Organic pie charts

Fabian Moerchen
Siemens Corporate Research, Integrated Data Systems
755 College Road East, Princeton, NJ, 08540, USA
fabian.moerchen@siemens.com

Abstract

We present a new visualization of the distance and cluster structure of high dimensional data. It is particularly well suited for analysis tasks of users unfamiliar with complex data analysis techniques as it builds on the well known concept of pie charts. The non-linear projection capabilities of Emergent Self-Organizing Maps (ESOM) are used to generate a topology-preserving ordering of the data points on a circle. The distance structure within the high dimensional space is visualized on the circle analogously to the U-Matrix method for two-dimensional SOM. The resulting display resembles pie charts but has an organic structure that naturally emerges from the data. Pie segments correspond to groups of similar data points. Boundaries between segments represent low density regions with larger distances among neighboring points in the high dimensional space. The representation of distances in the form of a periodic sequence of values makes time series segmentation applicable to automated clustering of the data that is in sync with the visualization. We discuss the usefulness of the method on a variety of data sets to demonstrate the applicability in applications such as document analysis or customer segmentation.

1. Introduction

Many problems in science and business involve high dimensional data that needs to be analyzed to support business tasks. Visualization can provide a high level overview of the similarity structure that aids in discovering groups of similar objects. The most commonly used visualization techniques can only display few of the dimensions together, for examples with grouped, stacked, and color-coded bar charts or a grid of pairwise scatter plots [8]. We present a visualization based on a one-dimensional non-linear projection with Emergent Self-Organizing Maps (ESOM) [18, 12] that supports interactive analysis of high-dimensional data. The data is displayed in a plot similar to pie charts [8] but with rugged lines that naturally emerge from the distance structure in the high dimensional space, hence the name *organic pie charts*. The visualization can be used to present high dimensional data in user interfaces to support interactive exploratory data analysis. We believe the analogy to simple pie charts makes this display easy to understand. We demonstrate the technique using datasets with different characteristics as they would also be found in many business applications (categorical data, numerical data, itemsets, documents).

2. Chart generation

The following sections describe the steps performed to generate organic pie charts in detail:
Projection: We train a one-dimensional ESOM to obtain a non-linear topology-preserving projection.
Visualization: The map nodes and data vectors projected on the map nodes are placed on the unit circle. The distances between neighboring map nodes lead to a sequence of values that defines the outline of the organic pie chart within the unit circle.
Clustering (optional): A multi-scale analysis of the sequence leads to a hierarchical clustering of the data that can be overlayed over the organic pie chart.

In order to demonstrate the individual steps and motivate the construction of the chart we will use the following dataset as a running example: We merged the training and testing parts of the Pendigits dataset from the UCI Machine Learning repository [2] that represents handwritten digits with 16 numerical features corresponding to 8 coordinate pairs of the trajectory. We only used the digits 2, 5, and 8. The latter two can be expected to be more similar to each other than when compared to the first one. The dataset consist of 3254 examples with all three classes almost equally represented. No preprocessing was performed as the data is already normalized.

2.1 Projection: Let $X = \{x_1, ..., x_l\} \subseteq R^n$ be a set of n-dimensional data vectors. Let $d : R^n \times R^n \mapsto R^+$ be a distance function between two data objects. A one-dimensional ESOM consist of an ordered list of k map nodes $m_i \in R^n$ $i = 1, ..., k$. Each map node thus connects a point in the high dimensional space with a position i on the map. The distance of the nodes on the map is defined by Equation 1.

$$n(i, j) = \min(j - i, i + k - j) \;\; i \leq j \tag{1}$$

The function n connects both ends of the one-dimensional map creating a circle of nodes. This avoids border effects when training the map. The vectors are initialized randomly using independent samples from a standard normal distribution assuming the data has been normalized to zero mean and unit variance in each dimension. Alternative initializations include samples from uniform distributions with the same range as the observed data or equidistant samples of a line representing the first principal component of the data in R^n.

