usually defined by a Gaussian-like function $h_{ij} = \exp(-d_{ij}^2/2r_t^2)$, where d_{ij} denotes the distance between the units i and j on the map, and r_t denotes the neighborhood radius which is set to decrease with each iteration t. The second step can be formulated as

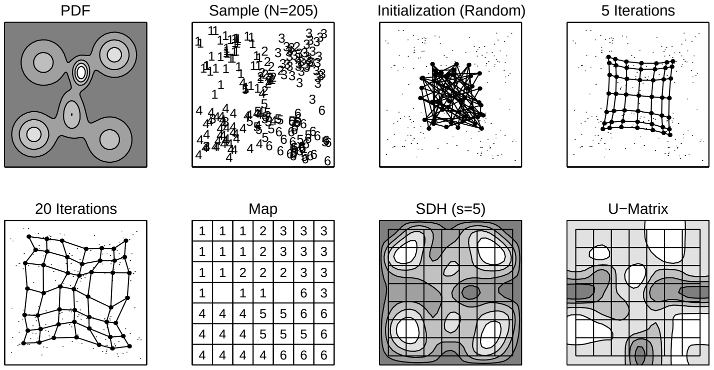
$$\mathbf{m}_i^* = \frac{\sum_k h_{ic_k}\mathbf{x}_k}{\sum_k h_{ic_k}}, \tag{2}$$

where $\mathbf{m}_i^*$ is the updated model vector. An efficient implementation is to first calculate the mean $\bar{\mathbf{x}}_i$ of the $\mathbf{x}_k$ which are mapped to $\mathbf{m}_i$ and then calculate $\mathbf{m}_i^*$ as follows

$$\mathbf{m}_i^* = \frac{\sum_j n_j h_{ij}\bar{\mathbf{x}}_j}{\sum_j n_j h_{ij}}, \tag{3}$$

where n_j denotes the number of items mapped to unit j, which is basically a weighted sum of all data vectors in the neighborhood parameterized by the radius r_t.

The SOM algorithm and some of its properties are illustrated in Figure 1. The dataset was generated by a mixture of 6 Gaussians with different covariances and priors. The model vectors of the SOM can be initialized randomly or based on more advanced techniques such as Principal Component Analysis (PCA) [11]. Figure 1 shows that after 5 iterations with a rather large neighborhood radius the SOM has roughly adapted to the structure of the data. Note that the model vectors are located around the center of the dataset because each model vector is representing many points due to the large neighborhood radius. After 20 iterations with a smaller neighborhood radius the SOM has spread out over the whole set and has adapted to the fine structures in the data. The resulting map is a non-linear projection of the data space where the overall order is preserved. One characteristic of the SOM is that not all units represent the same number of items. For example, some units represent no items, and thus are not labeled with a number. These interpolating units are a result of the neighborhood constraints. The *Smoothed Data Histogram* (SDH) visualization [26] utilizes this information to visualize the cluster structure.



1	1	1	2	3	3	3
1	1	1	2	3	3	3
1	1	2	2	2	3	3
1		1	1		6	3
4	4	4	5	5	6	6
4	4	4	5	5	5	6
4	4	4	4	6	6	6

Figure 1: SOM training, mapping and visualization. The plots from left to right and up to down depict: (1) the probability density function (PDF) of the dataset; (2) the sample drawn from this density, where each data point is labeled with a number from 1 to 6 indicating the Gaussian it was generated by; (3) the model vectors of a 7×7 SOM randomly initialized in the data space, where dots depict the model vectors and the lines connect immediate neighbors on the grid; (4) the model vectors after 5 iterations of the batch SOM algorithm and (5) after 20 iterations; (6) the map units in the visualization space, where each unit is labeled with the number corresponding to the Gaussian which is most frequently represented by this unit; (7) the Smoothed Data Histogram (SDH) and (8) the U-matrix visualization of the map.

Note that all 6 clusters and their relation to each other can clearly be identified. Another important characteristic of the SOM is that areas in the data space with a high density are approximated by more model vectors than sparse areas. Thus, the distances between model vectors are large in sparse areas and small in areas with a high density, which is visualized by the *U-matrix* [30]. The results of the U-matrix correspond roughly to those of the SDH. Clearly visible are the large distances between the clusters 1 and 4, 3 and 6, and between the clusters 4 and 6.

The SOM mapping for a less abstract test set is depicted in Figure 2. The animal dataset consists of 16 animals which are described by several attributes, such as how many legs they have, if they can swim, hunt, and so forth [16]. The map and the corresponding SDH visualization reveal that the dataset consists of three clusters, one larger one on the right representing mammals with a gradual separation of bigger ones in the upper regions and smaller ones below. On the left the birds are separated into two clusters.

The SOM has been employed in a wide range of applications, ranging from financial data analysis, via medical data analysis, to time series prediction, industrial control, and many more [5, 16]. It basically offers itself to the organization and interactive exploration of high-dimensional data spaces. One of its most prominent application areas is the organization of large text archives [17, 22, 29] which scale up to millions of documents [15, 12].

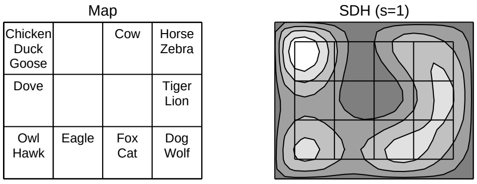


Figure 2: 4×3 SOM trained with animal dataset.

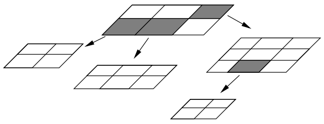


Figure 3: Example for a GHSOM architecture consisting of 3 layers. The shaded units are expanded on the next level.

3 Growing Hierarchical Self-Organizing Maps

The *Growing Hierarchical Self-Organizing Map* (GHSOM) [6, 7, 8, 23] enhances the capabilities of the basic SOM in two ways. The first is to use an incrementally growing version of the SOM, which does not require the user to directly specify the size of the map before hand, the second enhancement is the ability to adapt to hierarchical structures in the data. Both issues have been addressed separately by means of adaptive architectures, e.g. the *Growing Grid* [9], or the *Growing SOM* [1], or hierarchies of independent SOMs, e.g., the *Hierarchical Feature Map* [24] or the *Tree Structured SOM* [18, 19].

The key idea of the GHSOM is to use a hierarchical structure of multiple layers where each layer consists of a number of independent SOMs. One SOM is used at the first layer of the hierarchy. For every unit in this map a SOM might be added to the next layer of the hierarchy. This principle is repeated with the third and any further layers of the GHSOM. An example of a GHSOM architecture with 3 layers is depicted in Figure 3.

