DIPLOMARBEIT

APPLICATION OF SELF-ORGANIZING MAPS TO A POLITICAL DATASET

ausgeführt am
Institut für Softwaretechnik und Interaktive Systeme
der Technischen Universität Wien

unter der Anleitung von
ao.Univ.Prof. Dr. Andreas Rauber

durch
Georg Pölzlbauer
Servitengasse 19 / 5
1090 Wien
Matrikelnummer 9725498
Wien, Februar 2004

Zusammenfassung

Der Global Democracy Award ist eine zum Zeitpunkt des Schreibens dieser Diplomarbeit im Entstehen begriffene Auszeichnung. Diese soll jährlich an das Land verliehen werden, das die größte Verbesserung seiner Demokratiequalität aufzuweisen hat. Die Messung der Demokratiequalität und deren jährliche Änderung liegt im Forschungsbereich der Politikwissenschaften und ist nicht Thema dieser Arbeit; ausgehend vom "Pilot Ranking", einem ersten Modell zur Bewertung von 100 demokratischen Ländern nach allgemein zugänglichen Kriterien und Indikatoren, wird mit Hilfe von Kohonen's Self-Organizing Map Algorithmus die Struktur dieser Daten untersucht. Dabei geht es vor allem darum,ob die Resultate mit den zu erwartenden übereinstimmen. Da die erhobenen Daten des Pilot Ranking durchaus verschiedene Erwartungshaltungen implizieren, wie beispielsweise "die Demokratiequalität von Schweden und Norwegen ist sicherlich höher als die von Liberia", können die Ergebnisse der Self-Organizing Map mit diesen Annahmen verglichen werden.

Es liegt ein besonderer Schwerpunkt auf der Darstellung und Präsentation der Karte, zu diesem Zweck werden eine Reihe von Visualisierungsmethoden kurz vorgestellt und angewandt.

Weiters wird die Self-Organizing Map mit den ihr ähnlichen Verfahren aus angrenzenden Bereichen, wie Vektor-Quantisierung und Vektor-Projektion, verglichen. Diese Gegenüberstellung verdeutlicht die Vorzüge und auch die Grenzen dieses Algorithmus. Außerdem werden aufbauend auf der trainierten Karte gängige Methoden der Clusteranalyse angewandt, wodurch die Karte in homogene Bereiche gegliedert wird, was unter anderem durch die Analogie zu politischen Ländern zu sehr schönen Ergebnisse führt. Es werden unter anderem auch Verfahren besprochen, die die fehlenden Werte der Datentabelle behandeln.

Zuletzt werden die Daten nach bestehenden Gesichtspunkten untersucht, zum Beispiel nach der Verteilung der NATO-Länder auf der Karte. Außerdem wird ein Maß zur Bewertung der Demokratiequalität auf diese Karte angewandt, das auch eine feinere Bewertung nach Gesichtspunkten wie "Qualität des Gesundheitssystems" oder "Bildungswesen" zuläßt.

Abstract

The Global Democracy Award, which is developed at the time of this writing, is a prize to be awarded annually to the country that achieves the biggest improvement of its democratic quality. Measurement of this democratic quality and its annual change lies within the research field of political sciences and is not within the scope of this work; starting from the "Pilot Ranking", which is a first model to evaluate and rate 100 democratic countries by publicly accessible indicators and criteria, the structure and internal properties of this data manifold is investigated with the aid of Kohonen's Self-Organizing Map algorithm. The emphasis lies on novelty detection and comparison of results with expectations and bias on the data set. Since the data manifold from the Pilot Ranking implies certain anticipations, for example "the democratic quality of Sweden and Norway surely is better than Liberia's", it can be investigated whether the results from the Self-Organizing Map match these expectations.

The presentation techniques of the findings of the map will be another emphasis of this thesis. A range of visualization methods will be introduced and applied to the Democracy map.

Then, the Self-Organizing Map will be compared with algorithms from related fields, like vector quantization and vector projection. This will show the advantages as well as the limitations of this method. Also, the already trained map will be subjected to well-established clustering methods, so it is partitioned in homogenous regions, which leads to very nice results partly due to the analogy of the samples being political countries. Also, different methods of filling missing data points in the original data table will be discussed.

At last, the data set will be investigated by means of real-world considerations, for example by the distribution of the NATO-countries on the map. Also, a measure for democratic quality will be applied to the map. This measure allows evaluation at fine levels of detail by measuring criteria like "quality of the health care system" or "educational system".

Table of Contents

Zusammenfassung **......** ***ii***

Abstract **......** ***iii***

1. Introduction **......** ***1***

1.1. Overview 1

1.2. Related Work 2

1.3. Software and tools used in this thesis 4

1.4. Organization of this thesis 4

2. The Data Set **......** ***6***

2.1. The Countries 6

2.2. The Variables 7

2.2.1. Overview 7
2.2.2. Characteristics of the data matrix 8
2.2.3. Weights and different influence of individual dimensions 9

2.3. Benchmark Data Set 10

3. The Self-Organizing Map **......** ***11***

3.1. Vector Quantization 11

3.2. Vector Projection 13

3.3. Introduction to the SOM 17

3.4. SOM Training 18

3.5. Initialization and Parameterization of the Self-Organizing Map 21

3.6. Dealing with missing values in the data set 24

3.6.1. SOM training with data sets that contain missing values 24
3.6.2. Interpolation with BMUs 25
3.6.3. The Democracy Ranking approach to fill missing values 26

3.7. Parameters for the experiments 26

3.8. Quality measures of the SOM 27

4. Visualization of the SOM **......** ***30***

4.1. Overview and introduction to visualization 30

4.1.1. Importance of visualization 30
4.1.2. Plots that show a single value per map unit 30
4.1.2. Plots that show multiple values per map unit 33

4.2. Component planes 36

4.3. Distance Matrices 38

4.4. Hit Histograms 41

4.5. Labels **44**
4.6. The LabelSOM method **46**
4.7. Smoothed Data Histograms **46**
4.8. Fuzzy response and hit trajectories of an individual sample **49**
4.9. Projections of codebook vectors **51**
4.10. Attempts to combine several component planes in one plot **53**
5. Clustering of the SOM *56*
5.1. Introduction to clustering **56**
5.2. k-means clustering **57**
5.3. Hierarchical Clustering **60**
5.4. Cluster validity **65**
5.5. Growing Hierarchical SOMs **69**
6. In-depth discussion of the Democracy SOM *75*
6.1. Overview **75**
6.2. Visualizing the map according to continent distribution **75**
6.3. Hit histograms by treaties **77**
6.4. Computing a "score" for qualities within indicator categories **78**
7. Conclusion *83*
Appendix A: Countries and Data Collections *86*
List of Countries **86**
List of Indicators **87**
Political System 87
Gender Equality (Educational and Economic) 87
Economy 88
Health 88
Knowledge 88
Environmental Sustainability 89
Glossary *90*
Bibliography *91*

1. Introduction

1.1. Overview

The initial idea behind this thesis is applying the Self-Organizing Map algorithm to a real-world data set. This data set is part of a project that measures and directly compares the democratic quality of countries based on a number of indicators, the "Pilot Ranking" [Cam02, Cam03] of the Global Democracy Award[1]. While the Democracy Award aims at comparing the annual differences between the rankings and determining a winner country that has increased its democratic score by the largest amount, this thesis intends to examine the data set from a data mining point of view. At the time of the writing of this document, the GDA is still under development. The ranking is made up of 100 countries that are characterized by 60 indicators organized in 6 categories (Political System, Health, Environmental Sustainability, Economics, Gender Equality, and Knowledge). The selection of these variables has been performed by David Campbell and Miklos Sükösd and is still subject to further research in the area of political sciences (for a comprehensive list of these indicators, see Appendix A). The ranking can be viewed as a 100-by-60 table with several missing values, which is the basis for further computation and analysis.

The actual scope of this thesis is to discuss what can be learned from the Democracy data set with the help of the Self-Organizing Map. Particularly, various post-processing techniques will be discussed, with further emphasis on recent developments and extensions to the SOM algorithm. "Post-processing" means the interpretation of an already trained map. Since the SOM, as any other a neural network, does not have any inherent visualization, but can be visualized in very meaningful and intuitive ways because of its map-like structure (hence the name), several methods to perform this will be discussed. Another way to post-process the SOM is to cluster the resulting map into compact regions of similar data.

The SOM is a very versatile method, it is related to very diverse fields of research like vector quantization, vector projection, artificial neural networks,

---

[1] http://www.global-democracy-award.org

and unsupervised learning. The SOM can be compared to either of these; some of its properties will be discussed in Sections 3.1 and 3.2.

Interesting insights can be obtained by comparing the experimental findings from the various SOM techniques with the results from the Pilot Ranking. These attempts to compare the SOM model and the statistical ranking approach will be performed whenever they are applicable. For example, the issue of dealing with missing value is addressed by the Democracy Award with an averaging scheme described in Section 3.6, while the SOM provides its own interpolation method. Also, since one of the main advantages of the SOM is to group similar things together, it is interesting to compare the regions on the map with real-world categories like geographical location of the countries.

1.2. Related Work

Since the Democracy Award is a novel attempt to measure the quality of democracies based on indicators, there are currently very few comparable works available. However, data mining techniques like the SOM or multi-dimensional scaling have been applied successfully to all categories the Democracy data set consists of (like Health, Environment, etc.). Also, distantly related are studies of text collections with political contents, which have been analyzed with the SOM, especially the CIA World Factbook, on which parts of the Pilot Ranking indicators rely on, as has been investigated in [Mer98].

Another comparable piece of work is the Poverty Map[2], an application of the SOM that also shows a map of the world based on mostly economic indicators [Deb98].

Figures 1 and 2 show the resulting map as a Self-Organizing Map and a world map colored with values obtained from the SOM evaluation (the SOM algorithm will be described in Chapter 3). The Poverty Map was obtained by 39 indicators selected from the World Bank Development Indicators [WBDI01]. The same source was used for many of the Pilot Ranking variables, see Chapter 2.

---

[2] http://www.cis.hut.fi/research/som-research/worldmap.html

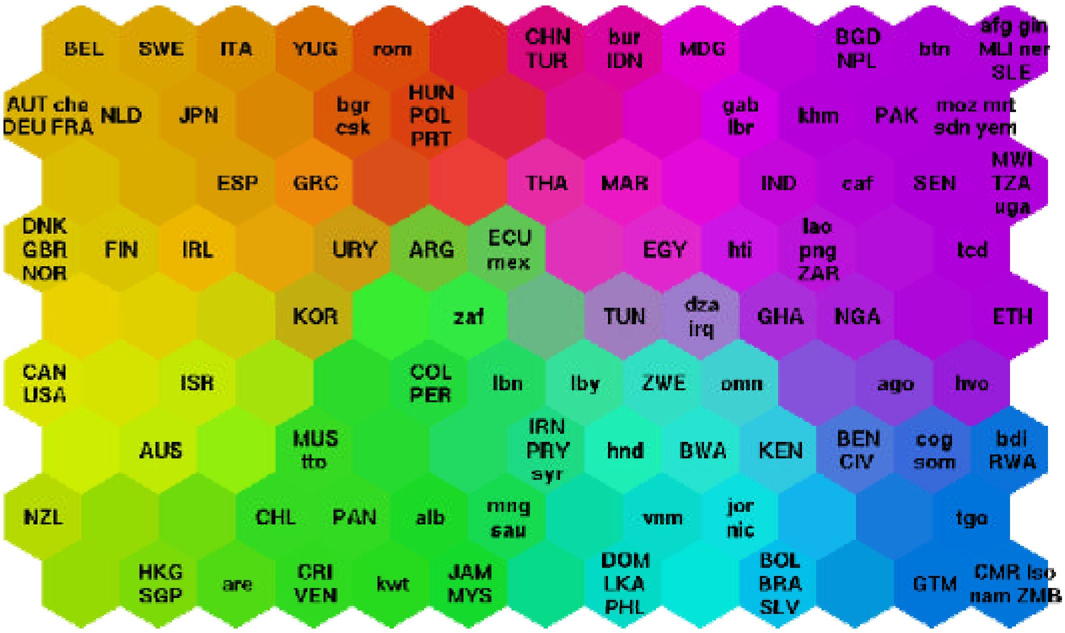

Figure 1: World Poverty SOM

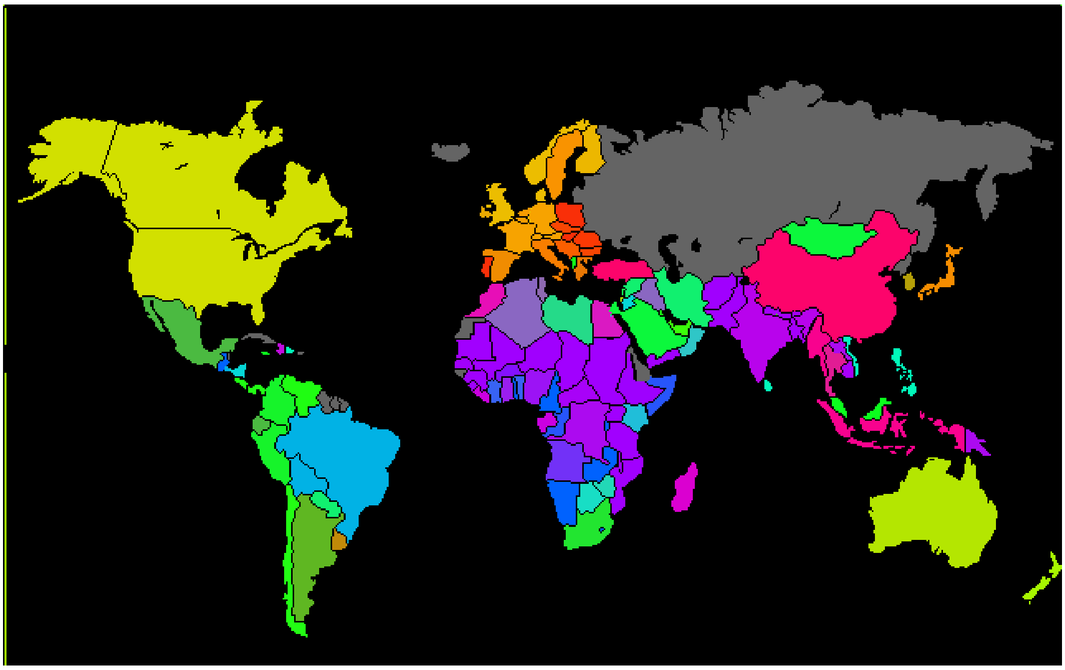

Figure 2: World Poverty Map

1.3. Software and tools used in this thesis

Most of the calculations described in this thesis have been performed with Mathworks' Matlab 6.5[3], a software tool and computer language for scientific computing. It was also used for the plotting the figures. For the most part, the SOM toolbox for Matlab[4] [Ves99] has been used. The results found in this thesis rely heavily on this package and its strong visualization capabilities. For computation and visualization of the smoothed data histograms described in Section 4.7, the SDH toolbox[5] was used. Section 5.5 was written with the help of the GHSOM toolbox[6].

1.4. Organization of this thesis

The rest of this thesis is organized as follows:

- Chapter 2 ("**The Data Sets**") provides an introduction to the Democracy data set, which will be the basis for the rest of this work. It presents an overview of the countries that are evaluated in the Democracy Award Pilot Ranking and the indicators that measure their democratic value. Several ways to categorize the countries will be shown, like grouping by continent. Another very important concept will be introduced, namely the distance metric that measures how similar or dissimilar two countries are, taking the relative importances of the indicators into accound. Also, a benchmark data set, the Iris data set, is presented, which is a commonly used in data mining literature. The Iris data will be applied to introduce the different methods used to analyze and visualize the data, because it is by far less complex than the Democracy data set.
- Chapter 3 ("**The Self-Organizing Map**") introduces the Self-Organizing Map (SOM), the unsupervised learning algorithm the Democracy data is analyzed with. Several ways the SOM can be parameterized are explained. The SOM will be compared to similar

---

[3] http://www.mathworks.com

[4] http://www.cis.hut.fi/projects/somtoolbox

[5] http://www.oefai.at/~elias/sdh

[6] http://www.ai.univie.ac.at/~elias/ghsom

methods that perform either vector quantization or projection. Also, the way missing values in the data set are handled by the SOM will be discussed. Finally, a summarization is presented of how exactly the Democracy SOM is computed (including all parameterizations etc.)

- Chapter 4 ("**Visualization of the SOM**") will present a wide range of common and recently developed visualization methods that can be applied to the SOM. These will be introduced by means of the relatively simple Iris SOM, and then applied to the more complex Democracy map. Visualizations include distance matrices, component planes, several labeling methods, hit histograms, smoothed data histograms, and codebook projections of the codebook vectors.
- Chapter 5 ("**Clustering of the SOM**") presents several commonly used clustering methods. Using clustering as a post-processing method divedes the SOM into coherent regions. These partitions can then be visualized in a meaningful way, such that similar areas on the map are revealed. Also, quality measures for different clustering methods are presented, so the validity of the partitions can be analyzed and compared. This chapter also describes a recently developed variant of the SOM that uncovers hierarchical structure in the data, the Growing Hierarchical SOM.
- Chapter 6 ("**In-depth discussion of the Democracy SOM**") describes a series of experiments that are a direct application of the visualization and clustering techniques introduced in the previous two chapters. Visualizations are proposed that exploit the unique characteristics of this data set, like the high correlation between the variables, a-priori knowledge about the countries (like membership of a certain treaty), or reduction of several similar (in terms of interpretation from the Democracy Award) variable dimensions to a single component. Also, an attempt to approximate the score of the Pilot Ranking is proposed, and several visualization methods are linked and displayed in the same plot to provide interesting insights into the structure of the data set.
- Chapter 7 ("**Conclusion**") summarizes results and findings made in this thesis, and provides an outlook for further investigations and work.

2. The Data Set

2.1. The Countries

The data set introduced in [Cam03] covers 100 countries which are examined for their democratic quality. These have been chosen by the Global Democracy Award (GDA) as a basis for the initial ranking. The countries regarded by the GDA have to be classified as either "Free" or "Partly Free" by the Freedom House (in the publication that covers the years from 1997 to 2001) [FH01]. Countries that are classified "Non Free" are not considered, since its aim is to rank democracies and not any other kinds of regimes. Also, small countries (with a population of 1 million people or less) are excluded because it would be problematic to compare them to the rest of the countries based on the same indicators. Further, countries that are not included in the World Development publication in [WBDI01] (like Macao or Taiwan) are not considered, which are mainly countries that are not recognized by the majority of the global community, or countries that are classified as "related territories" (like Hong Kong or Puerto Rico). For the complete list of countries, see Appendix A.