The training of the map is performed in $e = 1, ..., E$ epochs. In each epoch the map is presented with the data vectors in random order. For each data vector x the index of the closest map node is determined with the function $b : R^n \mapsto \{1, ..., k\}$ defined by Equation 2.

$$b(x) = \arg\min_i d(x, m_i) \tag{2}$$

The vector $m_{b(x)}$ and its neighborhood as defined by n is then adjusted to be even closer to the data vector according to Equation 3 where e is the current epoch and $+/-$ are element wise vector additions.

$$m_i^e = m_i^{e-1} + \kappa_{\rho^e}(n(i, b(x)))\lambda^e(x - m_i^{e-1}) \;\; i = 1, ..., k \tag{3}$$

The function κ is a kernel that decays with increasing distance from $b(x)$. We use a Gaussian kernel that is scaled to have a standard deviation of $\frac{\rho^e}{2}$. For computational reasons we set it to zero outside of two standard deviations. Otherwise each update would change each map node. ρ^e is the current radius of the neighborhood that determines which map nodes are affected by Equation 3. The radius of the kernel should be large in initial training epochs and small in the final epochs. In addition, the learning rate λ^e should be cooled down. We use a linear decay between given start and end values for both the radius and the learning rate.

After the training of the map each data point is assigned to the closest map node using Equation 2 to obtain a topology-preserving projection of the high dimensional data onto one dimension.

Several parameters are needed for the training of the map but only one is crucial: the size of the map. It is important to use large maps (Emergent SOM) to obtain a representation of the data with a high resolution. Traditionally many researcher have used small SOM where one map node is interpreted as a cluster representing all data vectors assigned to it with the function b. ESOM represent clusters by regions of map nodes and there can be many interpolating nodes that do not have any data vectors assigned to them.

2.2 Visualization: Unlike MDS or PCA the projection performed by self-organizing maps is not distance or variance preserving but topology-preserving. Vectors close on the map tend to be close in the original space and vice versa. A more detailed view into the high dimensional distance structure is obtained by calculating the U-Matrix [17]. For each map node the average distance to its immediate neighbors is displayed as a height value. For two-dimensional maps this results in a landscape, in our one-dimensional case we obtain a sequence of distance values defined by Equation 4.

$$u_i = \frac{1}{2} \sum_{n(i,j)=1} d(m_i, m_j) \; i = 1, ..., k \tag{4}$$

For display purposes the distances are commonly normalized to a maximum of one. Figure 1 shows the sequence of normalized distance values for the Digit example. There are eight local maxima in the normalized distance sequence with values above 0.5. The maxima indicate that these parts of the map have been stretched out in the high dimensional space because there were few data vectors. Recall that the first node and the last node are immediate neighbors by definition. The high dimensional vectors projected to the left of the first maximum are thus very close to the vectors projected to the right of the last maximum in the distance sequence. The distances and the projected positions would therefore be better displayed on a circle. This can be accomplished by using a polar coordinate system similar to a radar chart where the map nodes are associated with equally spaced positions on the unit circle as defined by the first

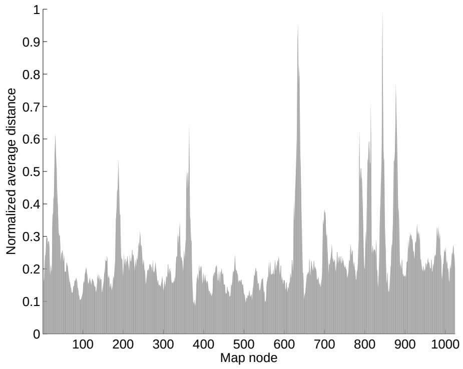

Figure 1. One-dimensional analog to the U-Matrix: Average distance to immediate neighbors on the map normalized to a maximum of one.

components of the points p_i defined in Equation 5. The second component is the radius and is defined as one minus the normalized distance value.

$$p_i = \left\langle \frac{2\pi i}{k}, 1 - \frac{u(i)}{\max(\{u(i)|i = 1, ..., k\})} \right\rangle \qquad (5)$$

