

Visualizing Changes in the Structure of Data for Exploratory Feature Selection

Elias Pampalk¹, Werner Goebel¹, and Gerhard Widmer^{1,2}

¹Austrian Research Institute for Artificial Intelligence (OeFAI)
Freyung 6/6, A-1010 Vienna, Austria

²Department of Medical Cybernetics and Artificial Intelligence
University of Vienna

{elias, wernerg, gerhard}@oefai.at

ABSTRACT

Using visualization techniques to explore and understand high-dimensional data is an efficient way to combine human intelligence with the immense brute force computation power available nowadays. Several visualization techniques have been developed to study the cluster structure of data, i.e., the existence of distinctive groups in the data and how these clusters are related to each other. However, only few of these techniques lend themselves to studying how this structure changes if the features describing the data are changed. Understanding this relationship between the features and the cluster structure means understanding the features themselves and is thus a useful tool in the feature extraction phase.

In this paper we present a novel approach to visualizing how modification of the features with respect to weighting or normalization changes the cluster structure. We demonstrate the application of our approach in two music related data mining projects.

Categories and Subject Descriptors

I.5.3 [Pattern Recognition]: Clustering—*similarity measures, algorithms*

Keywords

High-Dimensional Data, Interactive Data Mining

1. INTRODUCTION

A common problem in data mining is to extract and select the right features for further analysis. This is particularly true for complex high-dimensional data such as images or music. In many applications the different options of the preprocessing and feature extraction procedures can be described in terms of parameters which are adjusted to fit specific needs. For example, such a parameter can define the

value of the exponent of a power-law used to compress the range of specific values, the weighting between features, or if the data is variance normalized or not.

In exploratory data analysis there is generally no specific target function given which could be used to optimize these parameters. Furthermore additional parameters might be introduced by the distance function (e.g. Minkowski metric) which is necessary for cluster analysis. Finding appropriate values for all of these parameters and in consequence defining the feature extraction procedure remains a task which requires domain expertise and human intelligence.

In this paper we present a new technique to visualize the influence of such parameters on the cluster structure of the data. The intention is to offer domain experts the possibility to interactively explore the influence of the parameters, gain new insights, and inspire new hypothesis for further analysis.

The technique we propose is particularly applicable to very high-dimensional data represented by low-level features. It is based on a new extension to the Self-Organizing Map (SOM) [19] algorithm and on Smoothed Data Histograms (SDH) [28]. In particular, we present Aligned-SOMs, i.e., multiple SOMs stacked on top of each other and aligned so that they are organized in a similar way. The SOMs are trained to represent the same data items in different data spaces defined by slightly different values for the parameters mentioned above. We use Aligned-SOMs in combination with SDH to visualize how gradual changes in the feature extraction procedure slowly change the organization of the data in the 2-dimensional visualization space.

To demonstrate our approach we present a system designed in cooperation between a data miner and a musicologist to analyze expressive performances of classical piano music by internationally renowned pianists. Furthermore, we apply the same concept to create an interface for exploring the contents of digital libraries. In particular, we demonstrate how a music collection is organized and visualized based on a combination of timbre (i.e., sound characteristics which distinguish different instruments) and rhythmic characteristics and how this organization gradually changes when the focus of interest is changed in favor of one of these.

The remainder of this paper is organized as follows. In *Section 2* we review related work. In *Section 3* we review the SOM and the SDH visualization. In *Section 4* we present Aligned-SOMs. In *Section 5* we present two applications of Aligned-SOMs to explore expressive performances of classi-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGKDD '03, August 24-27, 2003, Washington, DC, USA
Copyright 2003 ACM 1-58113-737-0/03/0008 ...\$5.00.

cal music and to explore archives of popular music. Finally, in *Section 6* we draw some conclusions.

2. RELATED WORK

In general, visualization techniques are powerful tools that are frequently employed in knowledge discovery processes. Visualizations can make complex relationships easily understandable and stimulate visual thinking [9]. Especially, tools which visualize the cluster structure of data are valuable for exploring and understanding data. Such tools include data histograms for one-dimensional data as well as algorithms which project high-dimensional data to a two-dimensional visualization space trying to preserve the topology, i.e., trying to ensure that distances between data items in the visualization space correspond to the distances in the high-dimensional data space.

A popular choice for non-linear projections is the Self-Organizing Map (SOM) [19]. Alternatives include non-linear Multi-Dimensional Scaling (MDS) [21] or linear Principal Component Analysis [11]. For example, in [34] an approach was presented where a hyperbolic metric is combined with MDS. The user can interactively change the focus to different regions in the data space, thus, viewing the relationship of items in the chosen region with a relatively high resolution while maintaining the overall context. However, this approach is based on given distances between data items, while our aim is to aid the user in defining how these distances shall be calculated. Once an appropriate definition for the distances is found, focusing on details in different regions of the data space would be one of the next steps.

The distances in the data space depend on the metric, the extracted features, and how the individual features are normalized and weighted. In the experiments presented in this paper we use the Euclidean distance metric. However, in the same way the features can be weighted or normalized differently it is also possible to change the distance metric. The impact of distance metrics on high-dimensional data has been studied, for example, in [4]. The problem of defining a similarity between data items can be simplified if for some of the data the similarity is known [16].

Changing the distances between data items changes the structure of the data. For example, if the data consists of several piano pieces which vary in tempo, the structure would be one big cluster when focusing on sound characteristics only. One the other hand, the structure might consist of several small clusters if the focus is on rhythm.

To understand the relationship between different ways of weighting features, it is useful to visualize the gradual changes in the structure when shifting focus from one feature to another. Previous work in this direction includes Star Coordinates [12], which are based on scatter plots where the data is projected onto a non-orthogonal coordinate system representing the multi-dimensional data space. The resulting ambiguities are resolved when the user interacts with the visualization. The user can emphasize a particular feature by giving a single dimension more space, i.e., increasing the length of the respective axes, and rearranging all other axis so that they are orthogonal to the emphasized one. The main difference to our approach is, that we do not assume each data dimension by itself to be meaningful but rather assume many low-level attributes which as a whole resemble an abstract concept.