In the later chapters of this thesis, it will be interesting to compare the results from the SOM algorithm to real-world categories, thus the following classifications are used:

- geographical category: grouped by continent
- political category: membership of economic and military treaties

The treaties that will be investigated are:

- APEC - Asia-Pacific Economic Cooperation
- AU - African Union
- CE - Council of Europe
- EU - European Union (15 countries, before May 2004 enlargement)
- NATO - North-Atlantic Treaty Organization
- OAS - Organization of American States
- OECD - Organization for Economic Co-operation and Development
- OIC - Organization of the Islamic Conference
- OSCE - Organization for Security and Cooperation in Europe

2.2. The Variables

2.2.1. Overview

To describe the democratic quality of the countries that are covered by the pilot ranking, several indicators (variables) have been selected. The authors of the Democracy Award ranking describe the task of selecting these indicators in [Cam02]. Among others, the following criteria have to be met: "Minimizing Ideological Bias in the Indicator Selection" and "Minimizing Cultural Bias in the Indicator Selection". Finally, the pilot ranking consists of 60 variables from the following categories:

- **Politics (P, 8 indicators)**: describing political rights and civil liberties, such as how often changes of government take place. This is by far the most important category.
- **Gender Equality (G, 13 indicators)** (Educational and Economic): indicates the degree of equal distribution of opportunities for individuals who live in that society. Most of the variables in this group directly compare male/female indicators from educational and economic categories.
- **Economy (E, 9 indicators)**: a competitively performing economy is sometimes regarded as a necessity for a working democracy, and expresses a functioning interaction between politics (government) and the economy. Indicators from this group include unemployment rate, budget deficit, and GDP per capita.
- **Health (H, 10 indicators)**: variables from this group provide an overview of health care, which is considered a prerequisite for modern democracies and effectiveness of a social policy. Indicators include life expectancy, health expenditure, and hospital beds (per capita).
- **Knowledge (K, 15 indicators)**: express the "maturity" of a democratic society or how "advanced" the society is. Indicators from this group include school enrollment, illiteracy rate, and R&D expenditure.
- **Environmental Sustainability (En, 5 indicators)**: criteria in this category emphasize criteria of sustainability and, more specifically, the long-term effectiveness of a government policy. Indicators include CO_2 omission and energy use.

Each of these categories has a different number of indicators (between 5 and 15). These have been selected to provide the basis for a qualitative ranking, so their intention is to provide a score that reflects how "good" or "bad" e.g. Albania's economic system performs. However, it is not the scope of this thesis to provide a qualitative measure of democracies, but rather to detect similarities between them; therefore, the original raw data will be used. This is also a fundamental difference to the pilot ranking, which computes a "score" from the data set through a series of (mostly linear) transformations and calculations, whereas the algorithm used in this thesis is primarily used to analyze the internal structure of the data.

The data can be organized as a 100-by-60-matrix, where the row vectors represent 100 individual countries (in alphabetic order) and the column vectors contain the 60 indicators (variables) that will be addressed by the category's abbreviation followed by the index within this category, e.g. "G3" refers to the third indicator in the "Gender" category, which is "Employees, services, female". For a comprehensive list of the indicators, see Appendix A.

2.2.2. Characteristics of the data matrix

Since not all of the required indicators are available for each country, the data matrix contains missing values (MV). The number of MVs varies strongly between the indicators and ranges from 0% to 56%. However, the SOM can deal with MVs very efficiently, as described in Section 3.6, where it is compared to the pilot ranking approach. One additional problem is that less-developed countries tend to have more missing values than for example European countries. This leads to the situation that the countries in the lowest third of the scale are more dependent on the interpolation method that is used to fill MVs.

As mentioned in the previous section, the indicators have been selected to reflect how "good" a country performs in a specific category. The Democracy Award assigns a score to each of the variables (ranging from 0 to 100), where a higher score always means "better". To achieve this, a linear transformation is performed, either assigning a score of 100 to the country with the highest raw data value and the lowest 0 (the higher, the better), e.g. H7 (life expectancy at birth) or the other way around (the lower, the better), e.g. H8 (mortality rate, infant). This is, however, not the way that data is handled in this thesis, rather the variables are normalized to zero mean and unit variance, because this is a more common approach taken in data analysis applications.

2.2.3. Weights and different influence of individual dimensions

To achieve a meaningful overall impression of the quality of democracies, the indicators have to be weighted individually to reflect their relative importance. This weighting is performed in two steps:

- based on **category**: the "Politics" category (P) is the most important with an influence of 50 %; the other 5 categories all have an influence of 10 % each. Thus, the indicators have to be weighted in a way that the number of indicators within a category does not influence the importance of the category itself, so indicators within a category with a total of 5 indicators are weighted higher than variables within a category of 10 indicators.
- based on **dimension**: within a category, indicators can be of different importance, e.g. in the "Health" category, H7 (life expectancy at birth) is the single most important characteristic, so it is counted as 50% of Health's total influence (thus, the other 9 indicators share the rest of the overall influence).

Category Name	Abbreviation	Number of Indicators	Weight (as % of total)
Politics (Political System)	P	8	50 %
Gender Equality (Educational and Economic)	G	13	10 %
Economy	E	9	10 %
Health	H	10	10 %
Knowledge	K	15	10 %
Environmental Sustainability	En	5	10 %

Table 1: Categories of indicators

Each dimension is computed according to these two steps. For example, K6 ("Daily newspapers - per 1,000 people") is in a category with 15 dimensions;

thus, its influence is one fifteenth within this category. The "Health" category's overall influence is 10 %, resulting in K6's weight of 0.67 %.

The weights play an important role in the distance metric that reflects the proximity of two countries. The most commonly used metric is the Euclidian Distance

$$d_E(u,v) = \|u - v\| = \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}\ , \tag{1}$$

with *u*, *v* vectors of dimension *n*. The rest of this thesis will assume a modified version of d_E which takes the individual importance of the variables into account:

$$d_{E,w}(u,v) = \sqrt{\sum_{i=1}^{n} w_i (u_i - v_i)^2}\ , \tag{2}$$

where w_i is the weight of variable i.

2.3. Benchmark Data Set

The Iris data set is a popular multivariate data set which was introduced by R. A. Fisher as an example for discriminant analysis [Fis36]. It is much simpler than the Democracy data set and will be used to introduce the complex concepts in the later chapters, before these concepts are applied to the Democracy data. The Iris data reports on four characteristics of the iris flower, "sepal length", "sepal width", "petal length", and "petal width"; these characteristics are the variables of this data set, thus it is 4-dimensional; in contrast to the Democracy data set, which has a dimension of 60, it is much more convenient to demonstrate the SOM visualization techniques with the Iris data first, before they are applied to the Democracy SOM.

The data set contains 50 samples for each of the three species ("Setosa", "Virginica" and "Versicolor"), with a total of 150 samples. Setosa iris flowers are clearly different from the other two species, while Virginica and Versicolor are harder to distinguish. Furthermore, the values for the petal variables (width and length) are highly correlated.

3. The Self-Organizing Map

3.1. Vector Quantization

The task of finding a suitable subset that describe and represent a larger set of data vectors is called vector quantization (VQ) [Gra84]. In other words, VQ aims at reducing the number of sample vectors or at substituting them with representative centroids. Figure 3 shows the principle of VQ methods, reducing the original set of 8 samples to 5 samples.

The resulting centroids do not necessarily have to be from the set of samples but can also be an approximation of the vectors assigned to them, for example their average. VQ is closely related to clustering, which is a very important, far-reaching topic, so there is a whole chapter (5) dedicated to it, and one of the most important vector quantization techniques (k-means) will be discussed there.

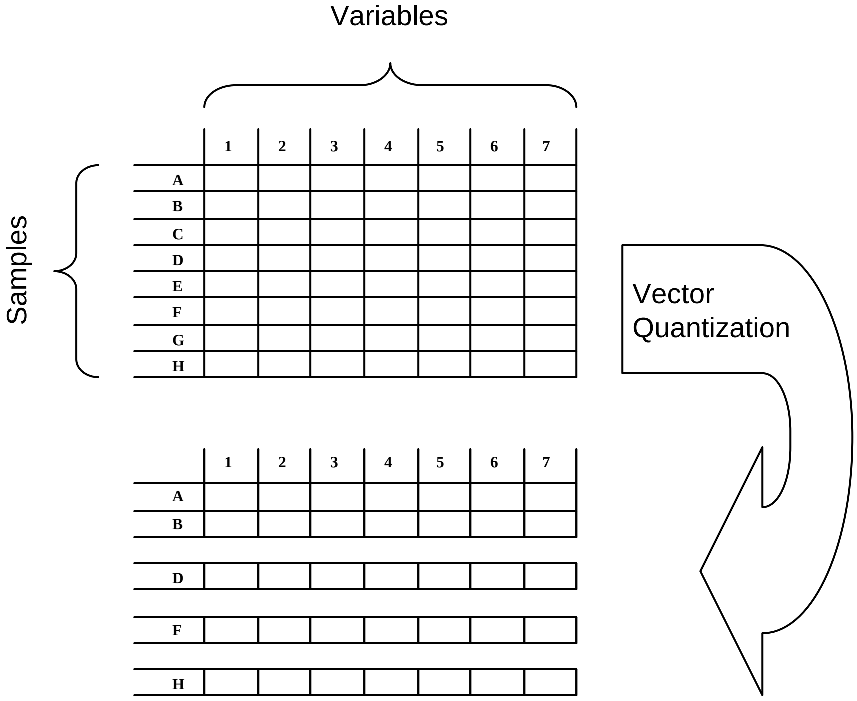

Figure 3: Schematic overview of vector quantization

Obviously, the SOM performs VQ since the sample vectors are mapped to a (smaller) number of prototype vectors, which will be explained in Sections 3.3 to 3.6. Due to its other capabilities like vector projection, the SOM has the following disadvantages from the viewpoint of VQ which distort the distribution among the prototype vectors as side effects from the neighborhood relation:

- Border effect: Neighborhood is not defined equally on the map, the units on the edges and in the corners do not have the same number of neighbors, but usually cover a much larger Voronoi region (in input space) than the ones in the center of the map. This leads to the phenomenon that the border units are selected more often as BMU, resulting in a concentration of the samples in these areas (this can be visualized by hit histograms, see Section 4.4).
- Interpolating units: If the data cloud is widely separated, the neurons on the map between regions that are highly different (in input space) are therefore updated through the neighborhood kernel of these units. It this case it is possible that some neurons are not targeted by any samples at all and do not represent any data vectors.

Apart from the SOM and k-means, another notable example for VQ is neural gas [Mar93].

To measure the quality of a VQ algorithm (and of course of the SOM), the quantization error of a prototype vector *m*

$$e_q(m) = \sum_{x \in C_m} \left\| x - m \right\|, \tag{3}$$

is introduced. It is found by calculating the difference between the sample vectors and their corresponding cluster centroids; C_m denotes the set of samples that are mapped onto prototype vector m (this concept will be discussed in Section 3.4). Thus, the quantization error indicates how accurate the data is represented by the codebook vectors. If the SOM is initialized in a linear way, the quantization error usually declines during training (as opposed to topographic error, which increases, see next section). The quantization error is also important as a validity measure for partitionings found by clustering algorithms and for an extension to the SOM that will be described in Section 5.5.

3.2. Vector Projection

Visualization is very important for data mining, and directly plotting a set of data can provide insights into its structure and underlying distribution that inspection of the numerical data table can not. However, data sets cannot be visualized on a sheet of paper or on a monitor if the dimensionality is higher than 2. There are ways to provide information by the use of colors or different shapes and sizes of the objects to be plotted; for the Iris data set, it would be theoretically possible to include the third and fourth dimension like this, but it would be hard to understand and imagine such a plot. For the 60 dimensions of the Democracy data set, plotting these simultaneously on a 2-dimensional space is outright impossible.

Vector projection (VP) aims at reducing the input space dimensionality to a lower number of dimensions in the output space, and mapping vectors in input space to this lower dimensional space; the "lower dimensional space" is usually 2-dimensional for visualization on a monitor or for printing to paper. Figure 4 shows the principle of VP, reducing a data set from dimension 7 to dimension 4. However, when compared to Figure3, where the resulting set of samples can be obtained by simply discarding obsolete ones, this can not be done with variables in vector projection; the resulting variables are usually obtained by complex algorithms.

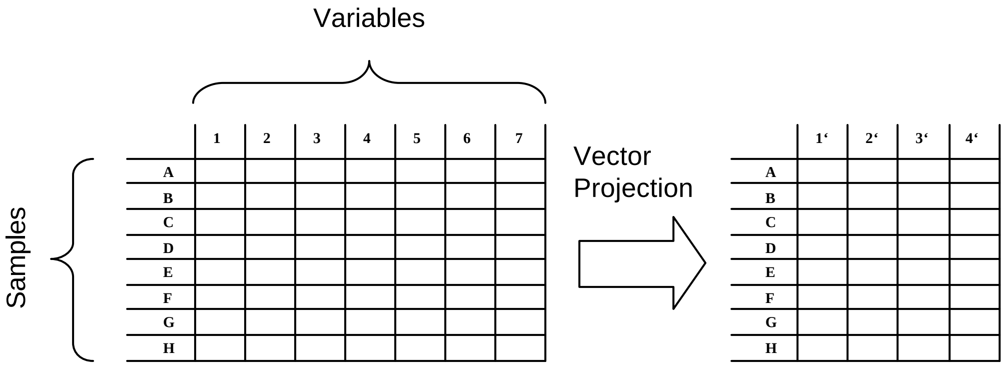

Figure 4: Schematic overview of vector projection

However, vector projection inevitably leads to loss of information in almost all cases. The VP mapping should occur in a way that the distances in input space are preserved as good as possible, such that similar vectors in input space are mapped to positions close to each other in output space, and vectors that are distant in input space are mapped to different coordinates in output space. Most

VP algorithms emphasize on the preservation of distances of vectors that are close to each other, while not necessarily preserving relatively large distances (as long as two samples that are far apart in input space are not placed next to each other in output space, it does not really matter just how far apart they are in output space). More generally speaking, the term "topology preservation" refers to this emphasis on local distances. Furthermore, VP methods can be categorized as performing either linear or non-linear mapping (NLM). Linear mappings are generally geometric projections to a plane (in the 2-dimensional case). NLMs try to uncover more complex structures in the data set. Usually, NML algorithms are less susceptible to outliers, but harder to compute and evaluate, since many of this category of algorithms include non-deterministic optimization techniques.

The SOM is of course also a VP method. It performs a non-linear projection by assigning the sample vectors to the units (BMUs) on the (usually) 2-dimensional grid. Other than the rest of the methods discussed in this chapter, the SOM is not a VP algorithm that maps to a continuous axis, but rather to a discrete number of map units. The SOMs Vector Projection qualities and limitations are discussed in more detail in [Fle97].

Some of the most prominent examples of VP are described here, with visualizations for the Iris and the Democracy data sets. Dimensionality reduction is performed from input dimension 4 to output dimension 2 for the Iris data, and 60 to 2 for the Democracy data. The plots for the Democracy data do not provide labels for all of the countries, since there is not enough room for this; only labels for 12 significantly different countries are shown: Argentina, Brazil, Hungary, India, Japan, Malaysia, Nepal, Norway, Russian Federation, South Africa, Turkey and the United States. Plots for the Iris data set show the iris flowers colored according to their category, where the blue squares stand for Setosa, green ("+") for Versicolor, and red ("o") for Virginica.

1. Metric Multidimensional Scaling (MDS):

MDS is widely used in psychology, the field it was originally developed for, by Torgerson [Tor52], his work extending that of Richardson [Ric38]. MDS tries to minimize pair wise distances of vectors, with error function

$$E_{MDS} = \sum_{i=1}^{N} \sum_{j=1}^{N} (d_{ij} - d'_{ij})^2 , \qquad \textbf{(4)}$$

where $d_{ij} = \left\| x_i - x_j \right\|$ in a suitable distance metric in input space, and $d'_{ij} = \left\| x'_i - x'_j \right\|$ the distance of the projected vectors in output space. This is achieved by moving the data points along the gradient of the error function.

2. Sammon's Mapping

Sammon's mapping [Sam69] was originally created as a non-linear alternative to Principle Component Analysis (see below). Of the VP algorithms presented here, Sammon's Mapping is computationally the most complex one. Given the error function

$$E_{Sammon} = \sum_{i=1}^{N} \sum_{j=1}^{N} (d_{ij} - d'_{ij})^2 / d_{ij} \, , \tag{5}$$

a solution can be computed iteratively with a gradient descent algorithm. The results for the Iris and Democracy data sets are shown in Figure 5.

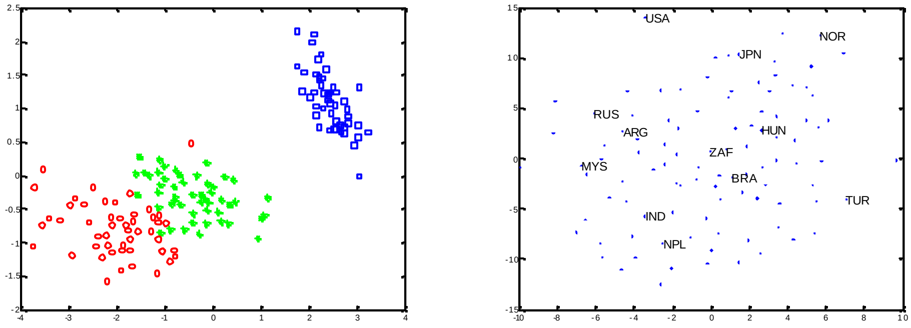

Figure 5a, b: Sammons Projection of Iris data (a) and Democracy data (b)

3. Principal Component Analysis (PCA):

PCA is a very prominent member of linear projection. It is based on eigenvalue decomposition and related to singular value decomposition (SVD). It algorithm rotates the input data such that the maximum possible variability is projected to the axes. PCA is computationally very fast since eigenvectors can be calculated very efficiently (it only requires solving a system of linear equations). The decomposition occurs in a way that the projections are ordered by their relative importance, with an output dimension up to the original input dimension. This is done by ordering the eigenvalues of the input data set (in a decreasing manner), and using their corresponding eigenvectors as axes for the projection. The sum of the eigenvalues used expresses the amount of the total variance of the data set that is explained by the axes used so far. Thus, the output dimensions are also ordered by decreasing importance.