Table 1 gives an outline of the basic GHSOM algorithm. The training process starts with a small map of 2×2 units in the first layer which is organized according to the standard SOM training algorithm. After the model vectors have converged some decisions are made. Namely, (2A) if a new row or column should be inserted to improve the quality of the representation on this level, (2B) if the quality of the representation should be improved on the next hierarchical level by expanding one or more units, or (2C) if the SOM represents the data well in its current form. These decisions are the core of the GHSOM algorithm as they control the hierarchical structure and the shapes of the individual SOMs. In case of 2A a new row or column is inserted and the training process is repeated. In case of 2B for each map unit which needs to be expanded a new 2×2 SOM is created in such a way that the overall orientation is preserved. The training process is repeated for each of these new maps. Otherwise, (2C) if neither decisions 2A nor 2B are made no additional

0	Initialize first layer SOM, → 1.
1	Train SOM, → 2.
2	Decide: A – Insert new row or column, → 3. B – Expand units on next hierarchical level, → 4. C – SOM represents data well, → end.
3	Insert new row or column, → 1.
4	For each unit to expand: Create new 2×2 SOM, preserve orientation, → 1 (using only the subset of the data mapped to the respective unit).

Table 1: Outline of the GHSOM algorithm.

steps are necessary.

2A – Should a new row or column be inserted?

The intention is to increase the map size until all data items are represented well. This is measured based on the quantization errors, which describe how well a data item is represented by its best matching unit.

The first step is to calculate the mean quantization error of each unit i (MQE_i),

$$\mathrm{MQE}_i = \frac{1}{|\mathrm{U}_i|} \sum_{k \in \mathrm{U}_i} \|\mathbf{x}_k - \mathbf{m}_i\| \,, \quad \mathrm{U}_i = \{k \,|\, c_k = i\}. \tag{4}$$

Then a summary which describes how well all units together represent the data is calculated as the mean of all MQE_i (MMQE). Alternatively, instead of using the mean the sum of all MQE_i (SMQE) can be used [3]. The MMQE (or SMQE) is then compared to the mean quantization error of all data items with respect to a virtual unit with its model vector set to the mean of the dataset. We will refer to the mean quantization error of the virtual unit as MQE_0. If the following inequality is fulfilled a new row or column of map units is inserted in the SOM,

$$\mathrm{MMQE} > \tau_1 \mathrm{MQE}_0, \tag{5}$$

where τ_1 is a user defined parameter.

Once the decision is made to insert new units the remaining question is where to do so. The GHSOM uses a slightly modified version of the Growing Grid heuristic. First the unit i with the largest MQE_i is defined to be the *error unit*. Then the most dissimilar adjacent neighbor, i.e., the unit with the largest distance in respect to the model vector, is selected and a new row or column is inserted between these.

2B – Should map units be expanded on the next hierarchical level?

If the Inequality 5 is not fulfilled, the next decision to be made is whether some units should be expanded on the next hierarchical level or not. This decision is not as critical as the previous since it is possible to expand additional units after the training process, or to remove a subbranch without influencing other branches of the GHSOM architecture.

The original GHSOM algorithm checks if the following inequality is fulfilled to decide whether unit i should be expanded,

$$\mathrm{MQE}_i > \tau_2 \mathrm{MQE}_0^*, \tag{6}$$

where MQE_0^* is the mean quantization error of the whole dataset[1] with respect to the virtual unit located in center of the whole dataset. Again, τ_2 is a user defined parameter. To simplify matters, instead of using Inequality 6, it is possible to use $|\mathrm{U}_i| > n$ as measure [3]. For example, $n = 10$ would mean that all units with more than 10 items are expanded.

Note that this will usually not lead to a balanced hierarchy. The depth of the hierarchy in different branches will rather reflect the diversity in the distribution of the dataset which should be expected in real-world data collections.

Parameters

The two main parameters of the original GHSOM algorithm are τ_1 from Inequality 5 and τ_2 from Inequality 6. Generally, the values for τ_1 and τ_2 are chosen such that $1 > \tau_1 \gg \tau_2 > 0$. The choice of the parameter τ_2 is not critical as it is possible to set it to values smaller than necessary and to decide after the training process how many layers are sufficient. On the other hand, τ_1 leads to decisions which cannot be undone at later stages without retraining the GHSOM. Setting τ_1 to small values will lead to small maps and a deep hierarchy, while large values will result in large maps with a flat hierarchical structure. Usually the user will train and manually evaluate several maps until an appropriate value for τ_1 is estimated.

Orientation

Without further precautions no relationship between different sub-branches of the hierarchical structure would exist, other than the relationship defined by the parent map which they have in common. For example, in Figure 3 the maps of the second layer could have an arbitrary orientation (i.e., they could be rotated in any direction), and thus would not allow to draw conclusions on the similarity of the 3rd layer map to any of the other two 2nd layer maps, other than the relationships represented by the first layer map. A simple solution to this is to enforce a specific orientation on the sub-layer SOMs through initializing the model vectors of the 2×2 SOMs in such a way that they correspond to the orientation of the parent map. For example, the 4 model vectors could be initialized with the means of the model vector of the parent unit and each of its 4 immediate neighbors. For units located at the border of the SOM we use the existing units to interpolate a virtual unit based on the assumption that the map can locally be approximated by a linear hyper-plane [3]. For example, if the unit i is to be expanded which is located on the left border, then the unit to its right j is used to create a virtual unit $v_{ij'}$ beyond the left border $\mathbf{m}_{ij'} = \mathbf{m}_i + (\mathbf{m}_i - \mathbf{m}_j)$.

Example

Figure 4 depicts a flat representation of the GHSOM trained with the same animal

[1] In contrast to MQE_0, which is the mean quantization error of the data items in the respective sub-branch of the GHSOM.

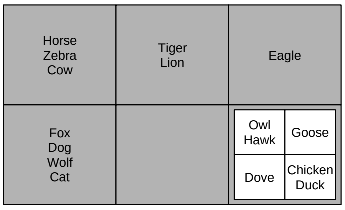


Figure 4: GHSOM trained with the animal dataset ($\tau_1 = 0.5$ using SMQE).

dataset illustrated in Figure 2. The dataset is split into 6 clusters on the first level separating the mammals into three groups of large herbivore, medium predators, and large predators on the left, and birds on the right. The cluster representing all birds but the Eagle is further refined on the second level. Note that Owl and Hawk are located closer to the Eagle than for example Dove and Chicken, which is a result of initializing the orientation of the sub-layer according to the orientation of the parent-layer.