A different approach to combining human intuition with

the processing power of computers to find a suitable projection of the data is [1, 3]. The approach is based on Polarized Projections, i.e., the data is projected into a subspace defined by polarization anchors which have a similar function as the model vectors in the SOM. Given a polarized projection the data is visualized using kernel density estimators allowing the user to easily identify clusters. Although this allows an interactive search for the best way to project the high-dimensional data, there are several differences to our work. Foremost, the approach focuses on finding clusters in the data, while we try to understand and find the right parameters for the feature extraction process. The projections which are defined through the polarization anchors cannot be interpreted directly in such a way that would allow direct feedback to the feature extraction process. Furthermore, in contrast to the Polarized Projections, our emphasis is on linking different views of the same data so that the data density visualization changes smoothly between slightly different projections of the same data.

Recently, a framework for visualizing changes in the density distribution of data was presented in [2]. In contrast to the approach we present, the framework was applied to understand changes in evolving data streams using differential kernel density estimation with various window sizes. In evolving data streams the same data spaces are used at different points in time while the data items change. Other approaches analyzing the changes in data characteristics include [10] where the focus is on measuring the effects on data mining models instead of intuitively visualizing changes.

3. SELF-ORGANIZING MAPS

The Self-Organizing Map (SOM) [17, 19], an unsupervised neural network, has successfully been applied in exploratory data analysis [13] with applications in various domains such as finances [8]. The SOM is a powerful tool for visual clustering [33] and analyzing correlations in the data [32]. Furthermore, a variant of the SOM, the Adaptive Subspace SOM [18] has been developed to automatically detect invariant features in dynamic signals.

One of the best known applications of the SOM is the *WebSOM* project [15, 20] where millions of high-dimensional text documents are organized according to their similarity to create an intuitive user interface for interactive exploration. Alternatives to the SOM include Multi-Dimensional Scaling [21], Sammon's Mapping [30], and Generative Topographic Mapping (GTM) [6]. The approach we present can be reformulated to use either of these, however, we have chosen the SOM because of its computational efficiency.

The idea of the SOM is to map the high-dimensional data to a 2-dimensional map in such a way that similar items are located close to each other. The resulting mapping reflects the cluster structure of the data, i.e., the clusters and their relationship to each other.

The SOM consists of an ordered set of units which are arranged in a 2-dimensional visualization space, referred to as map. Common choices to arrange the map units are rectangular or hexagonal grids. Each unit is defined through its distance to the other units, and is assigned a model vector in the high-dimensional data space. To map a data item from the data space to the map it is necessary to calculate the distance between the data item and all model vectors to find the model vector with the smallest distance, i.e., the most similar model vector. The data item is then mapped to

the respective unit, also referred to as best matching unit. Thus, every point in the data space can be assigned to a location on the 2-dimensional map.

3.1 The Batch-SOM Algorithm

The SOM can be initialized randomly, i.e., random vectors in the data space are assigned to each model vector. Several alternatives using, for example, a Principal Component Analysis to initialize the SOM can be found in [19].

The SOM training is basically a loop which is repeated until convergence. In each iteration the best matching unit for each data item is calculated. Then the model vectors of the respective units are updated so that they fit the data better. This would be identical to k-means clustering [25] were it not for a constraint which forces the model vectors of neighboring units to represent similar data items. The units which are considered to be in the neighborhood of a particular unit are defined through a neighborhood function based on the distances between the units on the map. An important aspect of the SOM is that the size of the neighborhood decreases slowly with each iteration to finally end up with very small neighborhoods allowing each unit to perfectly adapt to the data it represents.

To formalize the batch-SOM algorithm we define the data matrix \mathbf{D} , the model vector matrix \mathbf{M}_t , the distance matrix \mathbf{U} , the neighborhood matrix \mathbf{N}_t , the partition matrix \mathbf{P}_t , and the spread activation matrix \mathbf{S}_t . The data matrix \mathbf{D} is of size $n \times d$ where n is the number of data items and d is the number of dimensions. Each row represents one data item. The model vector matrix \mathbf{M}_t is of size $m \times d$, where m is the number of map units. The values of \mathbf{M}_t change with each iteration t . The distance matrix \mathbf{U} of size $m \times m$ defines the squared distances between the units on the map. Thus, \mathbf{U} is symmetrical with zeros on the diagonal. The neighborhood matrix \mathbf{N}_t can be calculated, for example, as

$$\mathbf{N}_t = e^{-\mathbf{U}/r_t^2}, \quad (1)$$

where r_t defines the neighborhood radius and monotonically decreases with each iteration. \mathbf{N} is of size $m \times m$, symmetrical, with high values on the diagonal, and represents the influence of one unit on another. The sparse partition matrix \mathbf{P}_t is calculated given \mathbf{D} and \mathbf{M}_t . In particular, at iteration t ,

$$\mathbf{P}_t(i, j) = \begin{cases} 1, & \text{if unit } j \text{ is the best match for item } i, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Thus, \mathbf{P}_t is of size $n \times m$ and the sum over all columns of each row equals 1. The spread activation matrix \mathbf{S}_t , with size $n \times m$, defines the responsibility of each unit for each data item at iteration t . Thus, $\mathbf{S}_t(i, j)$ will be high if j is the best matching unit for item i , and depending on how large the neighborhood is, all units in the neighborhood of unit i will have relatively high values too for i . The spread activation is calculated as,

$$\mathbf{S}_t = \mathbf{P}_t \mathbf{N}_t. \quad (3)$$

At the end of each loop the new model vectors \mathbf{M}_{t+1} are calculated as,

$$\mathbf{M}_{t+1} = \mathbf{S}_t^* \mathbf{D}, \quad (4)$$

where \mathbf{S}_t^* denotes the spread activation matrix which has been normalized so that the sum over all rows in each column equals 1 except for units without responsibilities. Note,

that if a unit is not responsible for any data item, i.e., the sum over all rows in the respective column equals 0, the normalization would cause divisions by zero. Thus, the columns representing units without responsibilities are not normalized and the respective model vectors are not updated.