Figure 6 shows the results for both data sets using PCA. In case of the Democracy data set, the first axis (in this example, the x-axis) of the projection explains 43.8 % of the variance, the second (y-) axis only 11.04 % (thus, the first two axes explain more than 50 % of the variance in the data manifold). The first 18 axis explain more than 90 % of the variance, and the last 20 axes explain less than 1%; Figure 7 shows the decrease of variance explained, where the i-th value on the x-axis refers to the i-th most important axis in output dimension. The y-axis shows the relative importance (between 0 and 1) of the corresponding x-value. The values on the y-axis have a sum of 1 (100 %). It is also important to mention that the resulting axes of the projection (which can be up to the original dimension) do not have any human-readable meaning anymore. Particularly, the variables of the Democracy data set do of course have a meaning, like "P1" referring to the political rights. The projected values can be achieved through linear transformations from the original variables, and are thus composite axes. If the number of dimensions of input and output space are the same, it is even possible to reverse this transformation, this means to project a vector from output space back to input space. It is, however, not possible anymore to assign a meaning to the axes anymore, other than that they represent a certain variance of the original data cloud.

PCA is especially important to this thesis in two ways:

- It will be used to project the codebook vectors of the SOM, which is a way of interpreting the map once it has been calculated
- For linear initialization of the map, PCA is applied to the training data, and the model vectors are initialized along the first and the second axis (which are the eigenvectors with the two largest eigenvalues). This is applied to the training of the Democracy Map, thus its initial state looks very similar to Figure 6b.

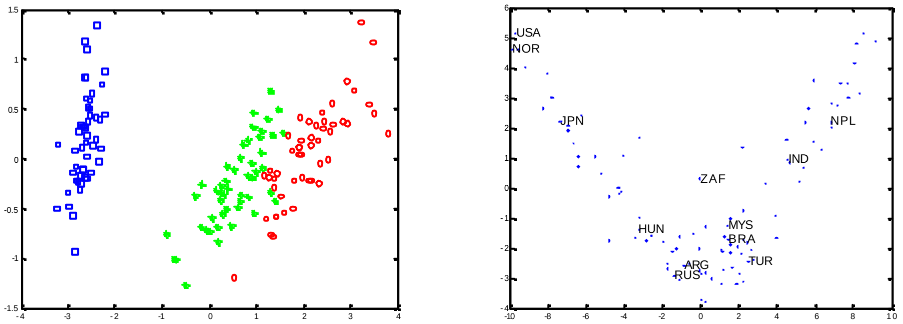

Figure 6a, b: Principal Component Analysis of Iris data (a) and Democracy data (b)

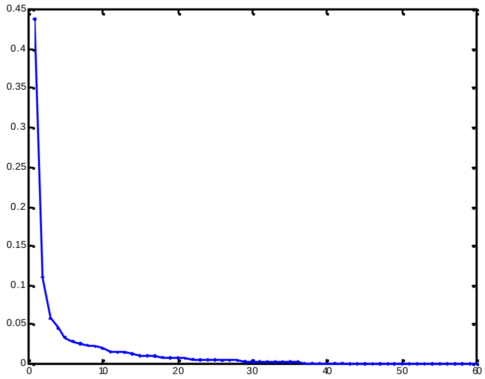

Figure 7: Decrease of variance explained in planes (PCA)

3.3. Introduction to the SOM

The Self-Organizing Map (abbreviated "SOM", also "Self-Organizing Feature Map" or "Kohonen Map") is a very popular artificial neural network (ANN) algorithm based on unsupervised learning. The SOM has proven to be a valuable tool in data mining and the larger field of Knowledge Discovery in Databases (KDD). It has been originally developed by Teuvo Kohonen [Koh01] and is mostly used for the visualization of nonlinear relations of multidimensional data. It has been subject to extensive research and has applications ranging from full text and financial data analysis, pattern recognition, image analysis, process monitoring and control to fault diagnosis; for a comprehensive list of references, see [Oja03, Kas98]. The original SOM algorithm has been extended in numerous ways [Fri94, Koi94], one of which will be discussed in Section 5.5. The SOM training algorithm is very robust; although there are some choices to be made regarding training length, map size and other parameters, these do not influence the results too heavily. Once a SOM has been trained, its results have to be post-processed. A large variety of post-processing methods exists, most notably the visualization methods (will be described in Chapter 4). The trained SOM can also be used for local modeling, segmentation (or "clustering", which will be discussed in Chapter 5), novelty detection, or classification of samples that are not part of the training set. Many of these features will be described and applied in this thesis.

3.4. SOM Training

The Self-Organizing Map learning algorithm is computationally extremely light, and with the Batch SOM algorithm [Koh01], which is described at a later point in this section, a substantial boost in performance has been achieved.

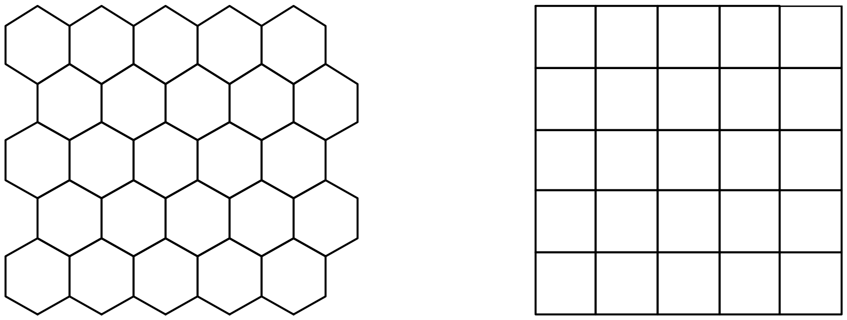

Figure 8a, b: Hexagonal and rectangular lattices

The SOM consists of a low-dimensional grid (or lattice) that contains a number *M* of neurons. In this thesis, only the 2-dimensional grid will be considered, since grids of higher dimensions are hard to visualize. The neurons are usually arranged either in a hexagonal or in a rectangular way (Figures 8a, b), other topologies exist but will not be discussed. The hexagonal lattice is the basis for most of the experiments with the Democracy and the Iris data sets, only in Section 5.5, which describes the Growing Hierarchical SOM, rectangular grids will be used, but its topology differs in many other ways from traditional lattices. The position of the neurons in the grid, especially the distances between them and the neighborhood relations, are very important for the learning algorithm. Each neuron has a so-called prototype vector (also "model" or "codebook" vector) associated to it, which is a vector of the same dimension as the input data set that approximates a subset of the training vectors (also "sample vectors" or "samples"). The dimension of the sample vectors (and the approximating model vectors) is called "input dimension", and is much larger than 2, the dimension of the grid (called "output dimension"). Thus the SOM is a so-called "Vector Projection" algorithm, because it reduces dimension (from the high dimensional input space to 2, the dimension of grid), a property that has been discussed in Section 3.2.

Once the codebook vectors are initialized with either random values or in another way (see Section 3.5), training begins. The training set of samples is presented to the SOM algorithm, and once all the samples have been selected, this process is repeated for t training steps. One complete round of training (when all of the samples have been selected once) is called an "epoch". The number of training steps t is an integer multiple of the number of epochs.

For training and visualization purposes, the sample vectors are assigned to the most similar prototype vector, or best-matching unit (BMU), formally

$$c(x) = \underset{i}{\operatorname{argmin}}\{\|x - m_i(t)\|\} \qquad \textbf{(7)}$$

where m_i are the prototype vectors, and x is the sample vector for which the BMU is determined. Sometimes instead of *c(x)* the longer form *BMU(x)* is written. The absolute value, as described in (1), is a suitable distance metric. For the Democracy SOM, however, a modified version of the Euclidian Distance metric is used that accounts for the relative importance of the variables, as described in(2). The learning process itself gradually adapts the model vectors to match the samples and to reflect their internal properties as faithfully as possible, which means that input vectors which are relatively close in input space should be mapped to units that are relatively close on the grid (output space). To achieve this, the training algorithm updates the model vectors iteratively during a number of training steps t, where a sample *x(t)* is selected randomly, and then the BMU and its neighbors are updated as follows:

$$m_k(t+1) = m_k(t) + a(t)h_{c(x)k}(t)[x(t) - m_k(t)] \qquad \textbf{(8)}$$

where $a(t)$ is the learning rate (which is decreasing monotonically with time) and $h_{ck}(t)$ is the neighborhood kernel. The neighborhood kernel determines the influence to the neighboring model vectors and its radius $s(t)$ is also decreasing with time. Thus, the learning process is gradually shifting from an initial rough learning phase with a big influence area and fast-changing prototype vectors to a fine-tuning phase with small neighborhood radius and prototype vectors that adapt slowly to the samples.

The above algorithm contains elements of two key concepts of learning, competitive and cooperative learning. Competitive learning is covered by selection of the BMU, the "winner", which is updated to the largest extent. Principles of cooperative learning are applied by not only updating the most similar model vector, but also its closest neighbors are moved to the direction of the sample to a lesser extent, creating similar areas on the map.

After training is finished, the SOM has folded onto the training data, where neighboring units usually have similar values. Each prototype is also associated with a Voronoi region in input space, which is defined as

$$V_k = \{x \mid \|x - m_k\| < \|x - m_j\| \, \forall j \neq k\}. \tag{9}$$

These regions reflect the area in input space for which a prototype is BMU. Input space is thus divided (or tessellated) into these non-overlapping Voronoi regions. If a unit's Voronoi region does not contain any sample vectors, it is called interpolating unit, which occurs if neighboring regions on the lattice contain distant prototypes in output space.

The algorithm described above is referred to as "sequential training" or "basic SOM". Another important learning rule is called "Batch map", which is based on fixed point iteration, and is significantly faster in terms of computation time. At each step, the BMUs for all input samples are calculated at once, and the model vectors are updated as follows:

$$m_k(t+1) = \frac{\sum_{i=1}^{N} h_{c(x_i)k}(t)\, x_i}{\sum_{i=1}^{N} h_{c(x_i)k}(t)}, \tag{10}$$

with N the number of sample vectors. Another option for updating the prototype vectors is calculating the weighted average of the Voronoi set centroids $n_k = \frac{1}{N_k}\sum_{x_i \in V_k} x_i$, such that

$$m_k(t+1) = \frac{\sum_{i=1}^{M} N_i h_{ik} n_i}{\sum_{i=1}^{M} N_i h_{ik}}, \tag{11}$$

with N_k the number of samples in V_k, or in other words, the number of samples for which prototype k is BMU; M denotes the number of prototype vectors. The batch map algorithm, with this extension, allows a very efficient implementation of the SOM.

3.5. Initialization and Parameterization of the Self-Organizing Map

Apart from the training algorithm, there are some choices to be made which can be seen as parameterizations of the SOM, namely choosing the functions $a(t)$ and $h_{ck}(t)$, the lattice topology, and the number of prototype vectors (and their initial state).

The initialization of the prototype vectors is usually one of the following:

- Random initialization: The model vectors are initialized randomly, which is not the best policy, but has been shown to converge to a topographic very similar map in the long run.
- Linear initialization: The prototype vectors are initialized according ascending or descending along the x- and y-axis of the lattice; the way this is done usually depends on the principal components of the data samples (this topic will be discussed in Section 3.5). This is the method that will be used in the rest of this thesis.
- Random Permutation of a subset of the samples: Similar to random initialization, random samples are picked as model vectors.

The linear initialization also has the advantage of being deterministic, thus reducing the randomness of the SOM training algorithm. This makes the results easier to reproduce.

The neighborhood kernel $h_{ck}(t)$ can be any function that decreases with increasing distance on the lattice $\|r_c - r_k\|$, with the components of the 2-dimensional vectors r its positions on the map. A typical example of a neighborhood kernel is derived from the Gaussian bell-shaped curve:

$$h_{ck}(t) = e^{-\frac{\|r_c - r_k\|^2}{2s^2(t)}}, \tag{12}$$

Figures 9 and 10 show four different neighborhood functions on a 30x30 lattice, with radius 6 (note that this is a very large map size and radius; it is only used for demonstration purposes). The center unit is selected as BMU and the influence on its neighbors is determined by the following neighborhood functions:

- Gaussian (Figure 9a): this is the neighborhood kernel as described in formula 8; since this function can never actually become 0, all the units on the map are influenced, even if this influence is very small for units that are far away from the BMU on the grid.
- Gaussian, cut off around radius (Figure 10a): same as above, but the influence region is abruptly cut off at radius $s(t)$; the advantage of this is that the map is not updated too frequently by minimal amounts.
- Bubble (Figure 10b): all of the units within the radius are updated by the same amount (the only values computed by this function are 0 and 1).
- Inverse proportional to the distance from the BMU (figure 9b), dividing the distance of the unit from the BMU by the square of the radius $s(t)$.

Figure 11 shows these functions as 2-dimensional plots, where the values on the x-axis denote the distance from the BMU, where the blue solid line represents the Gaussian model, the red dotted one the "Cut-off Gaussian" method, black dashdotted stands for Bubble, and green dashed for the inverse proportional method.

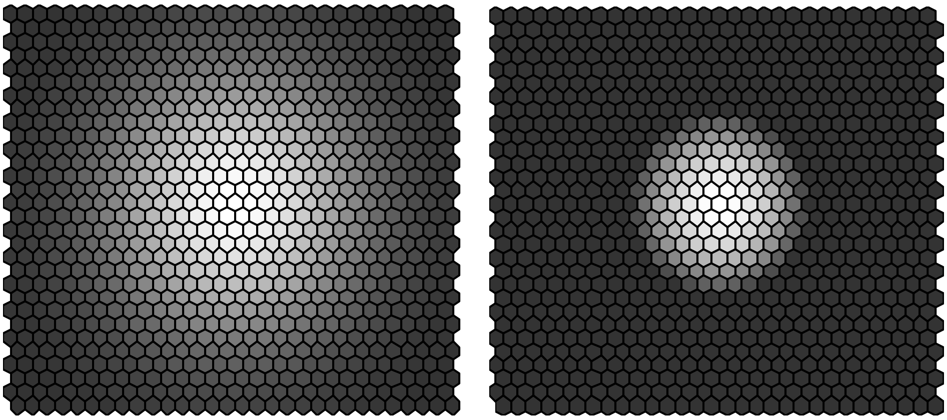

Figure 9a, b: Neighborhood functions: Gaussian and Inverse Proportional

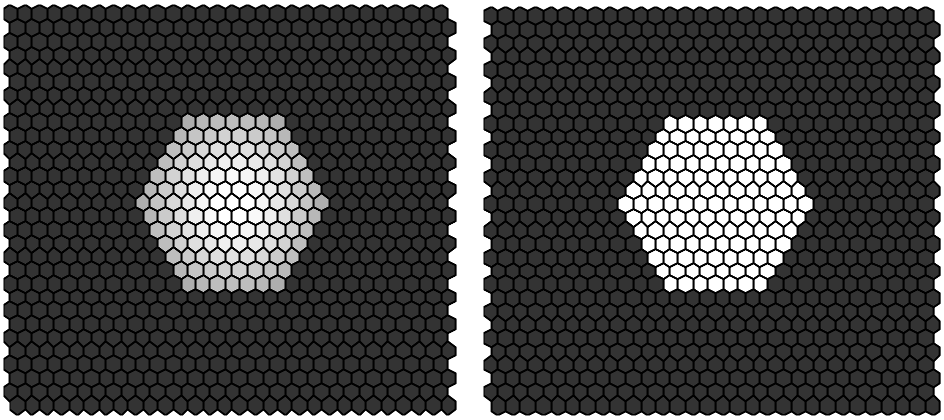

Figure 10a, b: Neighborhood functions: Cutoff Gaussian and Bubble

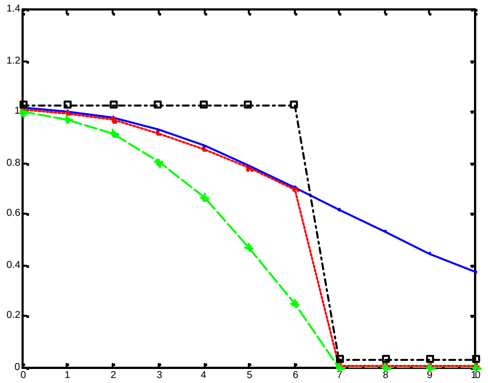

Figure 11: Comparison of neighborhood functions

The learning rate $a(t)$ is also decreasing monotonically with time, and should end up at zero when training is finished.

The choice of the map lattice (2-dimensional and hexagonal) will not be discussed; the number M of prototype vectors it contains, however, should be in the range of $\sqrt{N}$ (with N the number of samples). The map itself is usually rectangular, but not necessarily quadratic . Topologically, it would be better to use the same shape for the map as for the neurons, i.e. hexagonal, but these shapes are very inconvenient to display.

Surprisingly, the results do not vary significantly for different choices of any of the functions and parameters above, thus the SOM is a very robust algorithm with regards to its configuration.

3.6. Dealing with missing values in the data set

3.6.1. SOM training with data sets that contain missing values

As mentioned previously, the Democracy data set contains a number of missing values (MVs). The SOM is very robust with regards to MVs, but many of the pre- and post-processing steps cannot deal with them.

The SOM training algorithm has to consider MVs in two cases:

- Calculating the BMU of a sample with missing components
- Computing the updated model vectors of the BMU and its neighbors with a sample that holds MVs

Note that the model vectors must not contain MVs, and samples or dimensions must not consist of MVs only (that means, neither a row nor a column in the data matrix must consist solely of MVs). Apart from these obvious constraints, a sample or dimension should not contain too many missing values. The calculation of the BMU requires a modification of the distance metric, such that the missing variables are disregarded for the distance measure:

$$d_{E,w,MV}(m,x) := \sqrt{\sum_{i=1}^{n} w_i d(m_i, x_i)^2}\ , \tag{13}$$

and

$$d(m_i, x_i) := \begin{cases} m_i - x_i & \text{if } x_i \text{ is defined} \\ 0 & \text{if } x_i \text{ is missing} \end{cases} \tag{14}$$

This distance measure only applies to measuring distance between prototype m and sample x, where only the sample may contain MVs. This measure is not applicable to determine the distance between samples or between any two vectors which can potentially both contain MVs, but since the prototypes never contain MVs, the same missing component is simply ignored for all the candidates likewise. However, if $d_{E,w,MV}$ would be introduced as a distance measure between any two vectors, it would not qualify as a metric, since vectors with many MVs would be automatically closer than vectors that do not hold MVs.