4 Tension and Mapping Ratio Extension

The *Tension and Mapping Ration* (TMR) extension of the basic GHSOM algorithm introduced here addresses in particular Decision 2A: Whether new units should be inserted and in which row or column they should be inserted. As pointed out in the previous section this is the most critical part of the GHSOM algorithm.

The TMR extension has two main effects on the GHSOM algorithm. First of all, the quality of the individual SOMs is improved by inserting new units where they are really needed. Furthermore, the appropriate map sizes are determined automatically without requiring the user to guess a suitable value for the τ_1 parameter which controls Decision 2A.

Decision 2A is a model selection problem. In general selecting the model which best fits the data and avoids overfitting to noise in the data is a unsolved problem. Several different techniques for model selection exist and have been applied to complexity selection of SOMs, e.g. [21]. However, the problem of Decision 2A is more subtle than it might appear on first sight since the aim is not necessarily to have a low quantization error on each layer, or to minimize, e.g., the Minimum Message Length [32] necessary to decode the SOM and the data of a particular layer. Instead, the goal is to hierarchically cluster the data with clearly separated clusters on the top-layers and low quantization errors on the lower levels. Furthermore, for computational efficiency it is of interest to take special characteristics of the SOM into account, such as the neighborhood relationships between units. The TMR extension addresses these issues.

The two principles of the TMR are straightforward. The first principle is to ensure stable mappings between two neighboring map units i and j. The stability is measured by inserting a virtual unit v_{ij} between the units i and j with its model vector $\mathbf{m}_{ij} = (\mathbf{m}_i + \mathbf{m}_j)/2$. If a large number of items which are mapped to the units i and j would better be represented by v_{ij} then the arrangement is instable, on the other hand, if only a very small ratio of items would prefer v_{ij} then the relationship is stable. We refer to this ratio as *Mapping Ratio* (MR).

The second principle is to ensure a stable structure of the neighborhood topology. The topology is stable if there is little *Tension* (T), i.e., if the distances between all neighbors are about the same. In particular, we calculate the Tension to avoid a disproportional ratio between the average pairwise distances between all adjacent rows and all adjacent columns.

Given a trained SOM the following 7 steps are executed before making the decision where to insert units is finalized.

Step 1, the 4 virtual units for each unit i are defined. For border units the same approach introduced for preserving the orientation of sub-layers is applied, i.e., the virtual units are linearly interpolated based on the assumption that the map can locally be approximated by hyper-planes.

Step 2, for all $\mathbf{x}_k \in \mathrm{U}_i$ the best matching unit is selected from the 4 virtual units and the original unit. The number of items which are best represented by the original unit i is denoted as $n_{i(i)}$, the number of units mapped to the virtual unit v_{ij} as $n_{i(v_{ij})}$. Note that this information can later be reused in case unit i is expanded on the next layer, or in case a new row or column is inserted next to the unit.

Step 3, the Mapping Ratio MR_{ij} of each virtual unit v_{ij} is calculated as

$$\mathrm{MR}_{ij} = \frac{n_{i(v_{ij})}}{n_{i(i)}} + \frac{n_{j(v_{ij})}}{n_{j(j)}}. \tag{7}$$

Virtual units located beyond the border are calculated as $\mathrm{MR}_{ij'} = n_{i(v_{ij'})}/n_{i(i)}$, where i denotes a border unit. Higher MR values correspond to less stable mappings since a relatively large amount of items is mapped between the map units.

Step 4, the decision if the growth process should be continued or not is made. The sum of all MR_{ij} (SMR) is calculated. If the map is bigger than 2×2 and the SMR value has increased compared to the SMR of the previous, smaller map, then the current map is rejected and the growth process is terminated. Otherwise, the process is continued.

Step 5, for each row and column of virtual units z the average Mapping Ratio AMR_z is calculated as the median of all MR_{ij} in the respective row or column.

Step 6, the Tension is calculated. First, the Average Distance AD_z is calculated as the median of the distances between the neighboring model vectors which define the row or column of the respective virtual units, i.e, for all v_{ij} belonging to z, $\mathrm{AD}_z = \mathrm{median}(\|\mathbf{m}_i - \mathbf{m}_j\|)$. Furthermore we calculate AD_r as the median of all AD_z where z represents a row, and AD_c as median of all AD_z where z represents a column. Then the Tension T_z for each z representing a row is calculated as $\mathrm{T}_z = \mathrm{AD}_z/\mathrm{AD}_c$ and analogous for each z representing a column $\mathrm{T}_z = \mathrm{AD}_z/\mathrm{AD}_r$. For example, if a map with an equal number of rows and columns is trained on a 2-dimensional dataset with a high variance in the direction fitted by rows and a low variance in the direction fitted by the columns, then AD_r would be higher than AD_c and T_z for a row would be higher than for T_z for a columns. The Tension for rows and columns of virtual units beyond the borders of the map is set to the value of the respective row or column just before the border. Note that we use the median preferably because it is more robust towards outliers, however, using the mean will yield the same results for most applications.

Step 7, we multiply the Tension T_z with the respective average Mapping Ratio AMR_z and select the row or column of virtual units z with the highest value as location for new units. The reason why we combine the Tension and Mapping Ratio by multiplication is that both become scale independent, higher values of one of the two will increase the outcome, and vice versa. Furthermore, both the Mapping Ratio and the Tension are considered of equal importance for the stability of a SOM.

4.1 Illustrations

Figure 5 illustrates an example where the SOM mapping is unstable. The 2×2 SOM is trained on the dataset consisting of 6 Gaussian clusters. Each of the 4 units represents 1.5 clusters, in particular the 2 units in the upper row share 3 clusters and the units below as well. Thus inserting a new column between these units would solve the problem. However, the GHSOM bases its decision on the error unit, which is the unit in the upper-left, and on the most dissimilar neighbor, which results in the insertion of a new row. In the fine tuning phase of the SOM algorithm each model vector is adapted to best fit the items it represents which leads to the suboptimal topology structure depicted in the upper-right subplot of Figure 5. On the other hand, the TMR extension correctly inserts a new column of units which results in the stable topology illustrated in the lower-right subplot.

Furthermore, using the TMR extension it is possible to analyze the global effects an insertion has while the original GHSOM draws its conclusions based merely on local observations. An illustrative example where the global effects need to be considered is depicted in Figure 6. The data is generated by 12 Gaussians with equal covariances which are arranged on a rectangular 6×2 grid with a local disruption in the center. The GHSOM identifies the error unit and its most dissimilar neighbor, in this case the most dissimilar neighbor has the second largest mean quantization error. Both units are very unstable as their model vectors lie in between clusters. From a local point of view inserting a new unit between them would increase the cluster quality. However, inserting a new row of units will effect the structure of 8 other units which represent their items very well. The result is depicted in the center-right subplot, where the overall topology does not correspond well with the data. On the other hand, when applying the TMR extension the effects on all units in the row or column are analyzed. Specifically, the average benefit of the new units inserted in a particular row or column is calculated and the highest average benefit is selected. In this example, where obviously the mapping ratio is largest between the two center units the TMR extension decides to add a new column instead since all other units would not benefit from a new row.