There are two main parameters which need to be adjusted by the user. The first is the number of map units, i.e., how large the SOM should be. This decision is basically a computational one. More map units require more training iterations and for each iteration the computation of the best matching units becomes more intense. On the other hand, more units lead to a higher resolution of the mapping.

The most difficult parameter to adjust is the final neighborhood radius relative to the number of map units. To adjust this correctly it is necessary to know the level of noise in the data. Very noisy data requires a large final radius. The final radius defines the smoothness of the mapping.

3.2 Smoothed Data Histograms

Various methods to visualize clusters based on the SOM have been developed. The most prominent method visualizes the distances between the model vectors of adjacent units and is known as the U-matrix [31]. We use Smoothed Data Histograms (SDH) [28] where each data item *votes* for the map units which represent it best based on some function of the distance to the respective model vectors. All votes are accumulated for each map unit and the resulting distribution is visualized on the map.

As voting function we use a robust ranking where the map unit closest to a data item gets n points, the second $n-1$, the third $n-2$ and so forth, for the n closest map units. All other map units are assigned 0 points. The parameter n can interactively be adjusted by the user. The concept of this visualization technique is basically a density estimation, thus the results resemble the probability density of the whole dataset on the 2-dimensional map (i.e. the latent space). The main advantage of this technique is that it is computationally not more expensive than one iteration of the batch-SOM algorithm. On the other hand, it does not offer a clear statistical interpretation as, for example, the probability density defined by the GTM algorithm.

3.3 Illustration of SOM and SDH

Figure 1 illustrates characteristics of the SOM and the cluster visualization using a synthetic 2-dimensional dataset. Although in general it does not make sense to use the SOM to analyze 2-dimensional data this dataset allows us to illustrate some aspects of the SOM algorithm which would be difficult to visualize otherwise.

One important aspect is the topology preservation. Map units next to each other on the grid represent similar regions in the data space. This can be seen by the arrangement of the model vectors, which are connected by lines indicating which model vectors are assigned to neighboring map units (cf. Figure 1c). If k-means had been used instead of the SOM, the connecting lines would be drawn randomly between points, while the SOM has learned to represent the data in such a way that the model vectors are arranged according to the organization of the map units.

Another important aspect, which is illustrated in Figure 1c, is that the SOM defines a non-linear mapping from the data space to the 2-dimensional map. The distances between neighboring model vectors are not uniform. Areas in

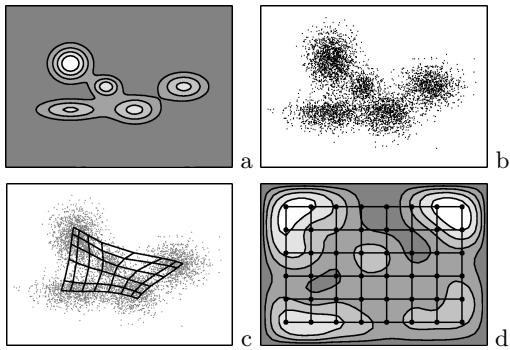


Figure 1: Illustration of SOM and SDH, (a) probability distribution in the 2-dimensional data space, (b) sample drawn from this distribution, (c) model vectors of the SOM in the data space, (d) map units of the SOM in the visualization space with clusters visualized using SDH ($n=3$ with spline interpolation). High density areas are visualized with white, low density with gray. The model vectors and the map units of the SOM are represented by the nodes of the rectangular grid.

the data space with a high density are represented by more model vectors, thus, in higher detail than sparse areas. This characteristic is exploited by the U-matrix visualization.

However, not all model vectors are located in dense areas and some model vectors remain in sparse areas to maintain the overall structure of neighboring units. These units which might not represent any data are known as interpolating units. The fact that not every unit represents the same amount of data items is exploited by the SDH (cf. Figure 1d).

4. ALIGNED-SOMS

The goal is to understand the relationship between different ways of representing the same data by visualizing changes in the cluster structure. Thus, we assume that the dataset can be represented in different but related ways depending on various parameters in the feature extraction process.

Aligned-SOMS are a new approach to visualizing the influence of such parameters by training multiple SOMs, i.e., we stack several SOMs on top of each other and obtain several SOM layers representing the same data from different points of view. Each layer has a slightly different point of view than its neighbors. The main constraint we apply to the layers is that they map the same data to similar locations as their neighbors. Then the user can move through the layers and see how the distribution of the data gradually changes as the parameters defining the feature extraction process are changed.

The Aligned-SOMS are trained in such a way that each layer maps similar data items close to each other within the layer, and that neighboring layers map the same items to similar locations. To ensure that two neighboring SOM layers have a similar organization we define a distance between two layers, which we choose to be smaller than the distance between two adjacent units on each map. For example, while the distance between two adjacent units within

a layer is set to 1, the distance between two layers can be set to $1/5$. Thus, two layers separated spatially by 4 other layers would be constrained as strongly as two adjacent units within a layer to represent the same data items. Note that it is not sufficient to use the normal SOM training algorithm to map the data to 3-dimensional grids [19], because each SOM layer represents not only the same data, but also has a different data space. Thus, a model vector from one layer cannot directly be interpreted in a different layer.

4.1 Training Algorithm for Aligned-SOMS

We formulate the Aligned-SOMS training algorithm based on the formulation of the batch-SOM in Section 2. To train the SOM layers we extend the distance matrix \mathbf{U} to contain the distances between all units in all layers, thus the size of \mathbf{U} is $ml \times ml$, where m is the number of units per layer and l is the total number of layers. Each layer i has its own model vectors \mathbf{M}_{it} of size $m \times d$ and data \mathbf{D}_i of size $n \times d$. The neighborhood matrix is calculated according to Equation 1. The sparse partition matrix \mathbf{P}_t is of size $n \times ml$ and calculated using Equation 2 with the extension that the best matching unit for a data item is calculated for each layer. Thus, the sum over all columns in each row equals the number of layers. The spread activation matrix \mathbf{S}_t is calculated as in Equation 3. The updated model vectors \mathbf{M}_{it+1} are calculated as,

$$\mathbf{M}_{it+1} = \mathbf{S}_{it}^* \mathbf{D}_i, \quad (5)$$

where \mathbf{S}_{it}^* denotes the normalized columns of \mathbf{S}_t which represent the model vectors of layer i .