Once the BMU has been determined and the training algorithm has to update the model vectors for the BMU and its neighbors, the prototypes' variables for which the sample's variable is missing are not updated, formally

$$m_{ki}(t+1) := \begin{cases} m_{ki}(t) + a(t)h_{ck}(t)[x_i - m_{ki}(t)] & \text{if } x_i \text{ is defined} \\ m_{ki}(t) & \text{if } x_i \text{ is missing} \end{cases}, \tag{15}$$

which is a slight modification of the usual learning rule, as it defines the updated vectors component wise (for each dimension *i*).

As shown above, the SOM can deal with MVs very effectively, but this does not hold for many clustering and visualization methods, and for methods that are compared to the SOM in any way (for example the vector projection methods discussed in 3.5.). As a preprocessing step, the gaps in the original data set have to be filled; in the following two sections, two methods will be introduced to interpolate MVs, one that is directly related to the SOM, and one that requires a priori knowledge of the structure of the data as intended by the Pilot Ranking. Of these two possibilities the SOM-related approach is taken and the modified data set is used throughout the rest of the thesis.

3.6.2. Interpolation with BMUs

Once the SOM has been trained with the data set that contains missing values as described in the previous section, the gaps in the data matrix can be filled in a very intuitive way. Since the model vectors are by definition approximations of the sample vectors, the BMU for each sample to be interpolated can be found, and the missing components are simply copied from the model vector, formally

$$y_i := \begin{cases} x_i & \text{if } x_i \text{ is defined} \\ m_i(x) & \text{if } x_i \text{ is missing} \end{cases}, \tag{16}$$

where y_i denotes the i-th component of (interpolated) vector y, and $m_i(x)$ is the i-th component of the BMU of sample x. The resulting sample vectors do not contain MVs any more. This approach is recommendable only if there are not too many MVs, and if the MVs are distributed uniformly over the data set, such that similar samples which are mapped to the same BMU do not all lack the same component, otherwise the training algorithm cannot update the component in question in a meaningful way. For the Democracy data set, this is problematic, because countries for which many values are missing are usually rather the less-developed ones, which are mapped to similar regions of the SOM, and the differences to the approach that was taken by the Democracy

Ranking are significant for several countries. Since this method requires decisions outside the scope of this thesis it will not be considered.

3.6.3. The Democracy Ranking approach to fill missing values

The approach that is taken by the Democracy Ranking exploits the obvious (and intended) similarity between dimensions within the same category. The missing values are approximated by determining the overall performance within their categories. For example, if P6 and P8 are missing, the weighted average of the rest of the variables within the politics dimension are calculated and the MVs filled with the resulting mean. This makes sense since the variables have been selected because of their high correlation and to provide a stable means of rating the quality of a specific aspect of a democracy. There are some exceptions, however, that are dealt with separately. For Bosnia and Herzegovina, for example, only one variable (of 9) is provided in the economy category; this variable is E7 - "Labor force, children 10-14 (% of age group)" with value of 0, which is the best this variable can achieve; this value is of course misleading and should not replace the missing values in the economy dimensions, because this would result in ranking Bosnia and Herzegovina higher than any other country in the economy category. Thus, the missing values are filled with averages from 2 other categories. It is obvious that assumptions of this kind cannot be made for the (rather technical) SOM-based approach of this thesis.

It is nevertheless interesting to compare the differences of the two interpolated data sets; the countries that differ most significantly are Bosnia and Herzegovina, Croatia, Georgia, India, Macedonia, Mexico, Moldova, Mongolia, Peru, and Sri Lanka.

3.7. Parameters for the experiments

The data set this SOM is trained with ("Democracy data set") is acquired by the following steps:

1. the original Pilot Ranking data is taken and normalized to zero mean and unit variance
2. a 7x7 map is trained with this data set (which still contains missing values)
3. the missing values are filled as described in Section 3.6.2.

Note that the SOM computed as described above is then discarded; its only purpose is to fill the gaps in the data matrix.

The Democracy SOM consists of a hexagonal lattice with 7x7 map units. The codebook vectors are initialized in a linear way as described in Section 3.5, with PCA performed on the data set, and the units initialized along the two most important axes. For training, the batch algorithm is applied for 25 epochs (5 epochs of rough training, 20 epochs of fine-tuning). This second map does not differ very significantly from the first one, it is rather used because some of the algorithms discussed in this thesis, like VP methods, do not work with missing values, and thus the Democracy SOM is more convenient to compare the results.

So, unless otherwise noted, the experiments and visualizations described in the following chapters are either performed on this Democracy SOM or the interpolated Democracy data set which is free of any MVs.

The mask for measuring the distance has been described in Section 2.2, and exact weights of all indicators are given in Appendix A.

The Iris SOM has a rectangual lattice of 16x4 map units with hexagonal layout. It is also initialized in a linear way, the training length is 23 epochs. Other than the Democracy map, this map uses traditional Euclidian distance, without a mask, so all the components have the same weight.

3.8. Quality measures of the SOM

After training has finished, it is important to measure the quality of the resulting map. As described in Sections 3.1 and 3.2, the SOM's strengths lie especially in the fields of vector quantization and vector projection. In the following paragraphs, several functions will be presented that provide a measure for these properties.

Firstly, the vector quantization properties of a map will be investigated. In Section 3.1, the quantization error $e_q(m_i)$ has been introduced. The mean quantization error E_q is based on this concept, formally

$$E_q = \frac{1}{M} \sum_{i=1}^{M} e_q(m_i), \qquad \textbf{(17)}$$

where M is the number of model vectors of the map. This value measures the data representation accuracy. If this value is high, the codebook vectors do not fit the data manifold.

Another error function measures the quality of the map from a vector projection point of view, or in other words, the structure of the map is considered. It compares the topology of the input and the output space. A simple method investigates the location of all the samples on the map. The key idea behind this is that the best-matching unit (BMU) should be next to the second-best-matching unit, otherwise this is regarded as violation of topology and thus penalized by increasing the error value. Formally, this can be written as

$$E_t = \frac{1}{N} \sum_{k=1}^{N} u(x_k), \tag{18}$$

this is called the topographic error, with

$$u(x) = \begin{cases} 0 & \text{if the BMU of } x \text{ is next to the 2nd BMU of } x \\ 1 & \text{otherwise} \end{cases}. \tag{19}$$

For the topology preserving properties of the SOM, other measures and error functions exist, see for example [Vil97].

If the SOM is initialized in the linear way as described in Section 3.5, the topographic error will usually increase during training and the mean quantization error will decline. In Section 4.9 the training process will be visualized, and the quality measures E_q and E_t will be given and compared at several stages during training.

However, the previous two functions cannot be considered energy functions that have to be minimized by the SOM algorithm to find the optimal solution. The topographic and mean quantization errors have to be considered as a trade-off between two important properties of the SOM algorithm, namely VP and VQ. The SOM has been shown to be hard to describe mathematically. However, it is possible to define an energy function for the SOM if the neighborhood kernel does not change, which is called the map distortion measure

$$E_d = \sum_{i=1}^{N} \sum_{j=1}^{M} h_{b_i j} \left\| x_i - m_j \right\|^2 . \qquad \textbf{(20)}$$

This cost function has been shown to be minimized by the SOM in [Koh91a]. This function is subject to current research, see [Lam00] for an in-depth discussion.

To summarize, the numbers of each of the quality measures for both the Iris and the Democracy SOM are given in Table 2. The results indicate that the Democracy SOM performs especially good in terms of topology preservation, mostly due to the linear nature of the data set. Note that the two SOMs in this table can not be compared due to different map sizes and training set quantities.

	E_q	E_t	E_d
Iris SOM	0.3030	0.0667	1.8953
Democracy SOM	0.5066	0	2.5146

Table 2: Quality measures of the Iris SOM and the Democracy SOM

4. Visualization of the SOM

4.1. Overview and introduction to visualization

4.1.1. Importance of visualization

Once a SOM has finished the trained process, it is ready for post-processing and visualization. This step is particularly important, since it actually presents results for further data analysis. The intuitive and meaningful visualizations are actually one of the most important strengths of the SOM. While neural networks usually are very hard to visualize, the SOM is a notable exception; this is one of the reasons for the popularity of the SOM. Visualizations that stress clusters usually work either with differences between neighboring codebook vectors (U-Matrix) or perform a clustering of the map and visualize this by coloring similar regions of the map with the same color. Visualizations that perform a projection usually require a data set which is then projected to the map, most prominently hit histograms. Other visualizations aim at clarifying the data's internal structure, like correlation, degree of linear dependency, etc.

The rest of this chapter is organized as follows:

Sections 4.1.2 and 4.1.3 introduce several ways to present and plot values on a map lattice. Section 4.2 describes a visualization that is applied to one single dimension at a time. Section 4.3 describes distance matrices which express the similarity between neighboring map-units. Sections 4.4 to 4.8 describe visualization methods that take the distribution of a set of data samples mapped onto the SOM into account (which does not have to be identical to the training set). In Section 4.9, a method is presented that directly visualizes both the codebook and the sample vectors in input space with vector projection methods. Section 4.10 shows an attempt to combine multiple component planes in a single plot.

4.1.2. Plots that show a single value per map unit

Most of the visualization techniques rely on computing a single value for each of the map units in a specific way. To make the results reproducible, explicit formulas will be given in each section, similar to this:

$$value_i = \dots ,$$

where $value_i$ is the value of the i-th model vector. The index i is running across the hexagonal grid as shown in Figure 12, and satisfies $1 \leq i \leq M$, where M is the number of prototype vectors.

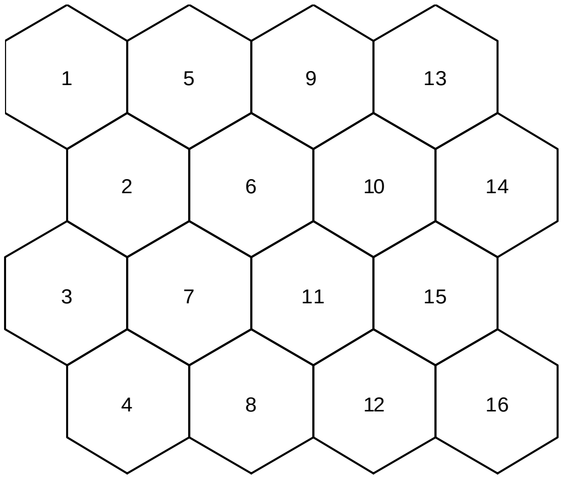

Figure 12: Indices of SOM units on a hexagonal lattice

In the rest of this chapter, the possibilities of generating figures with Matlab, the "SOM Toolbox", and the "SDH Toolbox" will be shortly described. If a single value has to be visualized for each map unit (as described above), there are several ways the plot can reflect this. Figure 13a shows (random) values that will be illustrated with these methods:

- **Color coding**: When all colors have been calculated, the patches are colored either according to Matlab "jet" colors (Figure 14a) such that the largest value is always dark red, the smallest dark blue, and cyan, green and yellow in between, or using grayscale (Figure 13b). The coloring scheme is always scaled linearly, so when two figures are compared, equal colors can possibly refer to different values, which can be misleading. Also, the color bar can be divided into a discrete number of classes. Figure 14b for example uses only 4 colors from the jet color bar.
- **Patch size**: Adjusts the patch size to reflect the value, large values result in large hexagons and vice versa. This is especially useful for visualization of distance matrices (to emphasize borders) and hit histograms, since it suggests that the large patches are "more important"

than smaller ones. This type of visualization can easily be combined with other methods.

- **Contour plots**: To visualize the values as an analogy to mountains and canyons, contour plots can be employed, which first use interpolated values for the space between the units, and plot the surfaces with different levels of detail (Figure 15b shows the contour plot with 4 levels). The peaks in this type of plot suggest that this area is very crowded, thus this visualization is best used for hit histograms and SDH (Sections 4.4 and 4.7).
- **Markers**: According to the value, the marker's size is scaled and painted atop the underlying map unit, which can be seen in Figure 16. The largest value corresponds to the marker which is as large as a whole hexagon. This visualization method is very similar to the patch sizes, since the value is depicted by adjusting the size of a geometric figure. To maximize the amount of information represented by the plot, it can be used in combination with other techniques like color coding. Like contour plots, this is useful for showing how crowded a region is.

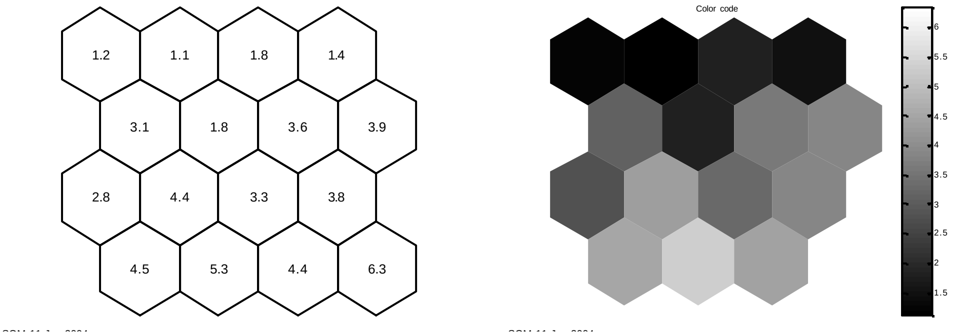

Figure 13a, b: Values shown as numbers (a); gray shading (b)

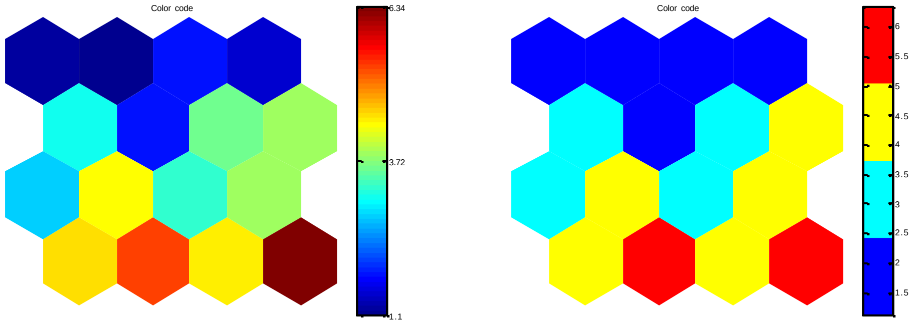

Figure 14a, b: MatLab "jet" color map (a); "jet" colormap with 4 levels (b)

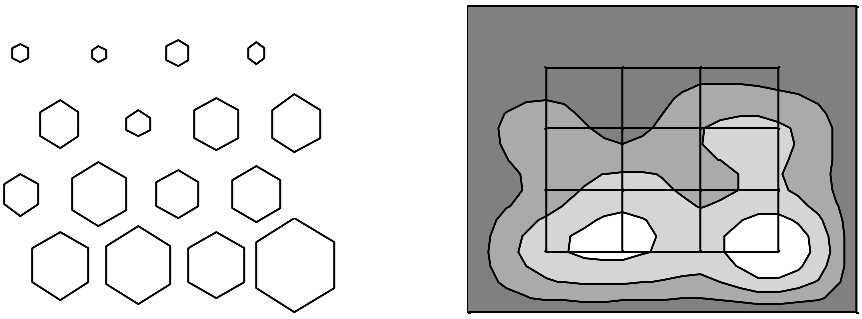

Figure 15a, b: Patch size according to value (a); contour plot with 4 levels (b)

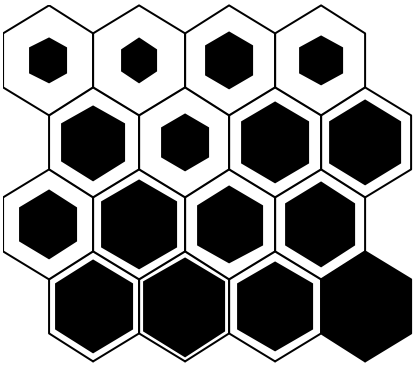

Figure 16: Marker size according to value on top of lattice

4.1.2. Plots that show multiple values per map unit

If a visualization technique requires plotting different variables, there are two methods to do this, depending on what kinds of variables have to be plotted: If the variables are coequal, if they are results of a computing algorithm that has more than one output value, like a vector, the plot should reflect this; if, on the other hand, the variables do not stem from the same source, the plot should preferably consist of a combination of multiple visualizations, for example patch size coding some value and color of the hexagon coding another one. The former case will be explained here in more detail. It can be used to visualize several components of a vector, which is suitable for a low number of dimensions. For the Democracy SOM, with a dimensionality of 60, this does not seem advisable, but later (in Chapter 6), several methods are proposed which are best visualized with these types of plots. In particular, the following styles are applicable (like above, Figure 17a shows the values for each vector, which will be depicted with the methods introduced here):

- **Bar charts**: Each component is represented by a bar, and each map unit holds a bar chart. The height of each bar corresponds to how large the corresponding value is. Negative values are depicted by bars that are below the horizontal axis. Also, bars on different units always have the same color, which makes it easy to identify single components, which makes this type of visualization advisable for low dimensional vectors (up to around 10). This plot is hard to combine with other visualization techniques.
- **Pie charts**: This type of visualization shows a single pie chart for each map unit. Other than bar charts, pie charts can not be applied to vectors that contain negative values. Pie charts show the relative part of each variable as fraction of the sum of all parts. Each component is depicted as a slice, the size of which reflecting the value to be displayed (see Figure 18a). However, the pie charts alone do not indicate how big the values are in comparison to other map units. This can be resolved by combining this visualization with adjusting the patch size according to the absolute size of the values, as shown in Figure 18b.
- **Projection into color space**: This requires a vector projection method like PCA. The high-dimensional vector is projected onto a color plane. Consider Figure 19, where the color value is obtained by projection of the 3 components to the 1st (most important) axis of the PCA decomposition. The problem with this is the obvious loss of information, and the fact that it is hard to imagine what the value obtained by this method actually means.

One type of visualization that is used in this thesis has not been explained yet, namely trajectories, because it is easier to describe in context of the algorithm it is used with in Section 4.8.