The resulting plot is shown in Figure 2 for the Digit example. The black lines indicate the projected position of the training examples. The previously separated ends of the distance sequence are now connected at 3 o'clock. The local maxima of the distance sequence cut deep into the unit circle defining pie segments that correspond to homogeneous clusters of data vectors. Another effect of the polar coordinate system is that more details of the local distance structure are shown for regions with small distances, i.e. inside clusters, as opposed to peaks of the distance sequence that represent sparse regions between clusters. This aids in discovering hierarchical cluster structures where a larger cluster might contain several smaller sub clusters with less distinct separations. A non linear scaling of the distance values could be used to (de-)emphasize local or global distances. In Figure 3 we show a manual segmentation of the pie into five large and 3 small pieces with dashed lines. Outside of the unit circle we display the mean value of all data vectors projected to a segment in form of trajectories. Each segments represents a trajectory that clearly corresponds to one of the digits in the dataset. We further analyzed the frequency of the ground truth classes and display the majority class inside the pie. We note in passing that the class information was not used in the training of the map and that only 8 examples (0.25%) ended up in segments with a different majority class. The segmentation of Figure 3 shows that the digit classes 5 and 8 are split up into two and 5 clusters, respectively. This is caused by different ways of writing the digits represented depending on where the writer started.

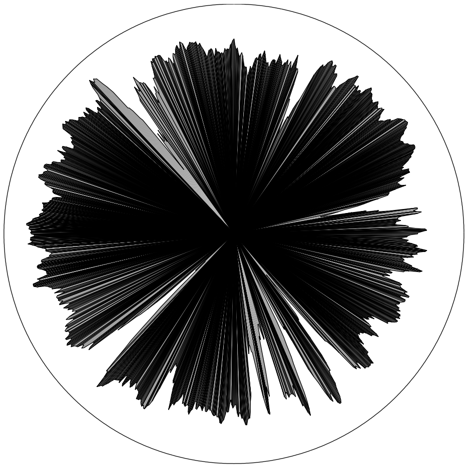

Figure 2. Organic pie chart for Digits: The map positions are equally spaced on the unit circle, the normalized distance value is subtracted from the unit circle to define the shape. The dark lines indicate projected data points.

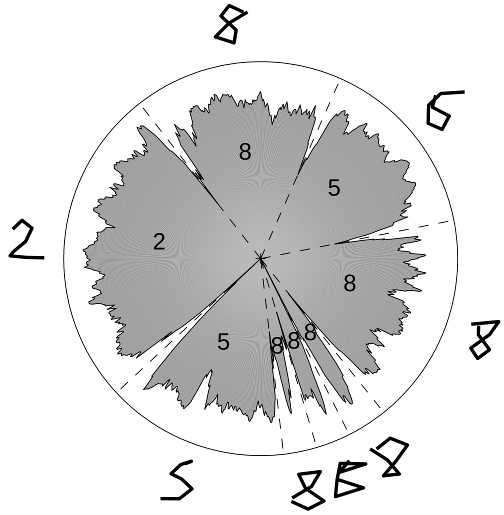

Figure 3. Organic pie chart for Digits with mean trajectories (outside) and majority class (inside) for manually defined pie segments (dashed). Each trajectory represents almost exclusively digits from a single class.