To initialize the Aligned-SOMS in our experiments we have first trained the layer representing the most complex data space, e.g., the layer in which sound characteristics (timbre) and rhythm are equally weighted, and then initialized the spread activation of all layers based on \mathbf{S}_{it}^* of the most complex layer.

The necessary resources in terms of CPU time and memory are proportional to the number of layers and depend on the complexity of the feature extraction parameters analyzed. Thus, the overall computational load is of a higher magnitude than training a single SOM. In practice, we have experienced that standard hardware (P4 2GHz with 512MB RAM) is sufficient to run Aligned-SOMS on our datasets. For example, the experiments we discuss in Subsection 5.1, which are calculated from over 10,000 multivariate time series segments, run within 1 hour including the time it takes Matlab to create over 500 image files for the HTML interface. For larger datasets several optimizations of the algorithm are possible, in particular, applying an extended version of the fast winner search [14] would improve the efficiency drastically, since there is a high redundancy in the multiple layer structure.

4.2 Illustration of Aligned-SOMS

To illustrate the Aligned-SOMS we use a simple dataset containing 16 animals with 13 boolean features describing their appearance and activities [19]. The dataset is depicted in Table 1. We assume, that it is not clear how to best represent the animals and that the weighting ratio between appearance and activity features is of interest. Thus, we have trained 31 layers of SOMs using the Aligned-SOM training algorithm. The first layer uses a weighting ratio between appearance and activity features of 1:0. The 16th layer, i.e.,

	Appearance								Activities				
	Small	Medium	Large	2-Legs	4-Legs	Hair	Hooves	Mane	Feathers	Hunt	Run	Fly	Swim
Dove	x			x					x			x	
Chicken	x			x									
Duck	x			x					x				x
Goose	x			x					x			x	x
Owl	x			x					x	x		x	
Hawk	x			x					x	x		x	
Eagle		x		x					x	x		x	
Fox		x		x	x				x				
Dog		x		x	x						x		
Wolf		x		x	x			x	x	x			
Cat	x			x	x				x				
Tiger			x	x	x				x	x			
Lion		x		x	x			x	x	x			
Horse			x	x	x	x	x	x			x		
Zebra			x	x	x	x	x				x		
Cow			x	x	x	x							

Table 1: 16 animals described by 13 attributes.

the center layer, weights both feature groups equally. The last layer uses a weighting ratio of 0:1, thus, focuses only on activities. The weighting ratios of all other layers are linearly interpolated.

From the resulting Aligned-SOMs 5 layers are depicted in Figure 2. For interactive exploration a HTML version with all 31 layers is available on the internet.¹ When the focus is only on appearance all small birds are located together in the lower right corner of the map. The Eagle is an outlier because of its size. On the other side, all mammals are located in the upper half of the map separating the medium sized ones on the left from the large ones on the right. As the focus is gradually shifted to activity features the structure changes. In particular, the animals are arranged in such a way that predators are located on the left and others on the right. Although there are several significant changes regarding individuals, the overall structure has remained largely the same, enabling the user to easily identify similarities and differences between two different ways of viewing the same data.

5. APPLICATIONS

In this section we present two applications of Aligned-SOMs. The first application is the identification of distinctive sequences in multivariate time series data representing musical performance strategies. The second application is the content-based organization and visualization of a music collection for interactive exploration. For both applications we use a HTML based user interface with JavaScript and many images to conveniently interact with the Aligned-SOMs. A demonstration is available on the internet.¹

5.1 Exploring Musical Performance Strategies

The first application is part of a large data mining project² whose goal is to study fundamental principles of expressive

¹<http://www.oefai.at/~elias/kdd03/>

²<http://www.oefai.at/music>

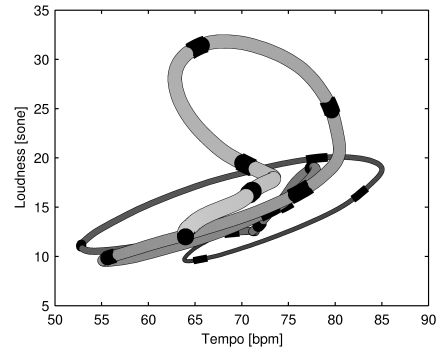


Figure 3: Part of a trajectory corresponding to an expert performance of Chopin etude op. 10, No. 3. The loudness and tempo curves are smoothed. The bar boundaries are indicated through the black sections. The time dimension is visualized through the thickness and shading of the trajectory.

music performance [35, 36]. Performances by concert pianists are measured with respect to timing and 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 Mozart piano sonatas, played by 6 internationally renowned pianists (Daniel Barenboim, Roland Batik, Glenn Gould, Maria João Pires, András Schiff, Mitsuko Uchida). Each performance is characterized by two series of numeric values that represent the measured tempo and loudness, respectively, over the course of the performance. An example of one such time series in the form of a smoothed trajectory in the two-dimensional tempo-loudness space is shown in Figure 3, with tempo on the vertical axis and loudness on the horizontal axis. Details of this form of performance visualization can be found in [23]. The various trajectories are cut into overlapping segments each represented by 60 low-level features. The purpose of the whole procedure is to find out whether there are indeed characteristic and interpretable classes of tempo-loudness strategies that pianists apply consistently, and whether these are different between performers.

At the current state of our research it is not clear how to best represent the performance trajectories to capture the main characteristics. Some of the open questions are related to the weighting of the tempo and loudness dimensions, strength of the trajectory smoothing, and the normalization of the data.