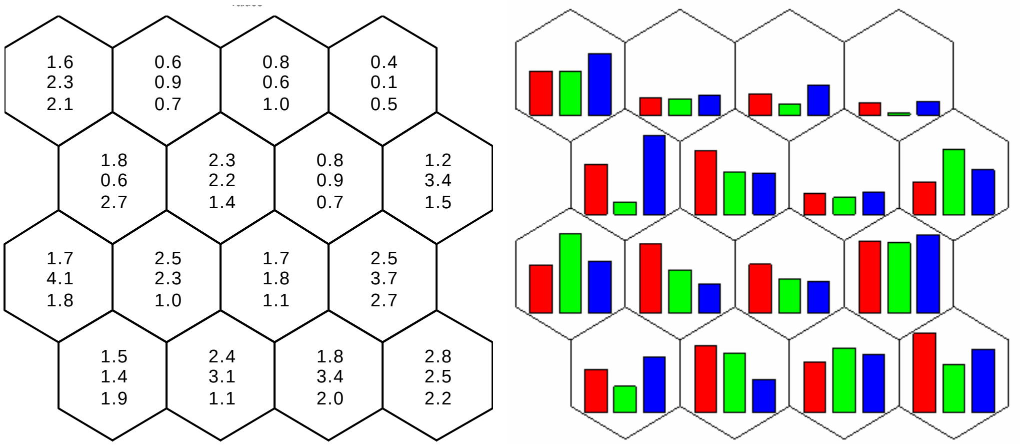

Figure 17a, b: Values as numbers (a); bar charts (b)

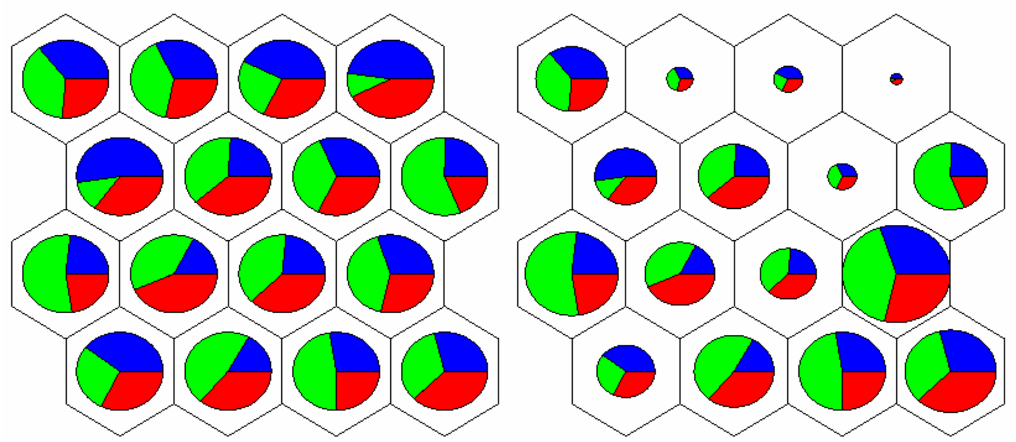

Figure 18a, b: Pie charts (a); pie charts with size according to sum of values (b)

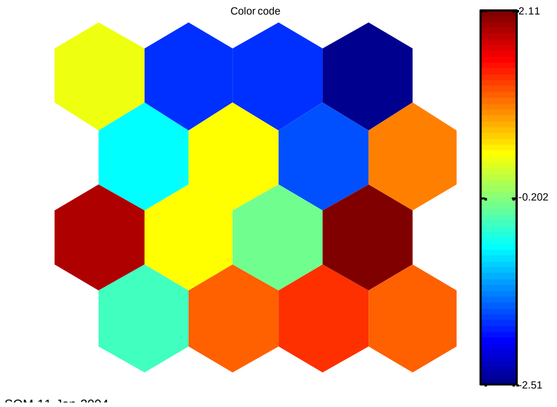

Figure 19: Projection to color plane (PCA)

4.2. Component planes

The codebook vectors usually cannot be visualized directly at once, since they are very high dimensional. This can be done with yet another vector projection method, i.e. application of another SOM to the codebook vectors of the first one, but this is not always convenient. It is, however, sometimes useful to look at individual variables and to identify regions on the map where they are influential to the distribution of the prototype vectors. These direct visualizations of variables are called "component plane visualizations". Pair wise comparison of component planes can reveal correlations, linear or non-linear dependencies between the dimensions in question. In case of very high dimensional data, the component planes become more difficult to evaluate because of the high number of plots. Several methods have been proposed to cope with this difficulty, i.e. arranging the component planes such that similar ones are displayed close together using a SOM-based projection [Ves00]. This can also be performed manually, but this is a very tedious and time-consuming task.

The colors are computed by simply selecting a single variable:

$$color_k = (m_k)_i \,, \tag{21}$$

where $(m_k)_i$ denotes the i-th component of codebook vector m_k, and i must be between 1 and N, the input dimension of the codebook and data vectors. In other words, component plane visualizations are projections to a single variable axis.

The component planes of the Iris data set are especially interesting, since there are only 4 of them. The Democracy data set consists of 60 dimensions, thus there are 60 component planes, which is not very concise anymore. Figure 20 shows the component planes for the Iris data. The upper right plot ("SepalW") is the most interesting one here, since very high (red) and very low (blue) values are close on the map: This leads to the assumption that there is some sort of border between these regions (which will be affirmed by the U-Matrix visualization in Section 4.3). Also, it seems that the remaining three variables are highly correlated, with minor deviations on the bottom of the plot.

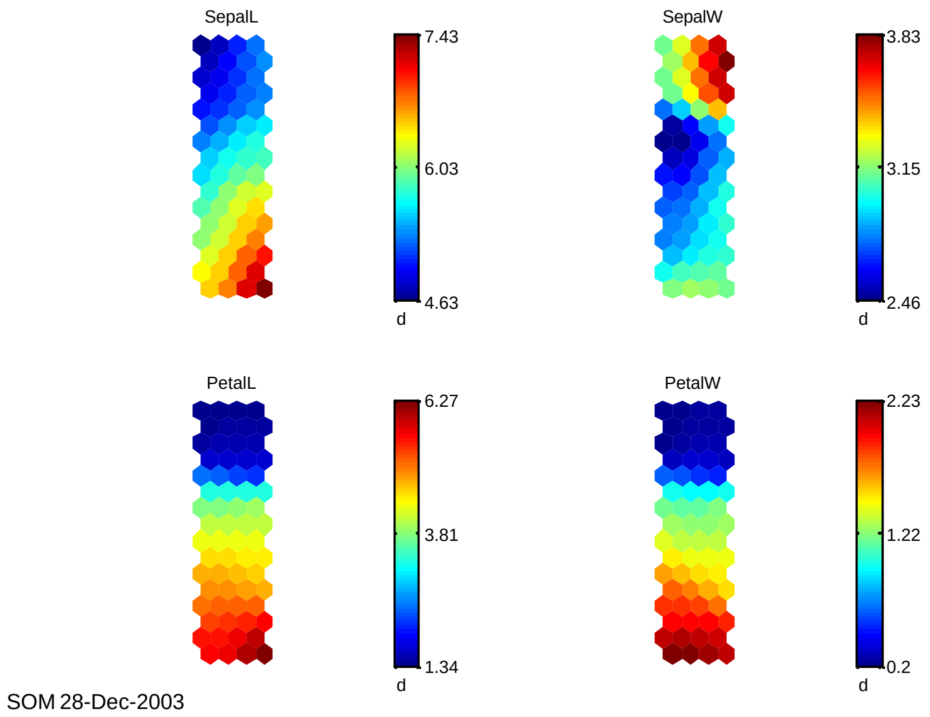

Figure 20: Component planes of the Iris Map

Figures 21 and 22 provide examples for component planes of the Democracy SOM. Figure 21a depicts dimensions *H2 – "Health expenditure, private (% of GDP)"* and *K13 – "Information and communication technology expenditure (% of GDP)"*. Figure 22a shows *P1 – "Political rights"* and *G1 – "Employees, agriculture, female/male (% of economically active female/male population)"*. Dimension P1 is the most important in terms of its weight, and is thus the most representative of the Democracy SOM as a whole, with high values on the upper right border of the map, and lower values along the low border. With K13, the high values are also located in the upper right area, but the low values are centered around the middle of the left area. With H2, the centers of the highest and lowest values are even very close together, both left of the middle. G1 again shows a distribution of high values in the upper right and the center of low values close to the peak values of H12 as described above. This leads to the assumption that most (not all) dimensions have high values in the upper right corner of the map, while the low values have different centers, but tend to be located on the lower part of the map.

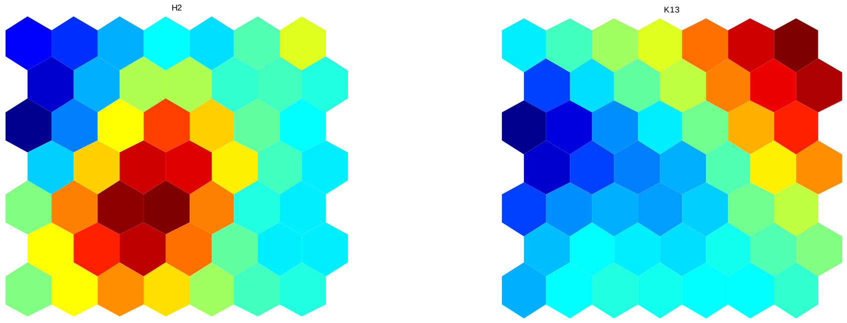

Figure 21a, b: Component planes H2 (a) and K13 (b)

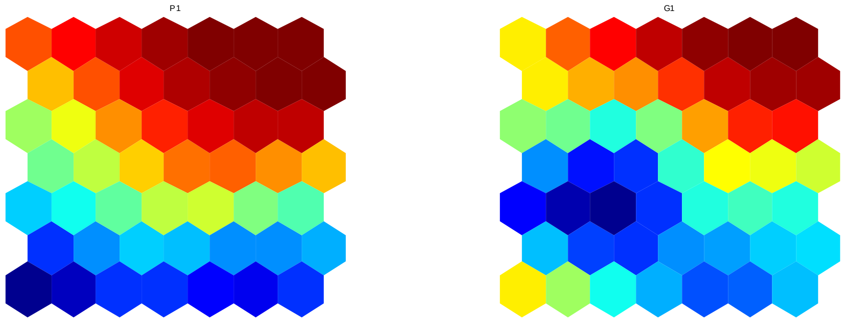

Figure 22a, b: Component planes P1 (a) and G1 (b)

4.3. Distance Matrices

Distance Matrices are among the most commonly used means of visualization of the SOM. Other than the component planes, they investigate the differences of adjacent prototype vectors. Thus, the resulting plot reveals the contiguous regions, areas with sharp borders to the rest of the map (this corresponds to the definition of clusters, see Section 5.4), and interpolating units, that are usually highly different from all of their neighbors. The two most prominent examples of distance matrices are the unified distance matrix, or U-Matrix [Ult90], and the D-Matrix.

The U-Matrix calculates pair wise distances of adjacent prototype vectors, according to the same distance metric the map was trained with (this is usually the Euclidian Distance, in case of the Democracy map, weighted Euclidian distance). For visualization purposes, the resulting values are displayed between the actual prototype vectors; the color of the prototype vectors themselves is usually an average of the surrounding units, so that there are no missing units on the plot. Thus, the U-Matrix map is actually bigger than the original one (approximately twice the size for both axes, thus about four times the number of map units).

The formula for U-Matrix computation is

$$color_{i,j} = \left\| m_i - m_j \right\|, \qquad \textbf{(22)}$$

with i, j being indices of adjacent map units, and this formula only determining color of patches between codebook units.

Figure 23 shows the U-Matrix of the Iris Map. As suspected in the previous section, there is a sharp border below the upper third of the map. The map is essentially split into two regions, the upper third is occupied by Setosa species, and the lower two thirds correspond to the Versicolor and Virginica species, between which no strict boundary can be determined.

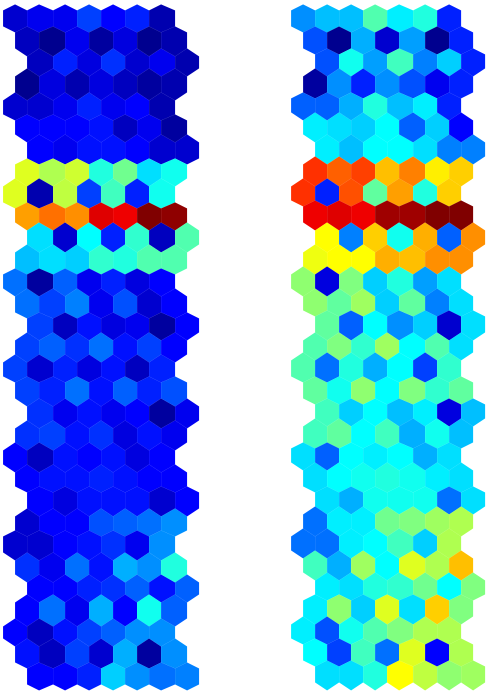

Figure 23a, b: U-Matrix for the Iris Map: normal colors (a) and log-scaled (b)

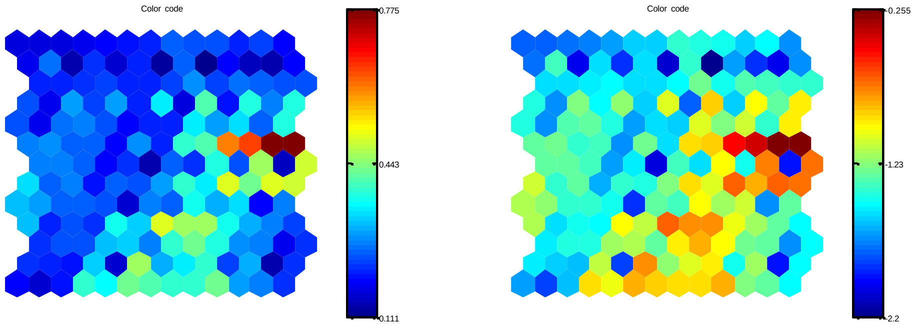

Figure 24a, b: U-Matrix for the Democracy Map: normal colors (a) and log-scaled (b)

For the Democracy data set, the U-Matrix shows the gap in the center right part of the map. This is due to the fact that the upper right corresponds to the highest rated countries, while the lower right corresponds mostly to developing countries; the transition between these regions can be clearly seen. Also, the bottom right part is rather incoherent, and the middle of the bottom part is separated from the left side.

There are actually two problems with visualizations of the U-Matrix: First, as mentioned above, the squares/hexagons between the units do have a color-value, but the units themselves (in the center) do not, so an interpolated value has to be used. Consider, for example, the dark blue hexagon in the center of the right border that is surrounded by yellow and red units; the yellow units on its left and right create the impression that the units in the horizontal line are highly different, which they are not, there is actually very little difference between them. The other problem has to do with the color scale. Since interpolating units yield high values, and the values between similar units are so low that they cannot be distinguished on the plot; as can be seen in Figures 23a and 24a, most hexagons are blue, only a few are red and yellow, where yellow should correspond to the middle of the scale. To be able to view borders between more homogenous clusters, either the color scale has to be transformed or the axis of the colors has to use logarithmic values. This is done in Figures 23b and 24b, with the effect that the plots seem brighter. In Figure 23b, one can see that the upper third of the map (which corresponds to Setosa) is more similar within its borders than the lower two thirds. In Figure 24b, that shows the Democracy SOM, it can be seen that the lower right area, which corresponds to the countries with the least score, is somewhat separated from the rest of the map, while the lower left region is highly coherent.

The D-Matrix is a derivate of the U-Matrix. For each map unit, the median of the previously computed distances to the neighbors is determined. Figure 25 shows the D-Matrix for the Democracy map (note that the values are rescaled, so the colors are different from Figure 24a). However, a better way to visualize the D-Matrix is by adjusting the size of the map hexagons, such that large patches correspond to small values. This way, it is very easy to recognize coherent regions from the plot.

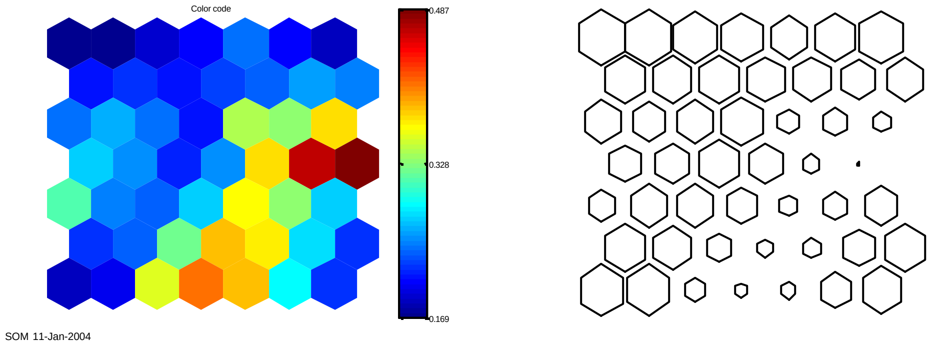

Figure 25a, b: D-Matrix visualization of the Democracy SOM: color coding (a) and patch size coding (b)

4.4. Hit Histograms

Hit histograms are a form of visualization that require an input data set which does not necessarily have to be identical to the training set. The data set is then visualized with the help of the SOM, namely by drawing the distribution of the data on top of the map lattice. Hit histograms are computed by finding the BMUs for the set of input vectors and counting how often each prototype vector has been selected. The units on the map lattice are then visualized such that they represent the number of times they have been selected. Most commonly, either a color scale is used to perform this visualization or a marker is placed on top of the map lattice plot, or sometimes this is done by changing the size of the unit in a way that frequently selected hexagons are large. Hit histograms are also a very reliable way of identifying interpolating units. Hit histogram computation can be written formally as

$$hits_i = \left|\left\{ BMU_i(x) \,\middle|\, x \in X \right\}\right|, \qquad \textbf{(23)}$$

where X is the set of data samples, and $|\bullet|$ denotes the cardinality of a set (i.e. the number of elements of this set).