Another advantage of the TMR extension is the improved decision regarding the required number of units for an individual SOM. To illustrate this we use two 2-dimensional datasets, both generated by Gaussians with equal covariances located on a rectangular grid so that two immediate neighbors have unit distance. The

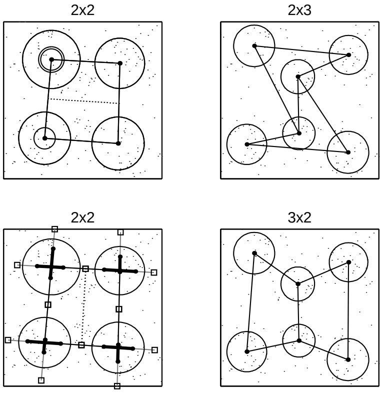


Figure 5: A wrong decision on the local level. The GHSOM is trained with the dataset presented in Figure 1. As in Figure 1 the model vectors which are adjacent neighbors are connected by lines. The larger circles around each model vector represent the mean quantization error. The first row represents the original GHSOM, the second row the TMR extension. In the upper-left subplot the error unit is marked by a circle with double lines (upper-left unit), the most distant neighbor with a simple circle (lower-left). The dotted line indicates where a new row of units is inserted, the second subplot in the first row presents the results after the GHSOM has been trained with the new row. The lower-left subplot depicts the factors influencing the decision of the TMR extension. The locations of the virtual units are marked as squares, the thick lines illustrate the ratio of items mapped to the respective virtual unit (MR). For example, many of the data items mapped to the unit in the upper-left would better fit the virtual unit located above. On the other hand, for example, hardly any of the items mapped to the unit in the upper-right would better fit the virtual unit located below it. Based on the TMR a new column is inserted, which is marked by the dotted line, and results in the map in the lower-right subplot.

first dataset was generated by 4 Gaussians, the second by 16 Gaussians. Thus, the best hierarchical SOM model would have 4 and 16 units at this level of detail while any number beyond could be interpreted as overfitting. However, if each of the Gaussian clusters is generated by a mixture of other Gaussian distributions, the structure of these would need to be represented on the next hierarchical level. Figures 7 and 8 show the behavior of four different quality measures on these two datasets respectively. We compare the sum of all Mapping Ratios (SMR) to the MMQE, SMQE, and to the *Davies-Bouldin* (DB) cluster validation index [4]. The curves depict the measured error for a 2×2 SOM initialized with PCA into which subsequently new columns or rows of units are inserted based on the TMR extension.

The DB index is used to determine the optimal number of clusters (for related indexes see, e.g., [2]). Given the number of map units C and using the mean quantization errors the DB index can be calculated as

$$\mathrm{DB} = \frac{1}{C} \sum_{i=1}^{C} \max_{i,j \neq i} \left(\frac{\mathrm{MQE}_i + \mathrm{MQE}_j}{\|\mathbf{m}_i - \mathbf{m}_j\|} \right). \tag{8}$$

For example, in the Figures 5 and 6, the Davies-Bouldin (DB) index would be large

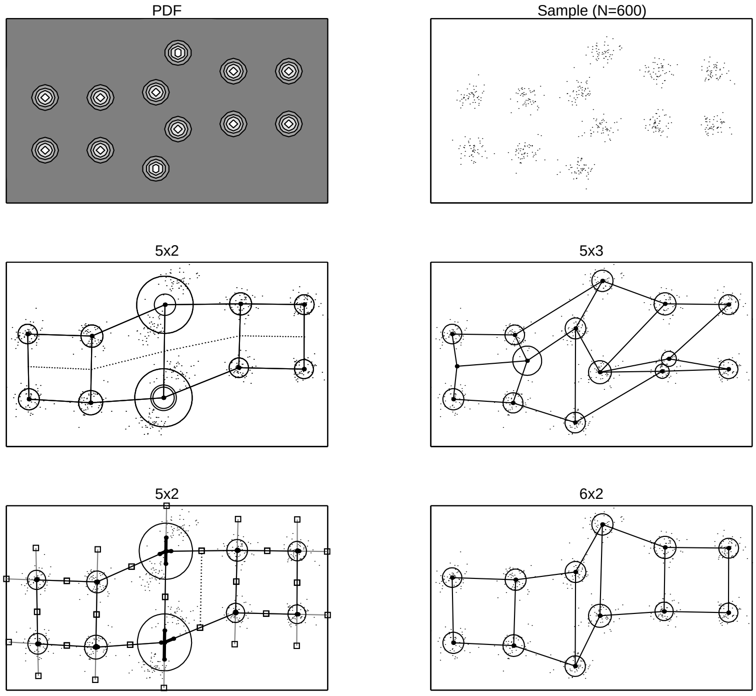


Figure 6: A wrong decision on the global level. The GHSOM is confronted with a situation in which the map generally fits the data quite well, except for a local disturbance. The first row depicts the probability distribution and the sample drawn from it. The second row presents the original GHSOM results and the third row the results of the TMR extension analogous to Figure 5.

where the circles, which indicate the MQE, are close to each other or even overlap, and small if the circles are separated clearly.

Note, that neither the DB index nor the SMR can be calculated for a single cluster. This limitation does not apply to the MMQE or SMQE. However, we argue that comparing the SOM which is under consideration with a bigger SOM leads to more reliable decisions whether the map size should be increased than comparing it with a one unit SOM (MQE_0).

In Figure 7, the MMQE error depicted is drastically reduced when 4 units are used instead of 1. Knowing that the correct number of clusters is 4 this could lead to the false assumption that it is sufficient to find the number of units after which the MMQE does not decrease significantly. The reason why this assumption is wrong is illustrated in Figure 8. Although there are 4 times as many clusters present the run of the curve is basically the same. Hence, trying to guess the optimal number of cluster from the MMQE is quite impossible.