2.3 Clustering: The sequence u_i of distance values is an abstraction of the distance relations in the high dimensional space and helps to visually identify coherent regions in the organic pie charts. Using algorithms for the segmentation of a series of values we can automatically determine a hierarchical clustering of the data. Many different methods for time series segmentation have been proposed [11]. Commonly the goal is to derive a simplified representation for the series that needs less storage space, removes noise, can be efficiently indexed, and so forth. The series is approximated on each segment using a low order polynomial. Here, we are concerned with significant features of the time series, local maxima in particular. We propose to use scale space methods [13] to evaluate the significance of the local maxima. This has been used in [3] to generate hierarchical segmentations of time series. The first step is a multi scale analysis using the Continuous Wavelet Transform (CWT) [13] that correlates signals with scaled and shifted versions of a continuous differentiable mother wavelet with compact support. It generates a two-dimensional intensity image that indicates at which positions and which scales the input signal is very similar to the wavelet function. The CWT for the Digit example calculated with the WaveLab toolbox [5] using a Gaussian wavelet and is shown in Figure 4. Dark shades indicate large wavelet coefficients. The most significant local maxima are those that persist when analyzing the signal at larger scales, i.e., they are still local maxima even when locally averaging the sequence with a Gaussian function of increasing variance and support. They are shown as dark vertical lines that extend from the position of the local maximum in the sequence downward in the scale space plot. The concept of local maxima in the scale space is formalized in the Wavelet Transform Modulus Maxima (WTMM) [13] that identifies local maxima at each scale and connects the maxima from several scales corresponding to the same feature in the original sequence. The maxima lines for the Digit example are shown in Figure 5. The length of a line can be interpreted as the significance of the feature in the value series. The longer a maximum persist in the scale space, the more prominent it is. Note that the position of a maximum shifts over the different scales, the origin of each line at the top of the lower plot in Figure 5 indicates the exact position of the maximum in the sequence. The circular neighborhood of the map is a natural fit for the wavelet analysis methods, that assume periodic signals. The maxima lines of the WTMM provide a hierarchical clustering of organic pie charts: The two most significant lines define a partition of the pie (and thus the projected data) into two clusters. Each additional maximum splits one cluster into two parts increasing the total number of clusters by one. The segments defined by the first 10 local maxima ordered by the length of the corresponding WTMM lines are shown in Figure 6 with their ranks. The cuts are similar but not identical to the manual segmentation used in Section 2.2. The first cut does not correspond to the largest distance but is placed in the area where four local maxima are relatively close. For larger scales the four close local maxima collectively contribute to this maximum in the scale space. This causes one of the corresponding WTMM lines to persist very long in the scale space (between 800 and 900 in Figure 5). Which of the four lines eventually survive while the others stop is a property that emerges from the data. The next four cuts (2-5) were also selected manually in Figure 3. The cuts 6 through 9 split existing larger clusters even though the cut number 10 corresponds to a larger distance. Again, this is an effect of the scale space analysis: The maximum of cut number 10 is overshadowed by the maximum of cut number 1. The visually less distinct maxima at cuts 6-9 are selected first because they are far away from larger maxima.

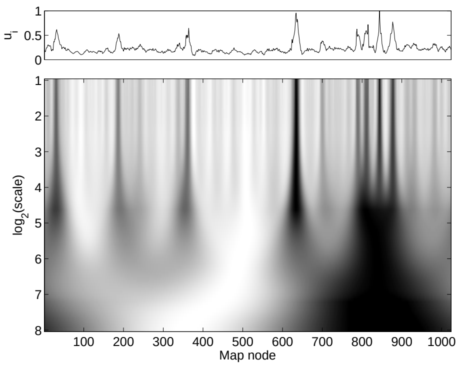

Figure 4. Continuous Wavelet Transform of distance sequence u_i.

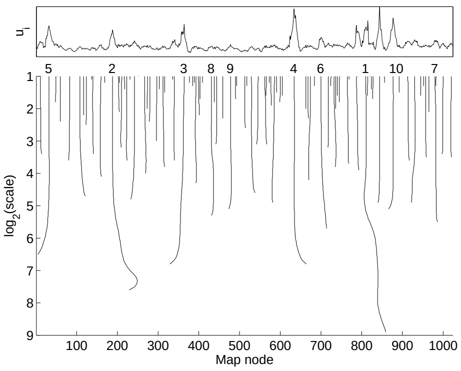

Figure 5. Wavelet Transform Modulus Maxima of distance sequence u_i. The numbers indicate the ranks of the 10 most significant maxima used for the clustering in Figure 6.

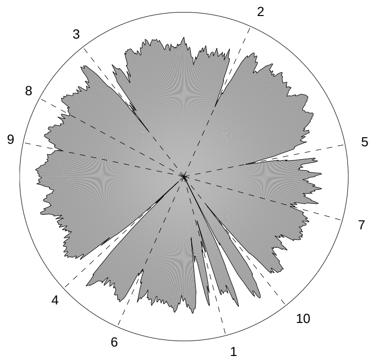

Figure 6. Organic pie chart for Digits with clusters defined by the ten most significant maxima of the distance sequence. The numbers indicate the maxima of Figure 5.

3. Examples

The following examples demonstrate the applicability of the methods to different domains. We explore capabilities and annotations of organic pie charts.

3.1 Documents: As an example of sparse numerical data we selected the LA12 dataset that represents news paper articles from the L.A. Times. We calculated TFIDF vectors normalized to unit length and selected a subset of 354 documents that contained the word 'bush' for display on the map. The cosine distance was used and all map nodes were normalized to unit length after each update. The organic pie chart shown in Figure 7 is automatically segmented into 20 clusters. Each segment is annotated with the majority class and the 6 most important word stems (obtained by sorting the entries of the centroid vector of all documents in a segment in decreasing order). The class labels indicate a successful global organization of topics: only the National category is present in two different areas of the map. The articles of each category have been grouped into more specific topics, e.g., for the Foreign category there are segments for Japan, Korea, China, Soviet Union, Libya, Israel, and Nicaragua.