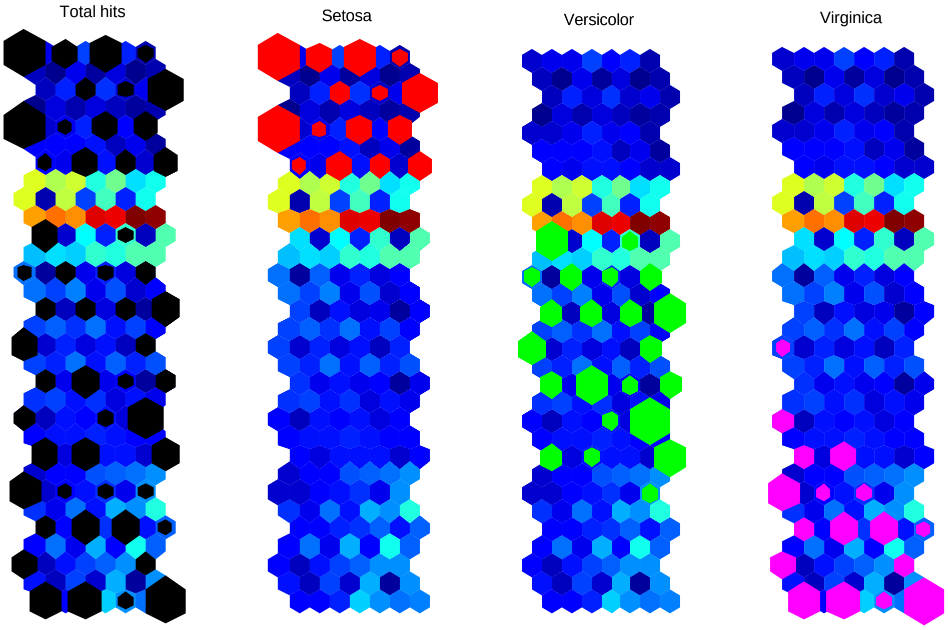

Figure 26a, b, c, d: Hit histograms: All samples (a), Setosa (b), Versicolor (c), Virginica (d)

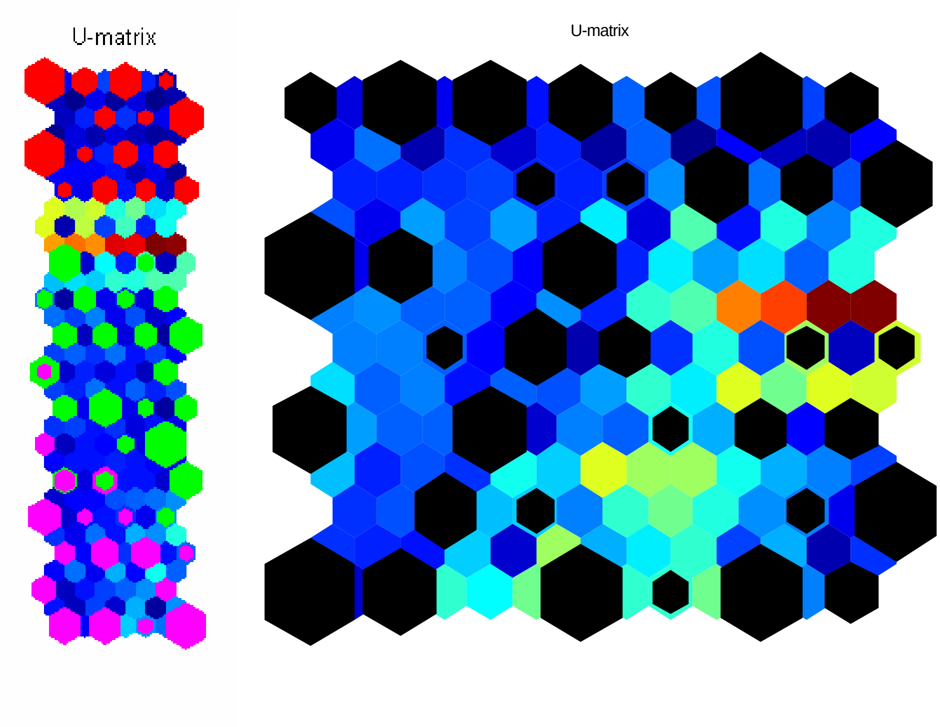

Figure 27a, b: Hit histograms: Iris Map with multiple hits (a), Democracy Map (b)

Figures 26 show the hit histograms for the Iris data set on top of a U-Matrix visualization. The first plot shows hits for the whole set of 150 samples without distinction of class. If only a representative subset of the samples is selected that represents a certain category of samples, the hit histogram becomes even more important. This is done with the remaining three plots which show histograms for the three species of iris flowers separately, and finally Figure 27a shows multiple hit histograms simultaneously in different colors. The hit histogram visualization for the Democracy map provides useful insight into the distribution of the countries as can be seen in Figure 27b. Note that obviously not only the regions around the center right are interpolating units, also numerous nodes around the center left of the map. Note also that in both cases, with the Iris and the Democracy data sets, the samples tend to be crowded near the borders of the map (this is called "Border Effect"), however this is less obvious in case of the Iris map due to the map's lengthy shape.

4.5. Labels

Another similar approach to show how well the map responds to the input data is labeling of the map units. How this technique is applied depends on the how many samples there are, or whether the samples are unique or occurrences of a class. Labeling is performed by assigning each sample's label to its BMU, and then displaying the map and showing the labels attached to the map units. If the number of samples is relatively low, each label can be plotted, but if there is more than approximately five labels per unit, the visualization will become very inconcise. In case of the Democracy data, each sample represents a unique class, since any two countries are always considered distinct. For the Iris data set, each sample is a randomly selected occurrence of one of the three classes of iris flower species. The labels for the Democracy data are thus i.e. "Albania", "Turkey" etc. while the samples of the Iris data set are labeled "Setosa" etc., where each label occurs more than once. It would be redundant to assign the same label to a unit repeatedly.

Figure 28 shows a labeled map of the Iris data set (the plot has been rotated by 90 degrees so the labels do not overlap). The approach taken here to prevent too many labels is called "voting", only the label with the most occurrences is shown. This leads to every model vector having at most one label (units that are not selected as BMU for any sample do not receive a label). Exactly the same visualization is shown in Figure 30 with a slightly different presentation: The labels are not printed verbally, but shown as color codes (green units are occupied mainly by Versicolor species, yellow refers to Setosa, blue to Virginica, and black fields are not selected as BMU at all).

Figure 29 show a labeled Democracy map on top of a U-Matrix visualization. Here, each label is assigned exactly once, but the map units can hold more than one label; in this case, voting is not applicable.

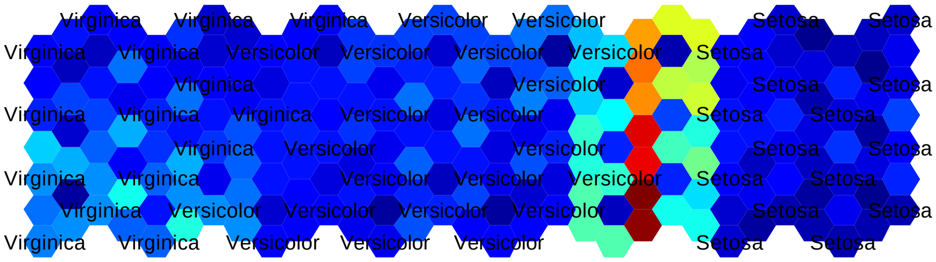

Figure 28: Labels acquired by voting of Iris Map

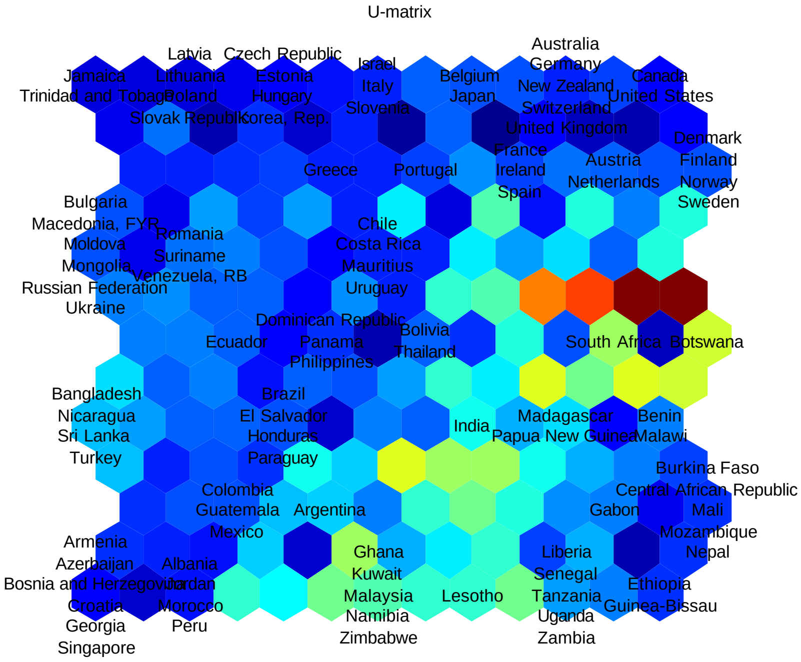

Figure 29: Labeled Democracy Map (on top of U-Matrix)

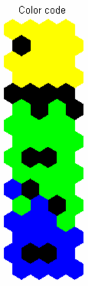

Figure 30: Colors according to most dominant labels of the Iris Map

4.6. The LabelSOM method

A different approach is taken by the LabelSOM method [Rau99]. It does not label the units with its samples, but with the variables that are most significant for them. For each map unit, the samples mapped to it are determined and for each of the components the deviation from the codebook value is determined, formally

$$q_{i_k} = \sum_{x_j \in C_i} \sqrt{\left(m_{i_k} - x_{j_k}\right)^2}, \tag{24}$$

where k is the dimension in question. Units that are not selected as BMU are not subject to labeling with this method. Also, since the map unit is supposed to be the mean of the samples it represents, the above formula is very similar to the standard deviation among these samples (if the standard deviation is used, then not only units with no samples mapped to it are not labeled but also units that only hold one sample, since computation of the standard deviation only makes sense for more than one value). Then, the dimensions can be ordered in an ascending way according to their q_{i_k}, which leads to the most similar variables among the samples to be ranked highest. These are the most characteristic properties of that unit. Usually, the number of labels is the same for all of the units, e.g. the 3 most important ones are then displayed. An example for LabelSOM will be given in Section 5.5 when visualizing the Growing Hierarchical SOM.

4.7. Smoothed Data Histograms

Another recently developed technique is the smoothed data histogram (SDH, see [Pam02]). As the name says, they are based on hit histograms, but the map units' counters are increased in a different way. For each sample, a ranking can be made for the map units ordered by the difference between the sample and the prototype vector, so the first entry is the best matching unit, the second one the second-best matching unit, to the worst-matching unit. The *s* best matching units are considered as hits and have their counters increased. The parameter *s*, the "spread", determines the length of the trace each sample leaves on the map. Since the *s* best-matching units usually lie on adjacent units in output space (unless the topology is violated), the resulting visualization does not suffer from the border effect as much as traditional hit histograms. Large *s* values leads to a blurring effect that results in only one big cluster with a peak value in the middle of the map, while the special case *s* = 1 is identical with the hit histogram. Additionally, a weighting scheme has to be defined that constitutes the decrease of influence the lower ranked units receive. Formally, this requires extra definitions for the membership degree, which employs a linear weighting scheme

$$membership_i(x) = \begin{cases} s / c_s & \text{if } \mathrm{m_i} \text{ is BMU for x} \\ (s-1)/c_s & \text{if } \mathrm{m_i} \text{ is } 2^{\text{nd}} \text{ BMU for x} \\ \vdots & \\ 1/c_s & \text{if } \mathrm{m_i} \text{ is s-th BMU for x} \\ 0 & \text{otherwise} \end{cases} \quad \textbf{(25)}$$

with c_S defined as

$$c_s = \sum_{i=0}^{s-1} s - i\,, \quad \textbf{(26)}$$

so total membership adds up to 1. The SDH is then acquired by adding all membership values for a specific map unit:

$$sdh_i = \left| \left\{ membership_i(x) \middle| x \in X \right\} \right| \quad \textbf{(27)}$$

Figure 31 shows a SDH contour plot for the Iris data set (note that it is slightly inaccurate since it assumes rectangular lattices). Again, it is obvious that there is a sharp border between the upper and the lower two thirds. The peaks are approximately in the same positions as they were in the hit histogram. Figure 32 shows the SDH as a contour plot for the Democracy data: Compared to the hit histogram, the peak regions are much more spread apart and form three clusters, one in each corner. Figures 33b shows the same SDH as a colored plot, and Figure 33a the normal hit histogram so they can be directly compared.

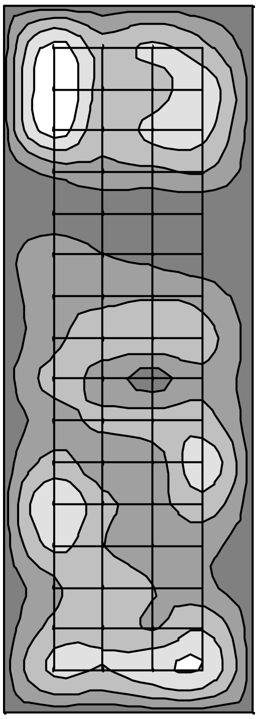

Figure 31: Contour plot of SDH applied to Iris Map with s=3

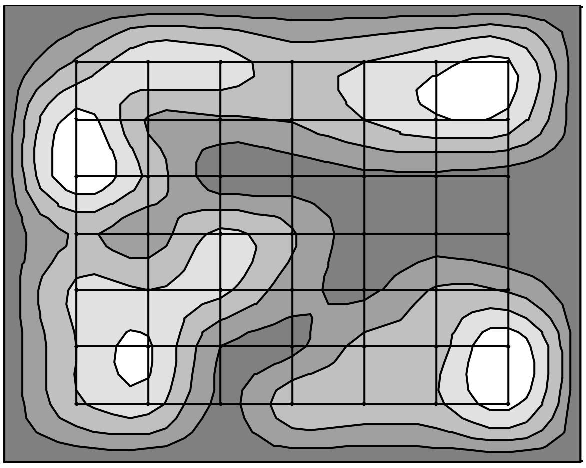

Figure 32: Contour plot of SDH applied to Democracy SOM with s=3

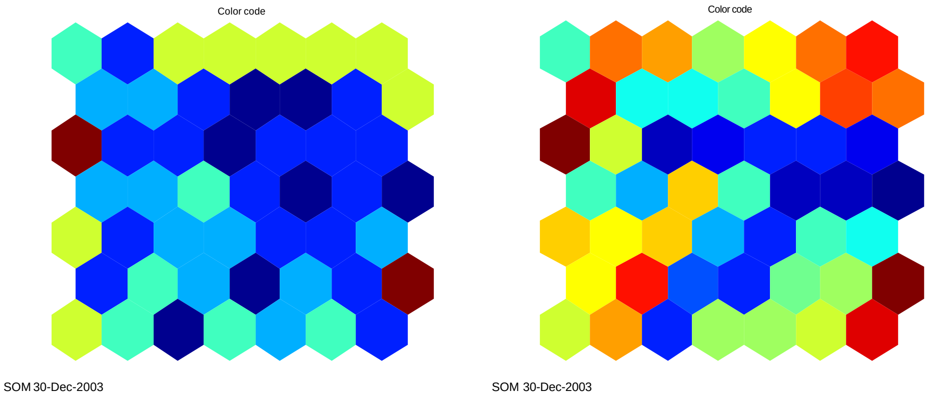

Figure 33a, b: Color coding for SDH of Democracy Map: Hit histogram (SDH with s=1) (a); SDH with s=3, same as figure above with contour plot (b)

4.8. Fuzzy response and hit trajectories of an individual sample

Sometimes it can be beneficial to visualize how one single, previously selected sample of interest is represented by the map. For example, it may be of interest what can be learnt from the Democracy SOM about Argentina. Several ways to do this have already been introduced in the previous sections, most based on BMU computations. In this section, however, two methods will be explained that show the Democracy SOM from a country's point of view. It will be shown how countries that share the same BMU differently respond to the neighboring units, which is related to the concept of quantization errors (as described in Section 3.1).

The first one visualizes the actual distances between a sample and each codebook vector in input space, formally

$$color_i(x) = \|m_i - x\|, \tag{28}$$

where x is the sample for which the distance is visualized.

The results can then be visualized for a plot of the map with color coded units representing the distance from the sample. Figure 34a shows the results for Argentina, where red units are relatively close to it, and blue ones are very distant. The unit with the lowest value (dark red) is this sample's BMU. This visualization method is called the fuzzy response of a sample.

Another method of visualizing the response of a samples is trajectories, which takes ranking of distances into account. This gives an answer to the question of how well a single data sample is represented by its BMU. Trajectory plots are usually applied to time-series related SOMs, but can also be used to clarify the ranking of a sample. Here, it will be used to show the sequence of distance of a sample to the map units. Figure 34b shows Argentina's BMU, 2nd BMU etc as a trace. If BMU and 2nd BMU are not neighbors on the map, the topology is violated (see Section 3.8). In case of Argentina, it is from a topological point of view very well represented by the SOM, because its 2nd to 7th BMU are all neighbors to its BMU.

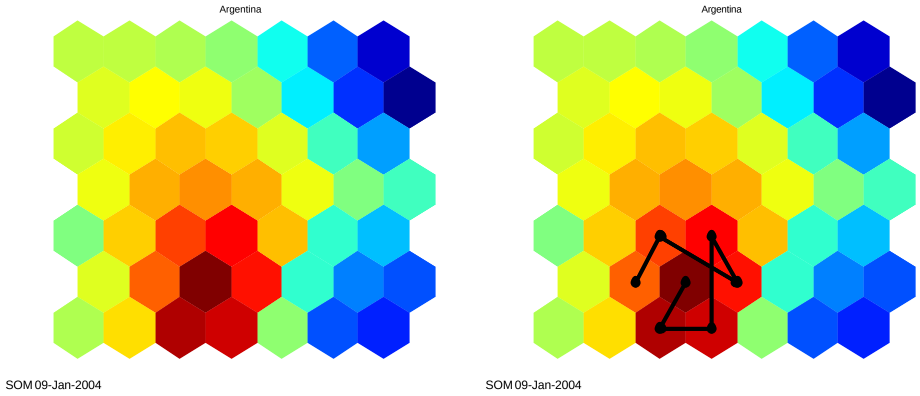

Figure 34a, b: Democracy Map's fuzzy response to sample "Argentina" (a); trajectory of BMU, 2nd BMU, up to 7th BMU (b)

Visualizations of this kind can be interesting for comparing different samples mapped to the same unit, in other words, samples that share the same BMU. Also, units on the edges or in the corners are specifically interesting since outliers are likely to be projected to these positions. Figures 35a and 35b show maps for two samples mapped to the lower left corner, Georgia and Singapore. Of these, Singapore is clearly an outlier, it has a large quantization error and is not represented well by its BMU (the coloring of the map has been set to grayscale so the differences between the two figures are clearly distinct). Georgia is also not too typical for this unit, since its trace can be followed across a series of non-adjacent units.