Regarding the normalization we have found 5 forms to be of particular interest which can be categorized in 3 levels. The first level is no normalization, the second level is normalizing the mean, the third level is to normalize mean and variance. The effect of the second level is that we focus only on *absolute changes* regarding loudness or tempo. For example, did the pianist speed up by 10 beats per minute (bpm)? In the third level we focus only on *relative changes*, for example, has the pianist played 10% faster or slower? Within the second and third levels we distinguish between 2 ways of normalizing the data, namely, normalizing over a short segment of the trajectory ('local') or normalizing over

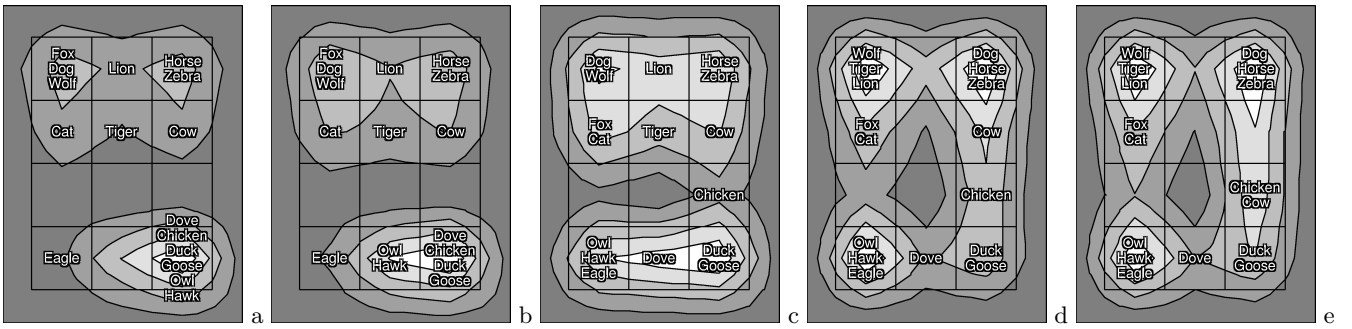


Figure 2: Aligned-SOMs trained with the animal dataset, (a) first layer with weighting ratio 1:0 between appearance and activity features, (b) first quarter layer with ratio 3:1, (c) center layer with ratio 1:1, (d) last quarter layer with ratio 1:3, (e) last layer with ratio 0:1. The shadings represent the density calculated using SDH ($n=2$ with linear interpolation).

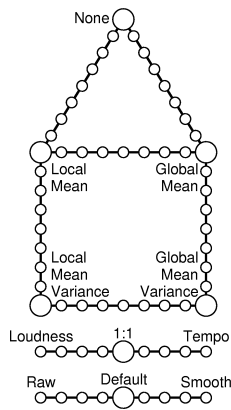


Figure 4: The feature extraction options and their relationships.

a longer sonata part ('global').

Figure 4 shows the connection between the 5 forms of normalizing the data. In addition, two dimensions are included which control the weighting between loudness and tempo, and the degree of smoothing the trajectories. For each of these different ways to extract features we analyze the effects on the cluster structure of the data. In particular, we analyze changes in the density distribution of trajectories in different sonatas, and in different pianists.

5.1.1 User Interface

A screenshot of the HTML user interface is shown in Figure 5. The data used consists of several fast Mozart piano sonatas with a segment length of 10 beats. The user interface is divided into 3 parts. The first part contains the navigation unit, below it an eigenvalue indicator, and a visualization of the SOM model vectors, namely the codebook. The second part contains the SDH visualization for each pianist. To make differences between the 6 SDH visualizations more apparent, in addition, we visualize the individual SDHs after subtracting the average SDH. From this contrasting visualization it is easily possible to identify which patterns are used particularly often or seldom by a pianist compared to the average usage of the patterns. The third part contains the SDH visualizations of each sonata part. Although we

are not primarily interested in finding patterns which distinguish sonatas from each other, this visualization has proven to be useful in evaluating the normalizations.

The navigation unit is the same as in Figure 4. By moving the mouse over the circles, the corresponding images are shown in the HTML viewer using JavaScript. Three markers are used to indicate the current position to the user. One marker indicates the normalization, one the weighting with respect to loudness and tempo, and the third marker indicates the degree of smoothing. Between two extremes we use 7 intermediate positions. Therefore, given the overall structure, over 3,800 SOMs would need to be trained to allow every possible combination of the three markers. Due to computational considerations we have limited the combinations such that a marker can only be moved to the small circles if the other two markers are located on big circles, thus, reducing the number of SOMs to 127.

The eigenvalue indicator displays the first, the second, and the sum of all other eigenvalues. The eigenvalues serve as indication of the complexity of the cluster structure in the data space. For example, for the sonatas used in Figure 5 if no normalization is applied most of the variance can be explained using a linear projection into a 2-dimensional space. The reason is that the relative variations within a segment are negligible compared to the large variations of the absolute loudness and tempo given several sonatas played by different pianists. On the other hand, the most complex data space is the one illustrated in Figure 5, i.e., local mean and variance normalization with unsmoothed trajectories and an equal weighting between loudness and tempo.

The codebook shows the model vector of each unit. The trajectory segments are visualized by blue lines with a red dot marking the end. The blue shading represent the variance of the data items mapped to the unit. The number in the lower right of each unit displays the number of data items mapped to the respective unit. For example, in Figure 5 there are 578 items mapped to the second unit in the first row. The trajectory starts fast and loud and almost linearly moves to slow and soft (ritardando-decrescendo). The codebook gives valuable insights into frequent patterns, however, it depends on the data analyzed if it is possible to visualize the codebook. When analyzing, for example, a text document collection it might be interesting to use a list of frequent words or other summarization strategies instead [22, 29].