On the other hand, the SMQE curve reflects the actual cluster structure better. However, to obtain the optimal number of clusters would require $1 \leq \tau_1 \leq 0.38$ (see Equation 5) in the case of 4 clusters, and $0.27 \leq \tau_1 \leq 0.19$ in the case of

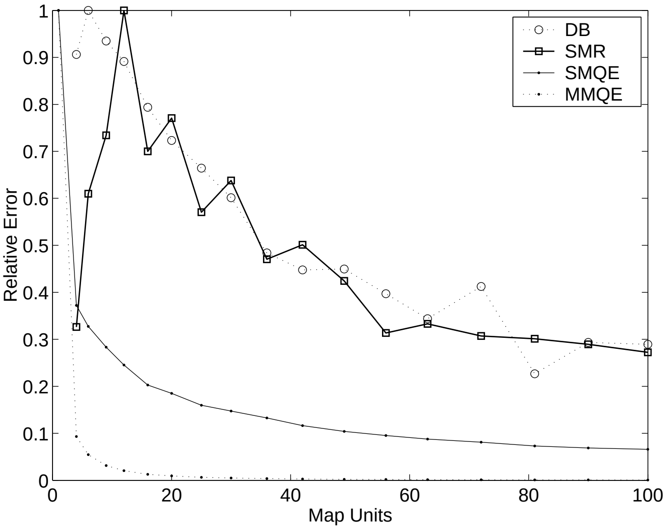


Figure 7: Relative errors for a dataset consisting of 4 clusters. The errors are normalized so that the maximum value equals 1.

16 clusters. Thus, although both datasets are only 2-dimensional and consist of Gaussian clusters with equal covariances, it is not possible to use the same parameter setting for τ_1 to obtain appropriate results.

The DB index responds better to the cluster structure than the SMQE. For example, in the case of 4 clusters, the index indicates that 4 units are a better approximation than 6 or 9 units, which would be sufficient information to terminate the growth process. However, in the case of 16 clusters the DB index does not have a local minimum at 16 units, although the index indicates only slightly better results for 20 units.

In the case of 4 clusters the SMR in the range of 4 to 50 is smallest for 4 (cf. Figure 7). Note however, that if the number of units approaches the number of data items in the dataset, then all 4 error measures will converge towards zero, since all items can be represented without any error. In the case of 16 clusters (cf. Figure 8) the SMR rapidly decreases until the map size reaches 16 units, and then increases again to an error level more than two times higher than the minimum at 16 units. Thus, the SMR outperforms the SMQE and the DB index, while we have shown that the MMQE is an insufficient measure to determine the correct number of clusters.

To round up this section the GHSOM with TMR extension trained with the animals dataset is presented in Figure 9. On the first level the animals are clustered into 3 groups, namely large mammals, medium mammals, and birds. On the second level the large mammals are further distinguished into carnivore and herbivore mammals. The hierarchical clustering result appears to have a slightly better quality than the one obtained from the original GHSOM. Results obtained from real-world data are reported in the next section.

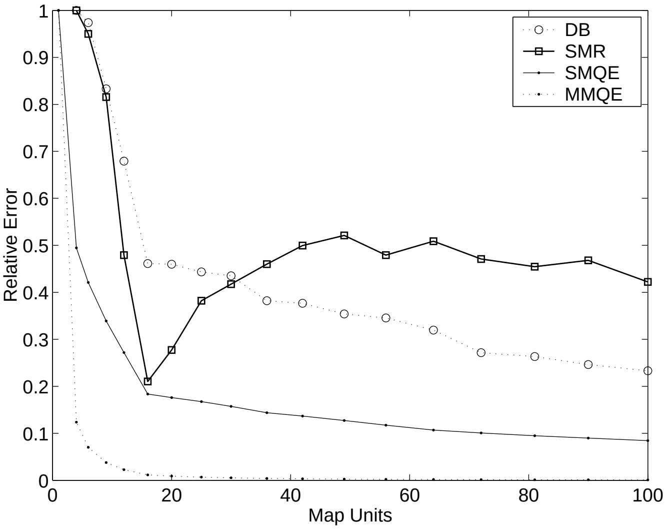


Figure 8: Relative errors for a dataset consisting of 16 clusters. The errors are normalized so that the maximum value equals 1.

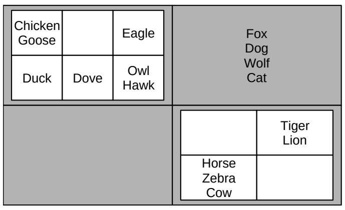


Figure 9: GHSOM-TMR trained with animal dataset.

5 Applications

In this section we will present two applications of the GHSOM with TMR extension. The first application is the content-based organization of a music archive. The second application is the identification of distinctive sequences in multi-variate time-series data representing musical performance strategies.

5.1 Content-Based Organization of Music Archives

The first dataset stems from the project *Islands of Music*[2] whose goal is to create intuitive interfaces to music archives by automatically analyzing, organizing, and visualizing pieces of music based on their perceived acoustic similarities [25, 27]. The pieces are given in compressed-MP3 format and are decoded to the raw audio signal. From the time domain the signals are transformed into the frequency domain where

[2] http://www.oefai.at/˜elias/music

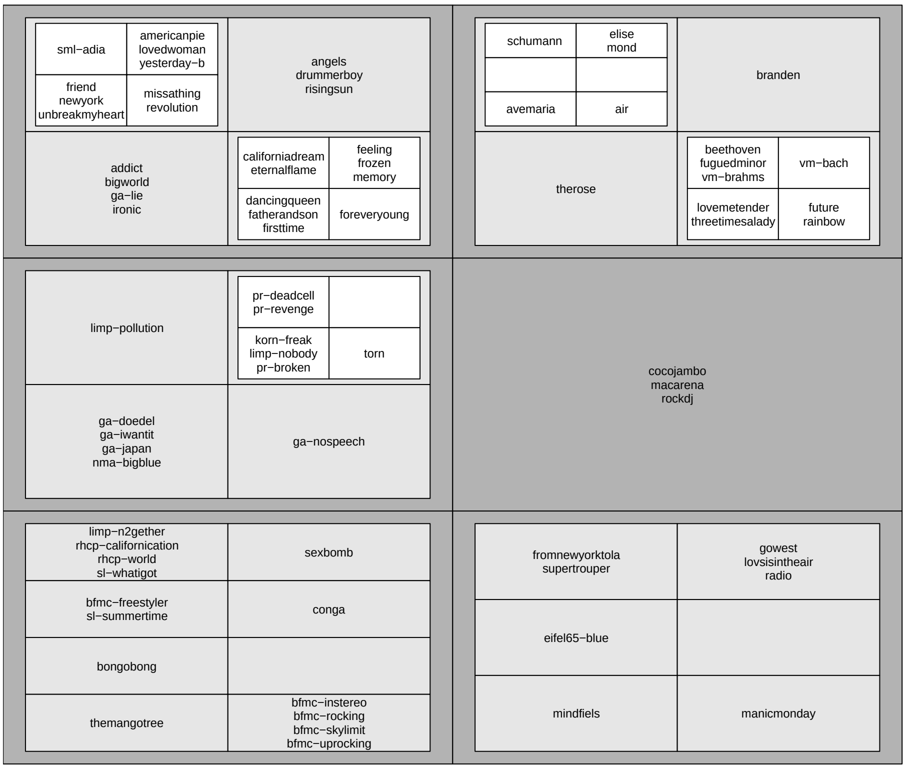


Figure 10: GHSOM of a music archive.