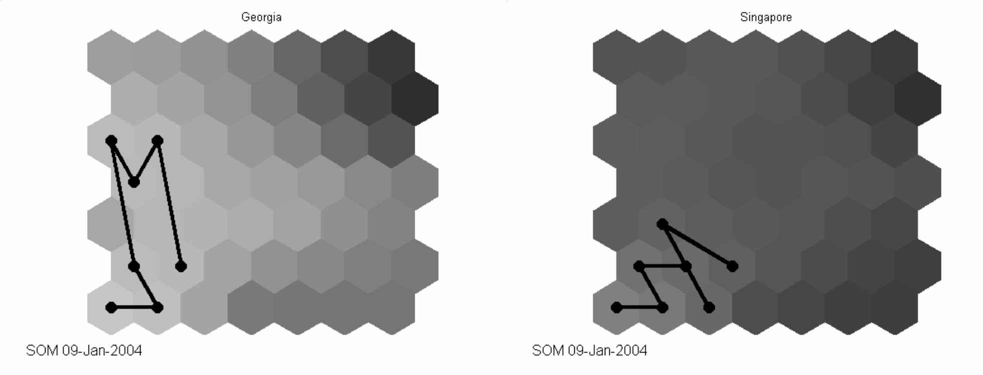

Figure 35a, b: Fuzzy response (non-scaled color code) and BMU trajectory: Georgia (a) and Singapore (b)

4.9. Projections of codebook vectors

This section describes the visualization of the map in (a projection of) input space. Principal Component Analysis (see Section 3.5) will be used here as projection method. This kind of visualization does not show the map lattice as in the previous sections, but a 2-dimensional plane with continuous axes. Both the data samples the map was trained with and the prototype vectors can be projected to this plane. Further, the codebook vectors are connected according to their neighborhood relations, thus the map is given its characteristic net-like look. Also, this visualization is useful to identify outliers and interpolating units. It will also explain which regions are more or less crowded than others.

This type of plot is especially useful for understanding the training process, if several of these projections are produced periodically during of SOM learning after certain steps. A disadvantage of this method, however, is, that the input space cannot be visualized directly, and thus one has to rely on another projection algorithm for reducing dimensionality.

Figure 36a shows sample and codebook vectors for the Iris data set. The black dots represent the model vectors, the lines between them show that these units are neighbors on the map. Blue dots denote Setosa samples, green means Versicolor, and red Virginica. Compared to the Iris Map visualizations that plot the map in its own topology, large distances for the connecting lines suggest that there is a "gap" in the map between the region that corresponds to Setosa and the rest of the data set.

In Figure 36b, the codebook of the Democracy map and the countries have been projected, and some countries have been highlighted to show the orientation of this plot. Again, it can be seen clearly that there is an interpolation region where the distances between the units are large. However, it has to be kept in mind that the PCA projection (as described in Section 3.5) only covers approximately 50% of the variance in the data manifold.

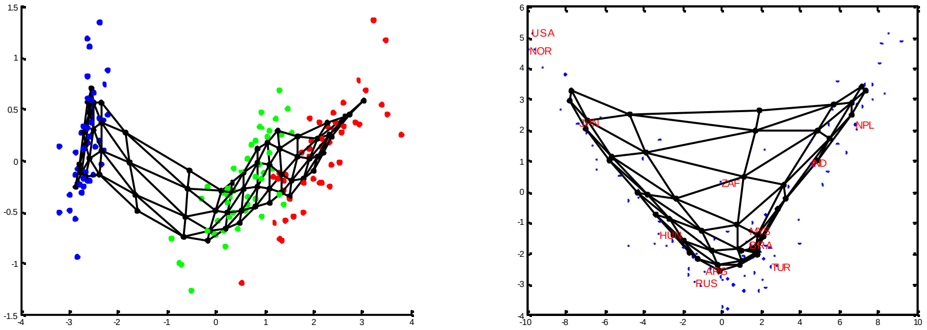

Figure 36a, b: Projection of data and SOM for Iris data (a) and for Democracy data (b)

An example of how the map adapts to the data manifold during training is shown in Figures 37 to 38. The initial state of the map is depicted in Figure 37a, after the prototype vectors have been set along the 2 greatest principal components (linear initialization, see Section 3.5). The diagonal connections between the units indicate that the lattice is hexagonal, since the nodes in the middle each have six neighbors. Then batch training starts, and Figure 37b shows the state of the codebook vectors after the first epoch. At the end of the 5th epoch, the rough training phase ends, and the training parameters for learning rate and the size of the neighborhood kernel are reduced considerably. Figure 38a shows the codebook projection before the fine-tuning phase starts. After another 20 epochs, training is finished, and the final map is shown in Figure 38b. When these figures are compared, it is obvious that most of the actual adaptation process happens in the first epoch of training. The difference between the map after rough and fine-tuning phases are not noticeable on these plots. The whole training process is summarized in table 3, where also the quality measures for topographic error and quantization error are given. In this example, the quantization error is steadily declining. The topographic error also declines after a short initial deterioration. The fact that it ends up at zero usually indicates that the data manifold is close to 2-dimensional.

The sequential training algorithm, which will not be discussed in detail, converges much slower than the batch version.

State	Quantization Error	Topographic Error	Figure
After initialization	0.6763	0.02	37a
After 1 epoch	0.5845	0.06	37b
After 5 epochs	0.5069	0	38a
After 25 epochs	0.5066	0	38b

Table 3: Training process of the Democracy SOM in detail

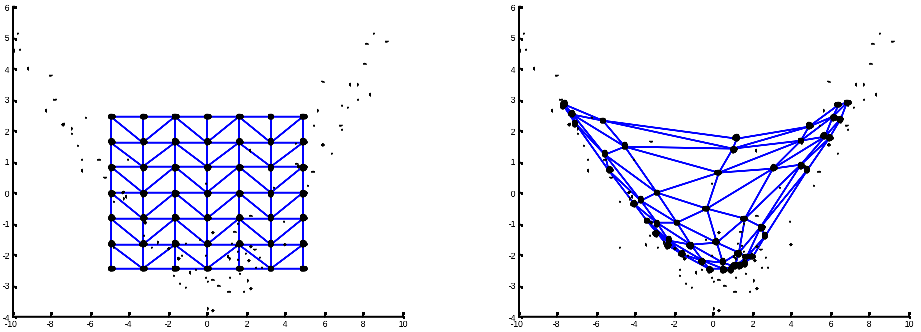

Figure 37a, b: Training of Democracy SOM: Initial state (a) and after 1st epoch (b)

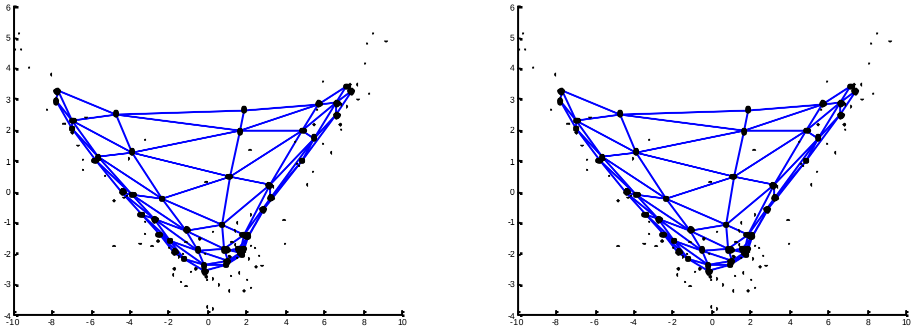

Figure 38a, b: Training of Democracy SOM: After rough training 5 epochs (a) and final state (b)

4.10. Attempts to combine several component planes in one plot

As described above, the codebook vectors cannot be displayed directly for high dimensions, so this is also not possible for the model vectors. The Visualizations discussed here try to show several components planes in output space simultaneously. Again, this is only relatively concise for low-dimensional input spaces. Also, if the components are highly correlated, plots of this category are easier to interpret.

The first method presented here is to show the model vectors' components with multiple bar-charts. The bars are arranged from the left to the right, and the size of the bar corresponds to the relative size of the value. Negative values are plotted below the horizontal axis. Also, each variable's bar is assigned its own color, so the components are easier to track across the plots. Each of the Figures 39 and 40 shows the component bars for Iris and Democracy maps. The Iris Map can be interpreted relatively easy. In case of the Democracy SOM, the individual components are impossible to distinguish. With 60 variables, this visualizations usually would not make any sense, but since the variables are very highly correlated, a fact that has already been exploited in large parts of this thesis, the bar-plots suggest that areas are depicted instead of independent components. Thus, regions can be clearly recognized, like the lower and upper right areas.

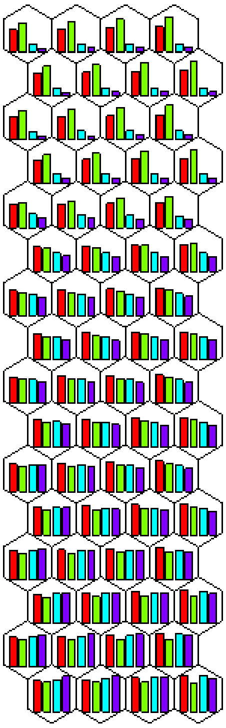

Figure 39: Components of Iris Map visualized as bar charts

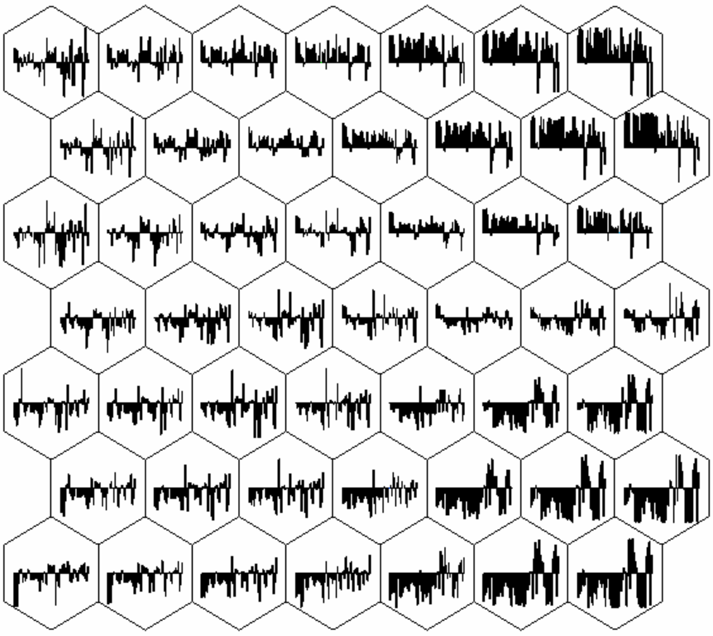

Figure 40: Components of Democracy Map visualized as bar charts

5. Clustering of the SOM

5.1. Introduction to clustering

Clustering is the task of dividing data points into blocks of similar objects. Since the whole data set is represented by a lower number of clusters, some of the details are lost, but the resulting subset is simpler and more significant regarding the characteristics of the data manifold. It is important to understand what actually makes a cluster a cluster: A "good" solution to clustering is usually defined as minimizing the distances between vectors mapped to the same cluster and maximizing distances between different clusters. Thus, once a partitioning has been found, it can be analyzed for the characteristics of the samples it holds, especially which features they have in common and which variables have a significant variance, and to identify the differences between distinct clusters.

Clustering is also a form of vector quantization, which aims at substituting the initial set of data sample by a smaller number of prototype vectors. The SOM actually performs vector quantization, since the sample vectors are mapped to the usually much fewer prototype vectors. In case of the Democracy data set, 100 countries are mapped onto a 7x7-SOM (= 49 prototype vectors), a reduction by approximately 50 %, but these are still too many units to be regarded as clusters. To achieve a better partitioning of the SOM, its prototype vectors are usually subjected to a clustering algorithm [Ves00], which ideally results in a partitioning into about 2 to 10 classes. Once this clustering has been computed, it can be visualized e.g. by plotting the map's grid and coloring the prototype vectors with a distinct color for each cluster. This provides an implicit quality measure of the SOM that the clusters are actually adjacent on the map.

The two most important categories of clustering algorithms are hierarchical and partitioning methods. Hierarchical clustering generates a cluster hierarchy that can be visualized as a tree-like graph ("dendrogram"). Usually, the algorithm starts with clusters that contain one sample each, in other words, every sample has its own cluster. Then, the two most similar clusters are merged at each iteration step, until there is only one cluster left which holds all the samples. Hierarchical clustering is discussed in Section 5.3. Partitioning algorithms usually have a fixed number of clusters to which the samples are assigned. Starting from an initial configuration, the samples are relocated between the clusters iteratively to gradually improve them. The most prominent example of partitioning clustering is the k-means algorithm, which will be described in section 5.2. Other clustering algorithms include grid-based methods and density-based algorithms. Grid-based methods divide the input space into regions, like of course the SOM, which generates a Voronoi Tessellation of the input space. Density-based methods try to identify highly crowded regions.

There are, however, many more clustering methods than the ones presented in this thesis, for a more in-depth discussion and comparison of the algorithms see [Ber02, Eck80, Jai99].

Clustering algorithms can also be categorized by the type of partitionings that are produced, either crisp or fuzzy clusters. This determines how the data samples are assigned to the clusters; in crisp clustering, each sample belongs to exactly one cluster, while in fuzzy clustering, the samples are members of several or all clusters to a varying degree. A prominent example for fuzzy clustering is fuzzy c-means. Only crisp clustering methods are discussed in this thesis.

Once a partitioning of the data samples (or in this case the SOM's prototype vectors) has been found, it is important to know if the clustering is plausible. Similar to the quality measures of the SOM (Section 3.8), a series of quality measures exist that evaluate the clustering. In section 5.4., the Davies-Bouldin Index will be introduced and applied to the previously found partitionings.

Section 5.5 introduces a recent extension to the SOM, the Growing Hierarchical Self-Organizing Map, which will be used to inspect the Democracy SOM at various levels of detail.

5.2. k-means clustering

K-means [Mac67] is probably the most popular clustering tool in scientific applications. It is an iterative partitioning clustering method that distributes the input vectors among k clusters. Each cluster is represented by the mean of the samples assigned to it, the centroid c_j. The error function the algorithm minimizes is

$$E(C) = \sum_{j=1}^{k} \sum_{x_i \in C_j} \left\| x_i - c_j \right\|^2 \quad \textbf{(29)}$$

where k is the number of clusters, x_i are the samples, and C_j are the actual clusters. After usually random initialization of the centroids, the samples are assigned to the clusters and the centroids are recomputed as the arithmetic mean of the samples. This step is repeated until the solution does not change anymore.

It has been shown that the k-means algorithm is actually a special case of the SOM with neighborhood radius of 0. Since the units cannot influence each other as opposed to the SOM algorithm, k-means is much more dependant on the initialization of the prototypes (centroids), and it is possible that one or more of the clusters remain empty.

Another disadvantage is that the number of clusters k has to be defined in advance. This can, however, be overcome by making use of an algorithm that automatically adapts to the best clustering size determined by a validity index, for an example see [Sip01]. One of these indices will be described in Section 5.4.

For the Iris data set, it is especially interesting to see if the k-means clustering algorithm yields the same partitioning results that the findings from the previous visualization methods suggested. The U-Matrix and SDH visualizations showed a gap that separates the upper third from the rest of the map, suggesting that for k = 2, the cluster should cover these regions separately. Figure 41a shows the iris map's partitions for 2 clusters, which matches this assumption. Another interesting choice is k = 3, to see whether the clusters also match the distribution of the iris species (compare Figure 27a on page 43). Figure 41b shows a k-means clustering of k = 3. When compared to the results for k = 2, note how the clusters overlap. This redistribution of vectors between clusters is characteristic for partitioning clustering algorithms, this could not happen with hierarchical clustering methods (see the next section). Again, the k-means clustering algorithm does not say anything about the plausibility of the choice of k or the quality of the partitioning.

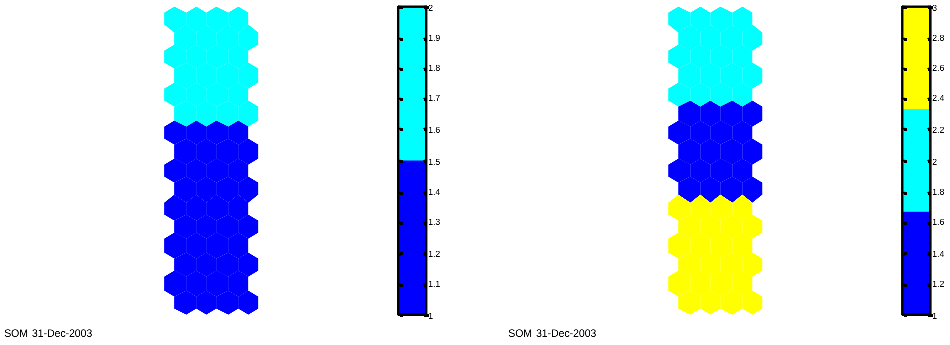

Figure 41a, b: k-means clustering of the Iris SOM: k = 2 (a), and k = 3 (b)

Figure 42 shows a possible k-means clustering of the Democracy map with 6 clusters. This choice for k is actually quite a good one, as the next sections will show, so the clusters will be described here (this description refers to "democratic score", which will be introduced at a later point, Section 6.4, which provides a quality measure of the sample and prototype vectors similar to the Pilot Ranking):

- Yellow cluster (top right): highest democratic quality countries, mostly industrialized countries (EU, Israel, North American and Australian countries, Japan)
- Dark red cluster (top left): second-best democratic score, Eastern European, South American and some Caribbean countries, Korea, Greece
- Dark blue cluster (bottom left): average ranked countries, South American and Asian (Middle Eastern) countries, Turkey, Bosnia and Herzegovina, Croatia, Morocco
- Light blue cluster (bottom right): relatively low ranked countries, mostly developing and poor countries from Asia and Africa
- Orange cluster (bottom center): lowest score, note that this cluster is in between two higher ranked ones (light blue, dark blue). This cluster holds 3 map units only, with developing countries from Africa, and Kuwait
- Cyan cluster (center right): holds mostly interpolating units, thus the democratic score is not very helpful; its lengthy shape is due to the border region between the yellow and the light blue cluster.