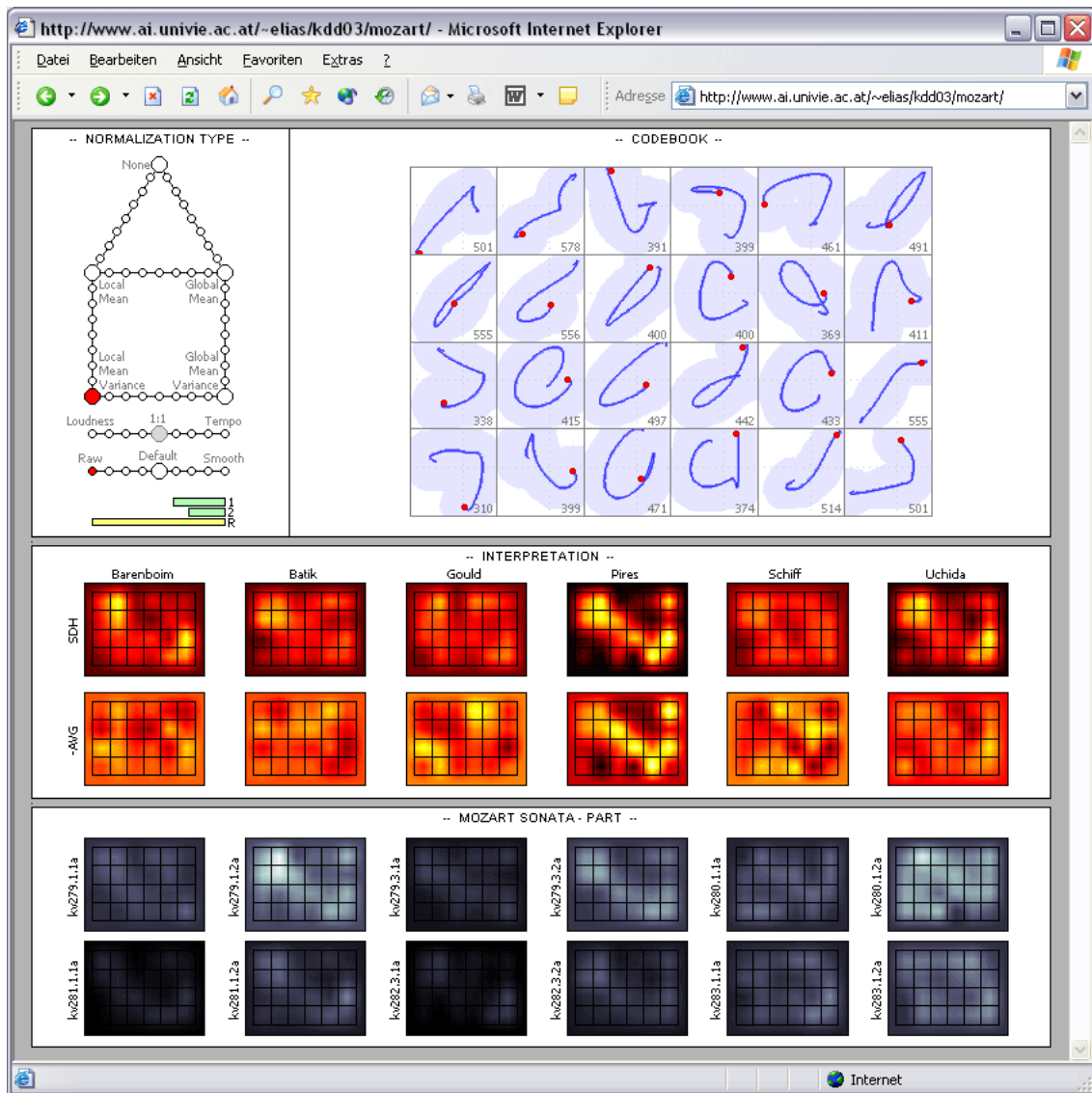


Figure 5: Screenshot of the user interface used for exploring the effects of the feature extraction on the expressive performances of Mozart piano sonatas.

5.1.2 First Results

Studying the influence of the low-level features on the distinguishability of performance strategies is ongoing research. Using the interactive interface presented above we have been able to understand the data better. Moreover, visualizing the effects of normalization and weighting have helped communicate ideas between data miners and musicologists in our research project. We invite the reader to verify the following observations by interacting with the visualization provided at <http://www.oefai.at/~elias/kdd03/mozart/>.

One example of a rather trivial result is the influence of the loudness-tempo weighting when no normalization is applied. When the focus is only on loudness major differences between the pianists are clearly recognizable in the SDH visualization. In particular, the recordings by Batik are much louder than the others while Schiff is by far the softest. On the other hand, the SDH visualizations of the sonatas are

very similar and hardly distinguishable. Both observations are intuitive since the CD recordings vary in terms of average loudness. Thus, the distinctions made with this normalization are not due to specifics of pianists, but rather to specifics of the recordings.

On the other hand, if the focus is on tempo, then the differences between the pianists diminish, except for Gould. Comparing the SDH with the codebook reveals that the performances by Gould are outliers since they are either extremely fast or slow. The SDH of the sonatas reveals that they are very distinguishable in terms of tempo which is obvious since some are simply faster than others.

This distinction between sonatas is lost when normalizing the mean either locally for a segment or globally for a whole sonata part. When normalizing the mean locally the SDH of the pianists reveals that Gould plays very frequently patterns of relative constant loudness and tempo. When removing the average from the SDH visualization, we

can see that Pires is quite the opposite to Gould and frequently performs very strong modulations of loudness and tempo. Schiff seems to modulate more the tempo than the loudness. This is underlined when viewing the effect of the loudness-tempo weighting. When focusing only on loudness the performances of Schiff is very similar to the performances by Gould while focusing on tempo reveals that there are strong similarities between Pires and Schiff. Note that although the visualization gives valuable insights into the data which would be difficult to obtain otherwise, it is necessary to quantify any observations and test them on new data.

5.2 Exploring Music Archives

The second application is part of the project *Islands of Music*³ whose goal is to create intuitive interfaces to digital libraries of music by automatically analyzing, organizing, and visualizing pieces of music based on their perceived acoustic similarities [26, 27].

The *Islands of Music* are calculated using the SOM with a SDH visualization and a specific color scale. Similar pieces are located close to each other on the map. The resulting clusters are visualized as islands. Subclusters within clusters are visualized as mountains and hills. A parameter defines the coastline which separates the water from the islands and helps to find distinctive clusters faster. A similar effect we obtain through the coastline has been used with Polarized Projections for visual clustering [3].

The main problem when organizing pieces of music is to automatically calculate the similarities between them. Music similarity can be viewed from several different perspectives. For example, the similarity can be based on the instruments used, the melody, or the rhythm. Although it is an easy task for a human listener to judge the general similarity between two pieces, there are currently no satisfactory computational models available.