the specific loudness sensation per frequency band is calculated based on psychoacoustic models [35]. The loudness fluctuation of the loudness amplitudes in each band are analyzed using a Fourier transformation. Furthermore, filters are applied to emphasize specific patterns. Finally, each piece is represented by 1200 low-level features describing the dynamic patterns of the loudness in frequency bands. Other than this data no additional information on musical styles, instruments, or musical content is given to the system.

Figure 10 shows the hierarchical organization of a music collection consisting of 77 pieces with a total playing time of about 5 hours. On the first level the training process has resulted in the creation of a 2×3 map, organizing the collection into 6 major styles of music. The upper-right represents mainly classical music, while the lower-left mainly represents a mixture of Hip Hop, Electro, and House and specifically music by *Bomfunk MCs (bfmc)*. The lower-right and center-right represent mainly Disco music such as *Rock DJ* by *Robbie Williams (rockdj)*, *Blue* by *Eiffel 65 (eiffel65-blue)*, or *Super Trouper* by *A-Teens (supertrouper)*. The upper-left represents Pop music while the center-left represents more aggressive Rock music. Please note that the organization does not follow clean ‘conceptual’ genre styles, splitting by definition, e.g. Hip Hop and House, but rather reflects the overall sound similarity.

Most of these 6 first-level categories are further refined on the second level. For example, the upper-right unit representing classical music is divided into 4 further sub-categories. Of these 4 categories the upper-left represents slow and peaceful music, mainly piano pieces such as *Für Elise (elise)* and *Mondscheinsonate 1st movement(mond)* by *Beethoven*, or *Fremde Länder und Menschen* by *Schumann (schumann).* The lower-right represents, for example, pieces by *Vanessa Mae* (vm), which, in this case, are more dynamic interpretations of classical pieces played on the violin. Other pieces located in the cluster are the end credits of the film *Back to the Future III (future)* and the slow love song *The Rose* by *Bette Middler (therose)*, and the *Brandenburg Concertos* by *Bach (branden).* Note that the third level refinement of the upper-left cluster separates the piano pieces from the slow violin piece *Air from orchestra suite #3* by *Bach (air)* and the *Ave Maria* by *Schubert (ave maria).* The 3rd level refinement in the lower-right distinguishes for example the two love songs *Three Times a Lady* by *Lionel Richie (threetimesalady)* and *Love Me Tender* by *Elvis (lovemetender)* from the other orchestra pieces.

Some interesting insights into the music collection which the GHSOM reveals are, for example, that the song *Freestyler* by *Bomfunk MCs* (lower-left) is quite different from the other songs by the same group. *Freestyler* was the groups biggest hit so far and unlike their other songs has been appreciate by a broader audience. Generally, the pieces of one group have similar sound characteristics and thus are located within the same categories. This applies, for example, to the songs of *Guano Apes (ga)* and *Papa Roach (pr)*, which are located in the center-left of the 6 first-level categories together with other aggressive rock songs. However, the song *Living in a Lie* by *Guano Apes (ga-lie)* is another exception (upper-left). Listening to this piece reveals that it is much slower than the other pieces of the group, and that this song which is about the end of a relationship matches very well with, for example, *Addict* by *K's Choice* which is about addiction to drugs.

5.2 Identification of Musical Performance Strategies

The second dataset stems from a large data mining project[3] whose goal is to study fundamental principles of *expressive music performance* [33, 34]. Performances by concert pianists are measured with respect to timing and dynamics (loudness) fluctuations; the goal is to find characteristic patterns that give insight into typical interpretation strategies used by pianists.

The dataset used for this particular experiment consists of performances of a piano piece by Frédéric Chopin (the Etude op.10 no.3), played by 22 skilled pianists [10]. Each performance is characterized by two series of numeric values that represent the measured tempo and loudness, respectively, over the course of the performance. A visualization of one such time series in the form of a trajectory in the two-dimensional tempo-loudness space is shown in Figure 11, with tempo on the vertical axis and dynamics (loudness) on the horizontal axis (see [20] for details of this form of performance representation). The various trajectories were then cut into a total of 184 segments (according to bars) each represented by 80 attributes

[3] http://www.oefai.at/music

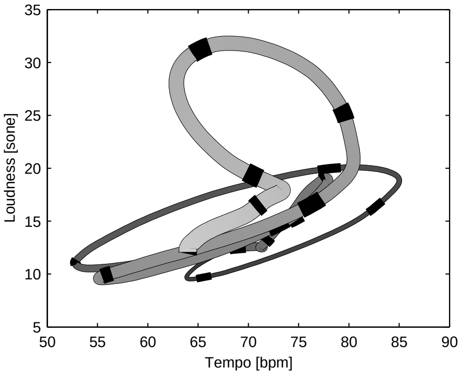


Figure 11: Part of a trajectory corresponding to an expert performance of the Chopin etude op.10/3. The loudness and tempo curves are smoothed. The bar boundaries are indicated through the black sections (this trajectory covers bars 4-19). The time dimension is visualized through the thickness and the shading of the trajectory, the most recent bars are represented by lighter shadings and a thicker line, while bars played in the past are represented by darker shadings and a thinner line.

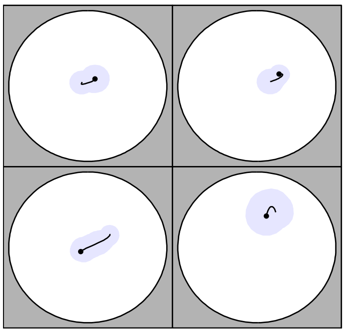


Figure 12: The top-level 2×2 SOM of the GHSOM trained on the music performances.

and subjected to our GHSOM-TMR algorithm. The purpose of the whole procedure is to find out whether there are indeed characteristic and interpretable classes of tempo/dynamics strategies that pianists apply consistently, and to quantify these.

The results are shown in Figures 12 and 13, where the model vectors are visualized. The direction of the trajectory segments in time is indicated by a dot marking the end point. The shadows indicate the variance of the data items mapped to the specific model vectors.