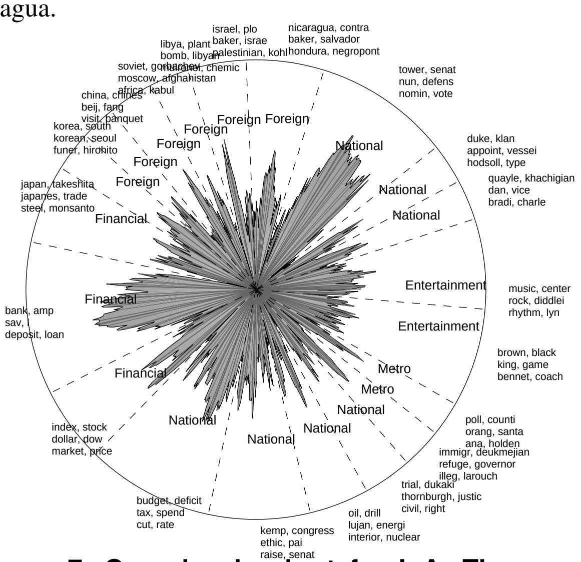

Figure 7. Organic pie chart for L.A. Times data with 20 segments. For each segment the majority class is shown inside the circle and the most important word stems around the chart.

3.2 Itemsets: The dataset Zoo from the UCI repository [2] is commonly analyzed with itemset algorithms. We preprocessed it similarly to the documents: Each itemset was converted to a binary vector where each position represents one item. Each vector was normalized to unit length and the cosine distance was used for the chart. Each map node was normalized to length one after every update. Other distances for itemsets could be used alternatively but a concept of a weighted centroid is required to update the map nodes. The Zoo dataset describes 101 creatures in 7 classes with mainly binary attributes. The chart in Figure 8 is annotated with symbols on the unit circle indicating the class and with labels for the animal names. The large segments of the pie correspond to the known classes, e.g., the birds at 12 o'clock or the insects at 2 o'clock. There is also reasonable organization within the classes. For example in the insects class there are three pairs of animal assigned to the same node, respectively: honeybee and wasp, moth and housefly, flea and termite. In the big mammals class in the lower left there is a group of African animals (antelope, buffalo, ...) and a group with mainly wild cats (cheetah, leopard, lion, lynx, ...).

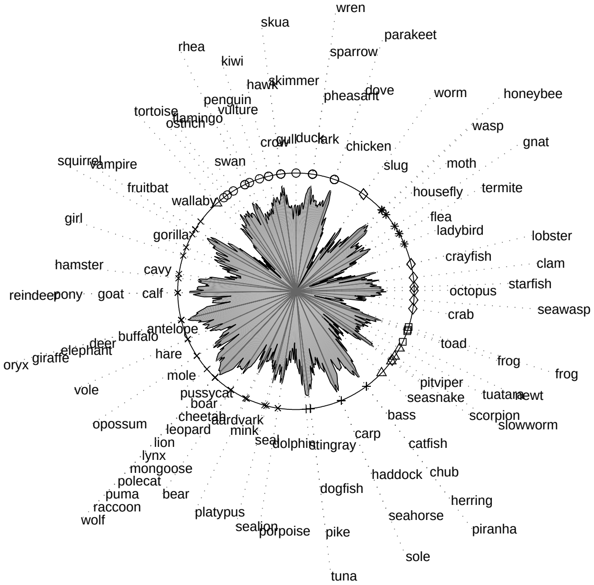

Figure 8. Organic pie chart for Zoo data. There is global organization of the classes (symbols on the unit circle) as well as meaningful grouping within each class.