Several approaches to calculate music similarities are based on Mel Frequency Cepstrum Coefficients (MFCCs), e.g., [24, 5, 7]. The MFCCs describe sound characteristics in terms of frequency band and energy at a specific point in time and are used to model the timbre.

In our previous work we presented rhythm patterns [26, 27] to calculate similarities. The rhythm patterns capture dynamic characteristics in the loudness modulation of specific frequency bands based on psychoacoustic models [37].

Instead of defining a specific way to calculate the similarity between two pieces of music, we are developing interfaces which allow the users to define what they individually consider to be relevant aspects of similarity. A screenshot of a prototype is shown in Figure 6. The user can interactively change the weights on features describing rhythm properties and sound characteristic (timbre) using a sliding bar.

The visualization shown in Figure 6 is a prototype we use to analyze the similarity measures. Beneath the islands and the sliding bar the model vectors of each layer are visualized. The model vectors contain two types of information which are displayed separately. On the left (red color) are the rhythm patterns and on the right (blue color) the MFCC patterns.

The current position of the sliding bar is on the right side, thus, the focus is on timbre characteristics. From this point of view there are five islands in the data, each representing

a specific type of music. For example, on the island in the lower left peaceful classical pieces are located such as *Für Elise* or the *Mondscheinsonata* by *Beethoven*. On the other side, in the upper right of the map is an island where we find pieces by the aggressive rock group *Papa Roach*.

Although the details of the model vectors are irrelevant for the targeted user it allows us in the current stage of development to analyze and understand why a particular piece of music is located in a specific region. For example, when looking at the model vectors of the map in Figure 6 there are several insights into the organization of the map we can gain. For example, we can see that the rhythm patterns are organized so that patterns with overall low energy can be found on the left. While the patterns with the highest energy can be found around the island in the lower right corner, which represents music with strong beats such as the songs by *Bomfunk MC's*.

Furthermore, we use the whole system to analyze and understand the relationships between different similarity measures and, thus, to evaluate them in an intuitive manner. For example, we can simplify the calculation of the rhythm patterns and compare the simplified version to the original version. If there are no significant changes in the organization of the collection, then the simplified version is likely to be just as good. However, it is more likely to have some sort of changes in the cluster structure, which can then be easily identified. An alternative approach would be to use objective and qualitative evaluations. However, due to a lacking ground truth such evaluations of music similarity measures are currently an unsolved problem in the music information retrieval community.

One of the observations we have made with this visualization is that focusing on timbre structures the data more in terms of types of instruments or artist. For example, in the lower left of the map there are some classical pieces of music. When focusing on timbre, there is a distinction between slow piano and slow string pieces. On the other hand, when focusing on rhythm this distinction is not made. Another example is the music in the upper right of the map. When focusing on timbre, for example, pieces by *Papa Roach* are clearly separated from others, while these are mixed together with other pieces from the same style when weighting the rhythm patterns more strongly.

6. CONCLUSIONS AND DISCUSSION

We have presented a novel approach to visualizing changes in the cluster structure of data when the features describing the data are changed. Using Aligned Self-Organizing Maps the user is able to gradually and smoothly change focus between feature extraction procedures to explore how they are related and what the differences are. We demonstrated the application in two data mining projects.

In the first application where the goal is to analyze musical performance strategies the main result was that the visualization helped communicate ideas between data miners and domain experts. In particular, it helped the data miners explain why it is necessary to consider weighting of different dimensions and different forms of normalization. On the other hand, the domain experts were able to help the data miners find interesting aspects of the data to analyze in more detail.

In the *Islands of Music* project where communicating ideas is not a critical issue there were two main results. The first