At the top level (Fig. 12), four distinct classes emerge which have a clear musical interpretation: The upper two clusters are typical strategies used by pianists at the beginnings of musical phrases: a speeding up (*accelerando* – movement from left to right in this space) is accompanied by a slight increase in loudness (*crescendo*); the cluster in the lower left represents the opposite tendency of ending a phrase with a slowing down (*ritardando*) and a decrease in loudness; and the cluster in the lower

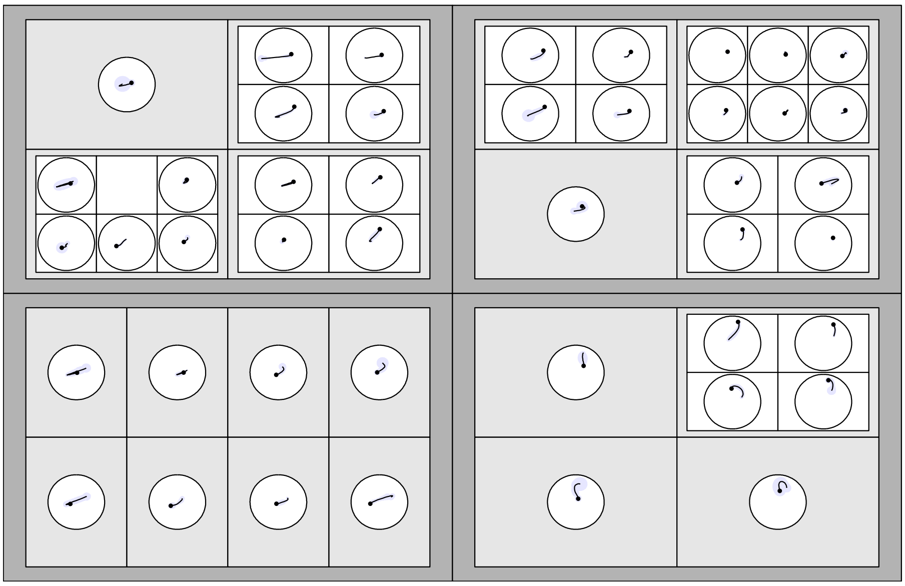


Figure 13: All levels of the GHSOM trained on the music performances.

right shows the typical way of shaping the middle part of a phrase, with a typical loudness maximum around the musical climax (the 'apex') of the phrase. All this conforms well with current musicological theories of expressive performance.

At more detailed levels (Fig. 13), these classes of elementary behavior are broken down into a number of interesting sub-categories. In the 'apex class' (lower right quadrant), a clear distinction between upward (towards the apex) and downward (after the apex) motions becomes visible — the *crescendo* and *decrescendo* parts of a typical phrase arch. Similarly, the 'phrase start' patterns in the upper two quadrants are now discriminated into *accelerando-crescendo* patterns and phases of 'stagnation' where the pianists keep tempo and loudness more or less constant (see the six classes in the upper right corner[4]). Untypical category members are collected in a separate set of subclasses (lower left in upper left quadrant, and lower right in upper right).

In summary, these results are highly intuitive and conform with current musicological theories. The hierarchical breakdown gives a nicely interpretable set of elementary performance strategies, at various levels of granularity — a sort of alphabet of elementary patterns that can be related to various types of musical structure, and out of which complex performances can be built [34].

[4] Note that the segments were not normalized, so that absolute position of a segment within the tempo-loudness space is significant (after all, absolute tempo and loudness are also relevant performance characteristics) — hence the rather high number of classes being distinguished here.

5.3 Qualitative Evaluation

The general problem of evaluating clustering algorithms is that for quantitative comparisons the true clusters in the data need to be known. This is usually the case in typical classification problems, or in artificial datasets. The former, as the name indicates, are not typical clustering problems, the latter can be tuned in such a way that any algorithm could outperform any other algorithm.

The main field of application for the GHSOM is exploratory data analysis. Thus, to evaluate the TMR extension we applied both the original GHSOM and the GHSOM-TMR to explore the 2 datasets mentioned above. In both cases the true cluster structure was unknown and the objective was to *explore the data* and gain new insights. Furthermore, in both cases any structure found in the data can easily be evaluated by either listening to music, or viewing 2-dimensional plots.

In general, the first step to validate any results obtained from visual clustering techniques including the GHSOM-TMR is to compare the results to the results obtained from other techniques and even to the same technique with different random initializations. This is necessary to ensure that the structure perceived to be significant is not only caused by some local minimum in an error function. In a second step the new insights gained from the clustering techniques need to be tested on new data.

For our evaluation both algorithms, i.e. the GHSOM with and without TMR extension, the first layer SOM was initialized equally using Principal Component Analysis to ensure equal starting conditions. Furthermore, the neighborhood radius r_t for the batch-SOM training was set generously to ensure best possible convergence and to avoid randomly falling into local minima specific for the particular dataset.

In particular, we first trained the original GHSOM and searched for values of τ_1 that lead to good hierarchical representations of the data. We found that the results were best when the GHSOM parameter controlling the size of the individual maps is set to $\tau_1 = 0.6$ in the case of the music archive and to $\tau_1 = 0.7$ for the musical performances when using the SMQE. We then trained the GHSOM-TMR (with the same τ_2; no τ_1 needs to be determined for the GHSOM-TMR) and obtained the same quality level. Achieving significantly better results would have been a big surprise since, for example, the results obtained from the second dataset had previously been carefully analyzed and optimized manually [28]. Thus, we would like to emphasize that the TMR extension is capable of creating the same hierarchical structures which were fine-tuned manually by GHSOM experts using the original algorithm.

Determining appropriate values for τ_1 requires the user to understand the GHSOM algorithm, and even a GHSOM expert will need to calculate several maps and evaluate these manually before guessing the best value for τ_1. On the other hand, with the TMR extension the appropriate map sizes are determined automatically and the user does not need to worry about setting τ_1 which simplifies the use of the GHSOM.

The computational load of the TMR extension is about twice as high as the original GHSOM algorithm. The main reason for this lies in the 'look ahead strategy' of the TMR extension. However, the original GHSOM algorithm requires the user to make additional runs to test and understand the parameter setting. The evaluation of these additional runs usually requires more time than the training process.

6 Conclusions

We have presented the Tension and Mapping Ratio extension (TMR) to reveal the inherent hierarchical structure in data based on Growing Hierarchical Self-Organizing Maps (GHSOM). The TMR addresses several shortcomings of the GHSOM with respect to decisions made in the growth process. Using simple 2-dimensional datasets we have shown in which cases the GHSOM-TMR can outperform the original GHSOM. In particular, we were able to show that the TMR improves the decision where to insert units on a local as well as on a global level. Furthermore, the TMR outperforms other quality measures used to select the appropriate map size.