4. Related work

Popular methods for 2D projection of high dimensional data include the variance preserving PCA and distance-preserving Multi-dimensional Scaling (MDS)[14]. Other non-linear methods include FastMAP [7], GTM [4], and ISOMap [16]. Two-dimensional self-organizing maps (SOM) [12] are popular for the projection of high dimensional data. The visualization of Self-Organizing Maps is usually performed using two-dimensional maps and component planes [20] for correlation analysis, U-Matrix [17] for distance analysis or P-Matrix [19] for density analysis. In [1] the OPTICS algorithm for ordering a dataset according to the intrinsic density structure is described. A visualization similar to Figure 1 is used to display hierarchical cluster structures for low dimensional data based on local density properties. For high-dimensional data a circular display with fixed segments representing attributes is proposed. This is the most common use of circular visualization techniques such as Circular Parallel Coordinates, RadViz, or Circle Segments (see [10, 9] and references therein) and recently DataRoses [6]. In contrast we use circle segments of varying size that represent clusters.

5. Discussion

The proposed organic pie charts offer a view into the high dimensional space that emerges from the data and can reveal hierarchical cluster structures present in the data set. The display offers analogies to the well known pie charts including the fact that the size of the segments is related to the number of projected data points. The multi-scale analysis of the organic pie charts can be used to automatically define a hierarchy of clusters in the data that corresponds to the visualization. The display offers room for additional annotations such as manual segmentations, distribution of classes if available, and displays of data vectors or prototypes per segment.

The complexity of the methods involved in the creation of organic pie charts are linear in the size of the data but with a large constant. The training of the map is linear and all subsequent steps (calculating the distances, CWT, WTMM) only depend on the size of the map not the size of the data. All these computations can be performed offline to prepare the visualizations for the user. Nevertheless, the iterative training of the SOM with many nodes can be time consuming. Several approaches have been proposed to increase the scalability by using parallel processing for map training, e.g., [15].

6. Summary

Organic pie charts provide an intuitive visualization of high dimensional data that supports interactive knowledge discovery. The shape of the pie emerges from the underlying structure of the dataset. The relation to the well known pie charts makes this a promising tool for use by both experts and laymen in applications such as customer segmentation or scouting where large amounts of patents and other documents need to be analyzed. The scale space analysis of the pie charts leads to an automated hierarchical clustering of the data that is in sync with the visual representation and not a black box result as would be obtained from many high dimensional clustering algorithms. We demonstrated the capabilities on several datasets from different domains to show the wide applicability.

References

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In *Proc. ACM SIGMOD*, 1999.

[2] A. Asuncion and D. Newman. UCI Machine Learning Repository. University of California, Irvine `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[3] B. R. Bakshi and G. Stephanopoulos. Reasoning in time: Modeling, analysis and pattern recognition of temporal process trends. In *Chemical Engineering: Paradigms of Intelligent Systems in Process Engineering*, volume 22, pages 485–547, 1995.

[4] C. M. Bishop, M. Svensen, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[5] D. Donoho, A. Maleki, and M. Shahram. WaveLab 850. University of California, Irvine. `http://www-stat.stanford.edu/~wavelab`.

[6] N. Elmqvist, J. Stasko, and P. Tsigas. DataMeadow: a visual canvas for analysis of large-scale multivariate data. *Information Visualization*, (7):18–33, 2008.

[7] C. Faloutsos and K.-I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proc. ACM SIGMOD*, pages 163–174, 1995.

[8] U. Fayyad and W. A. Grinstein, G.G. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, 2002.

[9] M. Ferreira de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *IEEE Trans. on Visualization and Computer Graphics*, 9(3):378–394, 2003.

[10] D. Keim. Information visualization and visual data mining. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[11] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In *Data Mining In Time Series Databases*, pages 1–22. World Scientific, 2004.

[12] T. Kohonen. *Self-Organizing Maps*. Springer, 1995.

[13] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[14] G. A. F. Seber. *Multivariate Observations*. Wiley, 1984.

[15] B. Silva and N. C. Marques. A hybrid parallel SOM algorithm for large maps in data-mining. In *Proc. Portuguese Conf. on Artificial Intelligence*, 2007.

[16] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[17] A. Ultsch. Self-organizing neural networks for visualization and classification. In *Proc. Conf. of the German Classification Society*, pages 307–313, 1992.

[18] A. Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In *Kohonen Maps*, pages 33–46, 1999.

[19] A. Ultsch. Maps for the visualization of high dimensional data spaces. In *Proc. Workshop on Self-Organizing Maps*, 2003.

[20] J. Vesanto. SOM-based data visualization methods. *Intell. Data Anal.*, 3(2):111–126, 1999.