³<http://www.oefai.at/~elias/music>

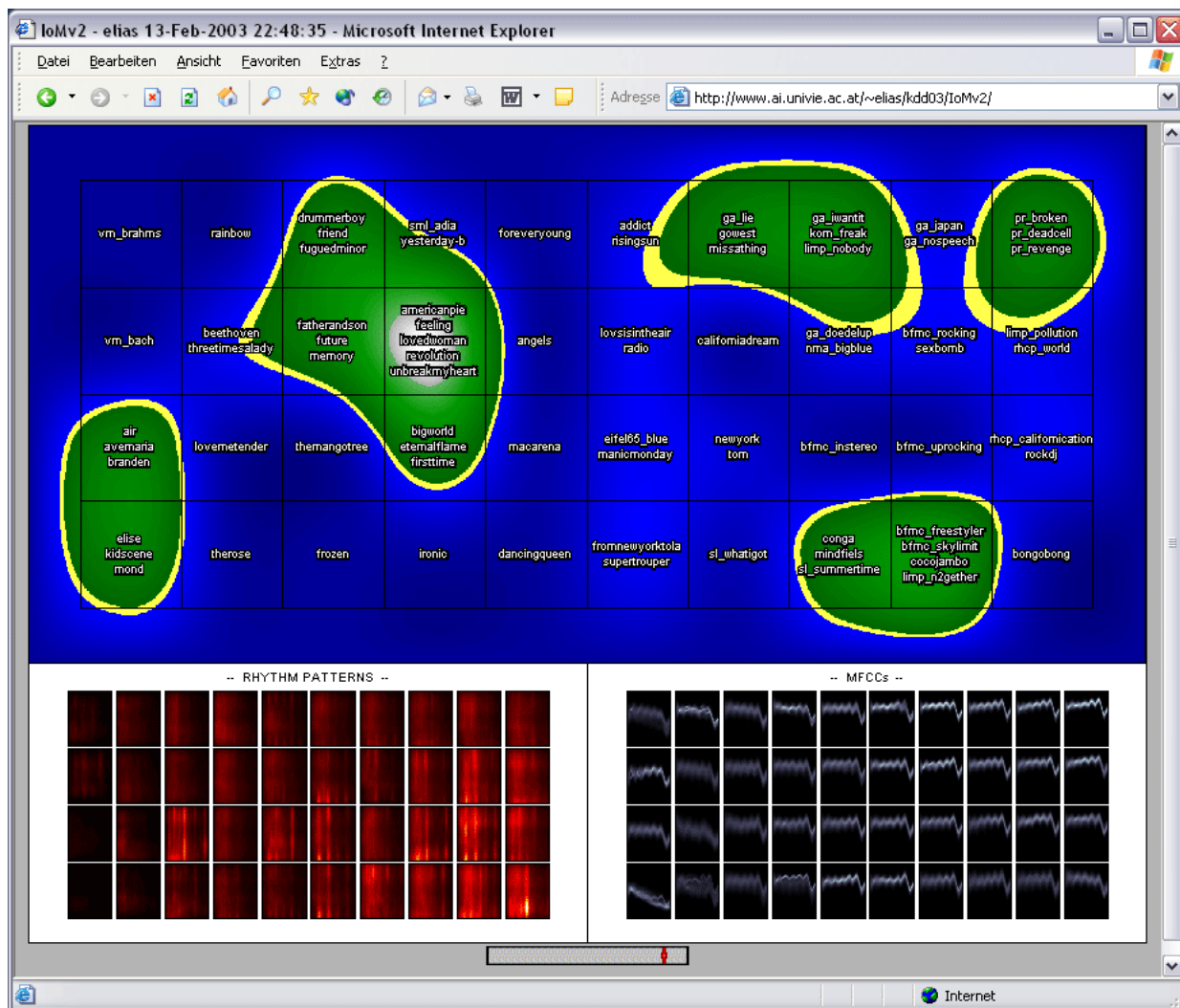


Figure 6: Screenshot of the user interface used for content-based exploration of music archives.

result is that the visualization can be used to study differences and similarities between different ways of computing similarity between music, i.e. studying how different ways of extracting features are related. Furthermore, preliminary results indicate that being able to change focus from one similarity aspect to another might be an interesting tool for browsing and exploring digital libraries of music.

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.

8. REFERENCES

- [1] C. C. Aggarwal. A Human-Computer Cooperative System for Effective High Dimensional Clustering. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2001.
- [2] C. C. Aggarwal. An Intuitive Framework for Understanding Changes in Evolving Data Streams. In *Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, San Jose, CA, 2002.
- [3] C. C. Aggarwal. Towards Effective and Interpretable Data Mining by Visual Interaction. *ACM SIGKDD Explorations*, 3(2):11–22, 2002.
- [4] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Spaces. In *Proceedings of the 8th International Conference on Database Theory (ICDT'01)*, London, UK, 2001.
- [5] J.-J. Aucouturier and F. Pachet. Music Similarity Measures: What's the Use? In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02)*, Paris, France, 2002.
- [6] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, 1998.
- [7] P. Cano, M. Kaltenbrunner, F. Gouyon, and E. Batlle. On the Use of Fastmap for Audio Retrieval and

- Browsing. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02)*, Paris, France, 2002.
- [8] G. DeBoeck and T. Kohonen, editors. *Visual Explorations in Finance*. Springer Verlag, Berlin, Germany, 1998.
- [9] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM*, 39(11):27–34, 1996.
- [10] V. Ganti, J. Gehrke, R. Ramakrishnan, and W.-Y. Loh. A Framework for Measuring Changes in Data Characteristics. In *Proceedings of the 18th Symposium on Principles of Database Systems*, 1999.
- [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] E. Kandogan. Visualizing Multi-Dimensional Clusters, Trends, and Outliers using Star Coordinates. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2001.
- [13] S. Kaski. *Data Exploration Using Self-Organizing Maps*. PhD thesis, Helsinki University of Technology, Department of Computer Science and Engineering, 1997.
- [14] S. Kaski. Fast Winner Search for SOM-Based Monitoring and Retrieval of High-Dimensional Data. In *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN'99)*, London, UK, 1999.
- [15] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen. WEBSOM – Self-Organizing Maps of Document Collections. *Neurocomputing*, 21:101–117, 1998.
- [16] S. Kaski and J. Sinkkonen. Metrics that Learn Relevance. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, Piscataway, NY, 2000.
- [17] T. Kohonen. Self-Organizing Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43:59–69, 1982.
- [18] T. Kohonen. The Adaptive-Subspace SOM (ASSOM) and its use for the Implementation of Invariant Feature Detection. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'95)*, Paris, France, 1995.
- [19] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Germany, 3rd edition, 2001.
- [20] 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, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585, 2000.
- [21] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. Number 07-011 in Paper Series on Quantitative Applications in the Social Sciences. Sage Publications, Newbury Park, CA, 1978.
- [22] K. Lagus and S. Kaski. Keyword Selection Method for Characterizing Text Document Maps. In *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN'99)*, London, UK, 1999.
- [23] J. Langner and W. Goebl. Visualizing Expressive Performance in Tempo-Loudness Space. *Computer Music Journal*, 2003 (in press).
- [24] B. Logan. Mel Frequency Cepstral Coefficients for Music Modeling. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, MA, 2000.
- [25] J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley and Los Angeles, CA, 1967.
- [26] 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>.
- [27] E. Pampalk, A. Rauber, and D. Merkl. Content-based Organization and Visualization of Music Archives. In *Proceedings of the ACM Multimedia*, Juan les Pins, France, 2002.
- [28] 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 Artificial Neural Networks (ICANN'02)*, Madrid, Spain, 2002.
- [29] A. Rauber. LabelSOM: On the Labeling of Self-Organizing Maps. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'99)*, Washington, DC, 1999.
- [30] J. W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.
- [31] 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)*, Dordrecht, Netherlands, 1990.
- [32] J. Vesanto and J. Ahola. Hunting for Correlations in Data Using the Self-Organizing Map. In *Proceedings of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99)*, Rochester, NY, 1999.
- [33] J. Vesanto and E. Alhoniemi. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000.
- [34] J. A. Walter and H. Ritter. On Interactive Visualization of High-Dimensional Data Using the Hyperbolic Plane. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, 2002.
- [35] G. Widmer. Using AI and Machine Learning to Study Expressive Music Performance. *AI Communications*, 14(3):149–162, 2000.
- [36] 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, 2002.
- [37] E. Zwicker and H. Fastl. *Psychoacoustics, Facts and Models*. Springer, Berlin, Germany 2nd edition, 1999.