In practical applications of the GHSOM we came to the conclusion that the main advantage of the TMR extension is its ability to automatically determine the appropriate granularity within certain areas of the hierarchical structure representing the data, thus the user is no longer required to manually determine the threshold parameter τ_1.

Furthermore, we have presented two interesting applications of the TMR extension in the music domain. The two datasets used as well as a Matlab toolbox of the original GHSOM algorithm and TMR extension are available from http://www.oefai.at/˜elias/ghsom.

7 Acknowledgments

This research has been carried out in the project Y99-INF, sponsored by the Austrian Federal Ministry of Education, Science and Culture (BMBWK) in the form of a START Research Prize. The BMBWK also provides financial support to the Austrian Research Institute for Artificial Intelligence. The authors wish to thank Marcus Ludl and Andreas Rauber for valuable discussions.

References

[1] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan. Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Transactions on Neural Networks*, 11(3):601–614, 2000.

[2] J. C. Bezdek and N. R. Pal. Some new indexes of cluster validation. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 28(3):301–315, 1998.

[3] A. Chan and E. Pampalk. Growing Hierarchical Self Organising Map (GHSOM) Toolbox: Visualisations and Enhancements. In *Proceedings of the 9th*

International Conference on Neural Information Processing (ICONIP'02), volume 5, pages 2537–2541, Singapore, 2002.

[4] D. L. Davies and D. W. Bouldin. A cluster seperation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(4):224–227, 1979.

[5] G. DeBoeck and T. Kohonen, editors. *Visual Explorations in Finance*, Berlin, Germany, 1998. Springer Verlag.

[6] M. Dittenbach, D. Merkl, and A. Rauber. The Growing Hierarchical Self-Organizing Map. In S. Amari, C. L. Giles, M. Gori, and V. Puri, editors, *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, volume VI, pages 15–19, Como, Italy, 2000. IEEE Computer Society.

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

[8] M. Dittenbach, A. Rauber, D. Merkl. Uncovering Hierarchical Structure in Data Using the Growing Hierarchical Self-Organizing Map. *Neurocomputing*, 48(1-4):199–216, 2002.

[9] B. Fritzke. Growing Grid – A self-organizing network with constant neighborhood range and adaption strength. *Neural Processing Letters*, 2(5):9–13, 1995.

[10] W. Goebl. Melody lead in piano performance: Expressive device or artifact? *Journal of the Acoustical Society of America*, 110(1):563–572, 2001.

[11] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441 and 498–520, 1933.

[12] S. Kaski. Fast winner search for SOM-based monitoring and retrieval of high-dimensional data. In *Proceedings of the International Conference of Artificial Neural Networks*, pages 940–945, 1999.

[13] T. Kohonen. Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[14] T. Kohonen. New developments of learning vector quantization and the self-organizing map. In *SYNAPSE'92, Symposium on Neural Networks*, Osaka, Japan, 1992. Alliances and Perspectives in Senri.

[15] T. Kohonen. Self-organzation of very large document collections: State of the art. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 65–74, Skövde, Sweden, 1998.

[16] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, 3rd edition, 2001.

[17] 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*, 11(3):547–585, 2000.

[18] P. Koikkalainen. Fast deterministic self-organizing maps. In *Proceedings of the International Conference on Neural Networks*, Paris, France, 1995.

[19] P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *Proceedings of the International Joint Conference on Neural Networks*, San Diego, CA, 1990.

[20] J. Langner and W. Goebl. Representing expressive performance in tempo-loudness space. In *Proceedings of the ESCOM 10th Anniversary Conference on Musical Creativity*, Liège, 2002.

[21] A. Lensu and P. Koikkalainen. Complexity Selection of the Self-Organizing Map In *Proceedings of the International Conference on Neural Networks (ICANN'02)*, pages 927–932, Madrid, Spain, 2002.

[22] D. Merkl and A. Rauber. Document classification with unsupervised neural networks. In F. Crestani and G. Pasi, editors, *Soft Computing in Information Retrieval*, pages 102–121. Physica Verlag, 2000.

[23] D. Merkl and A. Rauber. Uncovering the Hierarchical Structure of Text Archives by Using an Unsupervised Neural Network with Adaptive Architecture. In *Proceedings of the 4th Pacific Asia Conference on Knowledge Discovery and Data Mining*, pages 384–395, Kyoto, Japan, 2000, Springer.

[24] R. Miikkulainen. Script recognition with hierarchical feature maps. *Connection Science*, 2, 1990.

[25] E. Pampalk. Islands of Music: Analysis, Organization, and Visualization of Music Archives. Master's thesis, Vienna University of Technology, 2001. http://www.oefai.at/˜elias/music/thesis.html.

[26] E. Pampalk, A. Rauber, and D. Merkl. Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In *Proceedings of the International Conference on Artifical Neural Networks (ICANN'02)*, pages 871–876, Madrid, Spain, 2002.

[27] E. Pampalk, A. Rauber, and D. Merkl. Content-based Organization and Visualization of Music Archives. In *Proceedings of the ACM Multimedia 2002*, pages 570–579, Juan les Pins, France, 2002.

[28] A. Rauber, E. Pampalk, and D. Merkl. Using Psycho-Acoustic Models and Self-Organizing Maps to Create a Hierarchical Structuring of Music by Sound Similarities. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02)*, pages 71–80, Paris, France, 2002.

[29] D. Roussinov and H. Chen. Information navigation on the web by clustering and summarizing query results. *Information Processing and Management*, 37:789–816, 2001.

[30] A. Ultsch and H. P. Siemon. Kohonen's Self-Organizing Feature Maps for Exploratory Data Analysis. In *Proceedings of the International Neural Network Conference (INNC'90)*, pages 305–308, Dordrecht, Netherlands, 1990. Kluwer.

[31] J. Vesanto and E. Alhoniemi. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000.

[32] C. Wallace and D. Boulton. An Information Measure for Classification. *Computer Journal*, 11:185–194, 1968.

[33] G. Widmer. Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report. *AI Communications*, 14(3):149–162, 2000.

[34] G. Widmer. In Search of the Horowitz Factor: Interim Report on a Musical Discovery Project. In *Proceedings of the 5th International Conference on Discovery Science (DS'02)*, Lübeck, Germany. Berlin, Springer Verlag.

[35] E. Zwicker and H. Fastl. *Psychoacoustics, Facts and Models*, volume 22 of *Springer Series of Information Sciences*. Springer, Berlin, 2nd updated edition, 1999.