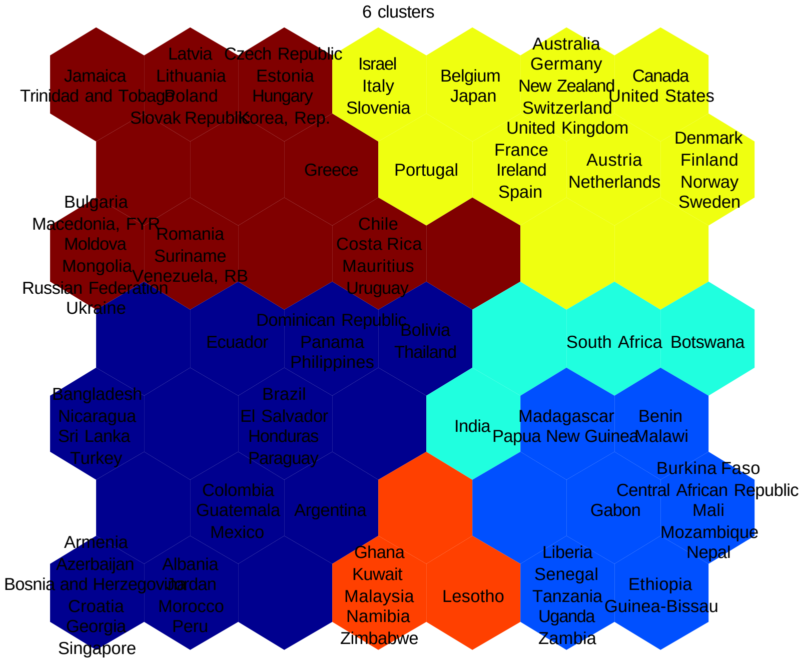

Figure 42: k-means of the Democracy SOM with k=6

5.3. Hierarchical Clustering

Hierarchical clustering differs from partitioning clustering in the way that the number of clusters does not have to be given in advance as a parameter, and elements can not be redistributed between clusters at later iterations. Hierarchical clustering algorithms are categorized into divisive and agglomerative methods, where only the latter will be discussed in this section, and an example for a divisive method will be given in the Section 5.5. Agglomerative methods build a hierarchy of partitions in a bottom-up manner starting with each element in its own cluster. At every iteration, the two closest (according to a specific distance measure) clusters are merged, until there is only one big cluster left that contains all the elements. This whole process can be visualized as a dendrogram, a tree-like figure that shows exactly at which step two clusters are joined. The distance measure, also called linkage metric, is very important here (note that "distance" is meant as distance between clusters, not between vectors, so Euclidian Distance, for example, is not applicable). The distance measure is written as $d(r,s)$ with r, s distinct cluster indices (also, in the following, x_{ri} and x_{sj} are the vectors assigned to clusters *r* and *s*, respectively), and the major linkage metrics are:

- **Single Linkage** (also "nearest-neighbor linkage"): This metric is defined as the distance between the closest vectors across the clusters, formally

  $$d_{\text{single}}(r,s) = \min(\|x_{ri} - x_{sj}\|). \tag{30}$$

  One of the problems with single linkage is that it is subject to a phenomenon called chaining, which may occur in special kinds of data-sets, and refers to the fact that clusters are wrongly joined in case of outlier data points ("chaining points") connecting them. Single linkage is related to finding the minimal spanning tree. For an example of how single linkage works, see Figure 43a; it shows the distances from cluster I to clusters II and III, obviously I will be joined with III. Single linkage is able to find clusters of arbitrary shape, which means that these clusters are not necessarily close to a cluster center. This makes single linkage unique compared to the other clustering methods discussed in this work, which all discover spherical clusters. A disadvantage of single linkage is that it tends to join single vectors (outliers) very late in the hierarchy, which often leads to many meaningless clusters and few very large ones.
- **Complete Linkage** (also "furthest neighbor linkage"): This distance measure is defined as distance between the two vectors that are farthest away from each other,

  $$d_{\text{complete}}(r,s) = \max(\|x_{ri} - x_{sj}\|). \tag{31}$$

  Figure 43b depicts how complete linkage works with the same clusters as above: Instead of joining clusters I and III, according to this distance metric, I is closer to II. Complete linkage, as opposed to single linkage, finds spherical clusters instead of clusters of arbitrary shape, and does not suffer from chaining effects.

- **Average Linkage**: This distance measure is defined as the mean of the distances between all possible pairs of data points across the clusters, formally

$$d_{average}(r,s)=\frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_s}\left\|x_{ri}-x_{sj}\right\| \tag{32}$$

  where n_r and n_s are the number of samples that belong to C_i and C_j. Average linkage results in a partitioning of the data points that is somewhere in between single and complete linkage partitionings. Average linkage also finds spherical clusters.

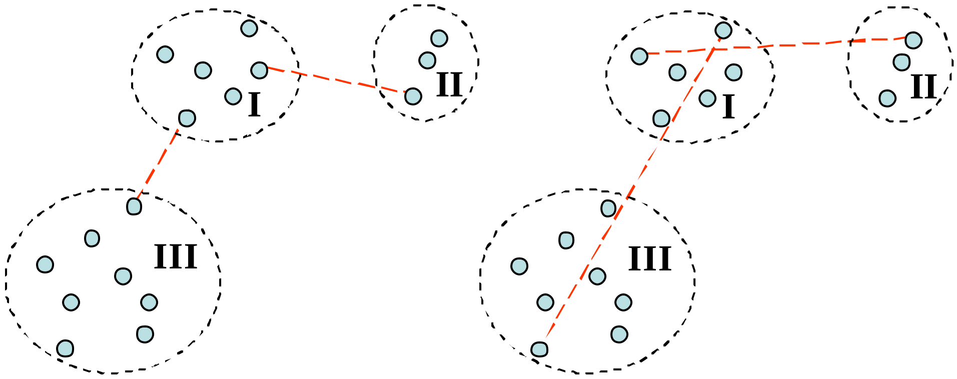

Figure 43a, b: Distances measures of hierarchical clustering methods: Single linkage (a) and complete linkage (b)

Other important linkage methods include Ward's hierarchical clustering method (also "error sum of squares method") and centroid linkage, which will be discussed in Section 5.4. All these algorithms have been proven to be special cases of the equation of Lance and Williams [Lan67].

In Figures 44, 45, and 46, the Iris Map is shown after it has been clustered with the 3 linkage methods described above, and the results depicted for direct comparison of the dendrogram and clustering results for 10, 5, 3 and 2 clusters. Figure 44 shows the results for single linkage, Figure 45 for average linkage, and Figure 46 for complete linkage. There is a significant difference between single linkage and the other two methods. The most interesting figures are the ones that show 3 clusters. In case of average and complete linkage, these approximate the expected division into the regions occupied mainly by either of the 3 iris flower species. Here, single linkage differs: The interpolating border between the upper third and the rest is identified as a cluster. It has to be noted though, that the distances between the clusters are very close in simple linkage distance as can be seen from the dendrogram in Figure 44a The similarities between complete and average linkage are also reflected by their similarly-looking dendrograms.

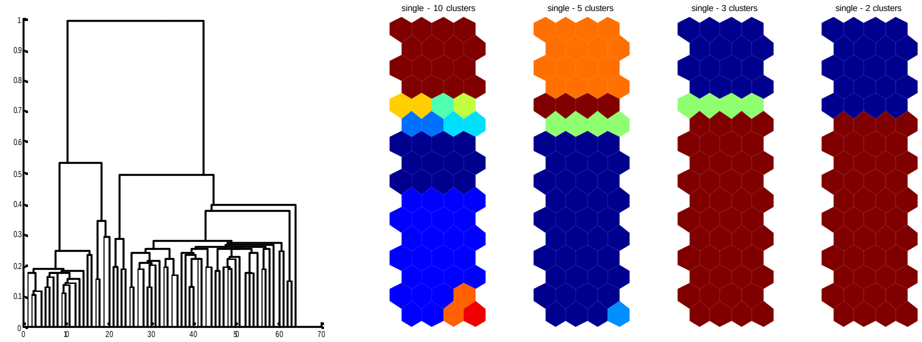

Figure 44a, b, c, d, e: Single linkage of Iris Map: Dendrogram (a), cluster sizes 10 (b), 5 (c), 3 (d) and 2 (e)

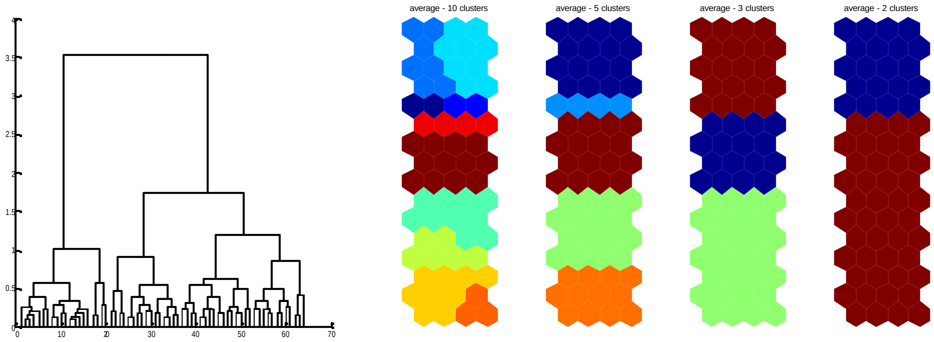

Figure 45a, b, c, d, e: Average linkage of Iris Map: Dendrogram (a), cluster sizes 10 (b), 5 (c), 3 (d) and 2 (e)

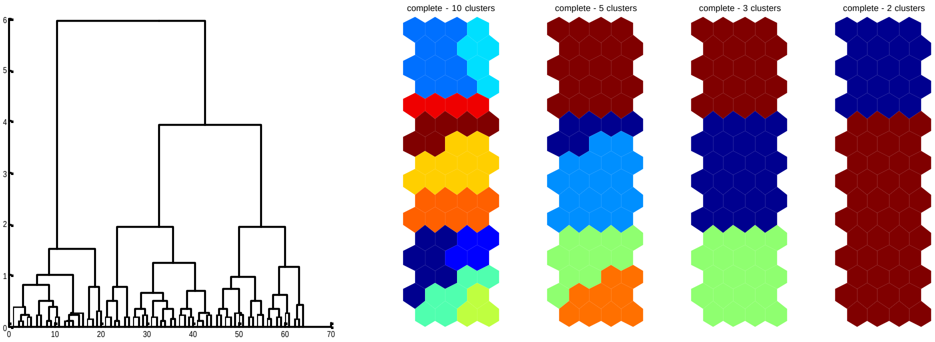

Figure 46a, b, c, d, e: Complete linkage of Iris Map: Dendrogram (a), cluster sizes 10 (b), 5 (c), 3 (d) and 2 (e)

When applied to the Democracy Map, the results for complete and average linkage are also very similar, but not to the same extent as with the Iris Map. Yet again, single linkage differs significantly. The results of the 3 linkage clustering methods are shown in Figures 47, 48 and 49, with plots for 10, 7, 5 and 3 clusters. If the dendrogram for single linkage is as skewed as in this case, this is an indicator that chaining (as described above) has occurred. Also, skewed dendrograms hint that there is one large cluster and many very small ones, in case of the Democracy SOM, many clusters that consist of only one single map unit. Although the partitionings obtained by single linkage differ strongly from the clustering algorithms of the other methods, the result indicates that the region on the center of the bottom (which is recognized as a cluster in Figures 47e) is clearly separated from the rest of the map.

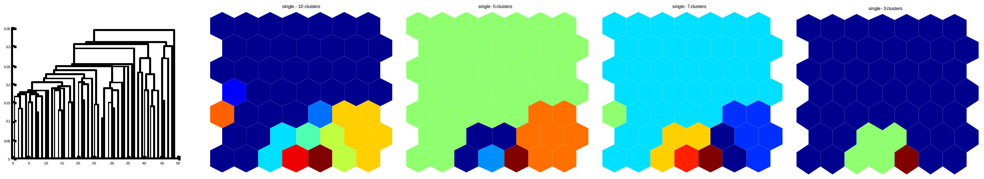

Figure 47a, b, c, d, e: Single linkage of Democracy Map: Dendrogram (a), cluster sizes 10 (b), 5 (c), 3 (d) and 2 (e)

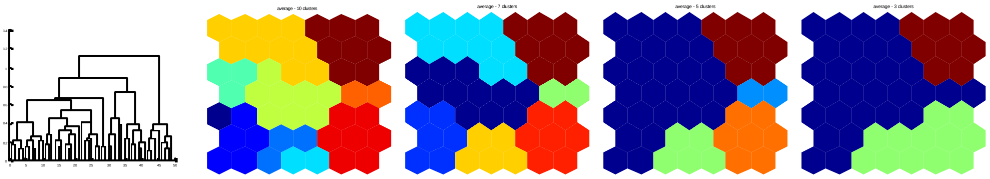

Figure 48 a, b, c, d, e: Average linkage of Democracy Map: Dendrogram (a), cluster sizes 10 (b), 5 (c), 3 (d) and 2 (e)

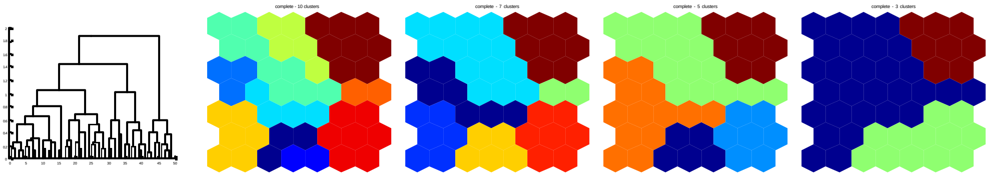

Figure 49 a, b, c, d, e: Complete linkage of Democracy Map: Dendrogram (a), cluster sizes 10 (b), 5 (c), 3 (d) and 2 (e)

5.4. Cluster validity

An important question is what actually makes a cluster a cluster. This is the task of cluster validity measures, which try to evaluate any given partitioning. To find out which partitionings are "good" or "bad", some definitions have to be made. It is commonly agreed upon that the clusters should be as compact and as sharply separated from other clusters as possible. One possible way to describe this formally is

$$\frac{S(C_i)+S(C_j)}{d(C_i,C_j)}, \tag{33}$$

where $S(C)$ measures the compactness or density of cluster C, what is called within- or intra-cluster distance, and $d(C_i,C_j)$ describes the between- or inter-cluster distance. Thus, formula 33 has to be as low as possible for all pairs of clusters to achieve a good clustering.

The distance measures d and S require additional definitions. Some of the most commonly-used between-cluster distances have been introduced in the previous section: d_{single}, $d_{average}$, and $d_{complete}$ in formulas 30, 31 and 32. A distance measure that is required for the definition of an important cluster validity index is called centroid distance, and is shown in Figure 50. The plot shows the clusters as in the examples above (Figure 43), and shows the cluster centroids, which are computed by calculating the mean of all vectors in each cluster. Then, the distance is measured with the usual metric between these points (as indicated by the red lines in Figure 50). This can be expressed formally as

$$d_{centroid}(r,s)=\left\|\overline{x}_r-\overline{x}_s\right\|. \tag{34}$$

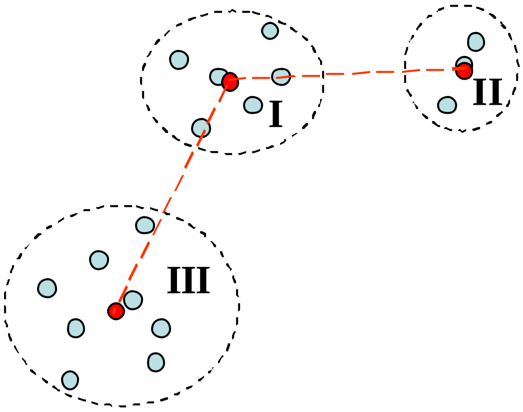

Figure 50: Centroid distance measure between clusters

For measuring the dispersion, the following methods have to be introduced, which are the most important within-cluster distance metrics:

- **Nearest Neighbor:** similar to the concept of single linkage, the average distance between each point and its nearest neighbor is computed. Thus, it is a so-called local method. The formula for this measure can be written as

$$S_{nn}(C_i) = \frac{\sum_{x \in C_i} \min_{x' \in C_i,\, x' \neq x} \{\|x - x'\|\}}{|C_i|}. \quad \textbf{(35)}$$

- **Centroid:** As with $d_{centroid}$, this density measure is based on centroid computation. The average deviation is calculated from the cluster center and indicates how compact the cluster is, formally

$$S_{centroid}(C_i) = \frac{\sum_{x \in C_i} \|x - \bar{x}\|}{|C_i|}. \quad \textbf{(36)}$$

  This method emphasizes spherical clusters, and is non-local since it does not rely on nearest neighbors only.

- **Variance:** Similar to the centroid method, this one favors short distances and small clusters:

$$S_{variance}(C_i) = \sum_{x \in C_i} \|x - \bar{x}\|^2. \quad \textbf{(37)}$$

  This is also a non-local method.

Now that these distance concepts have been introduced, the Davies-Bouldin index (DB-Index, [Dav79]) can be introduced. It is based upon centroid distance measures $d_{centroid}$ and $S_{centroid}$, and thus assumes that (good) clusters are hyperspheres. This way, local methods like single linkage are penalized. The Davis-Bouldin index is computed as

$$I_{DB}(C) = \frac{1}{|C|} \sum_{i=1}^{|C|} \max_{j \neq i} \left\{ \frac{S_{centroid}(C_i) + S_{centroid}(C_j)}{d_{centroid}(C_i, C_j)} \right\}, \qquad \textbf{(38)}$$

where C is the set of clusters.

When applied to the hierarchical clustering methods, this index can be computed for each level of the hierarchy, and since these algorithms are deterministic (in contrast to k-means), the resulting indices are reproducible. Other quality methods exist, especially for hierarchical clustering methods [Hal01], but cannot be introduced here. In Figure 51, the DB-Index is plotted for single (black, solid line), average (blue, dotted) and complete (red, dashed) linkage, where the x-axis refers to the number of clusters, and the y-axis to the corresponding index. This figure shows cluster sizes from 2 to 49, where 49 is the clustering where each cluster consists of only one vector (49 is the number of prototype vectors in the 7x7 map). Complete linkage seems to produce the best clusters according to this index, with peak values at the levels for 4 and 8 clusters. Single linkage is constantly below the other two methods, with one notable exception at 3 clusters (see Figure 47e).

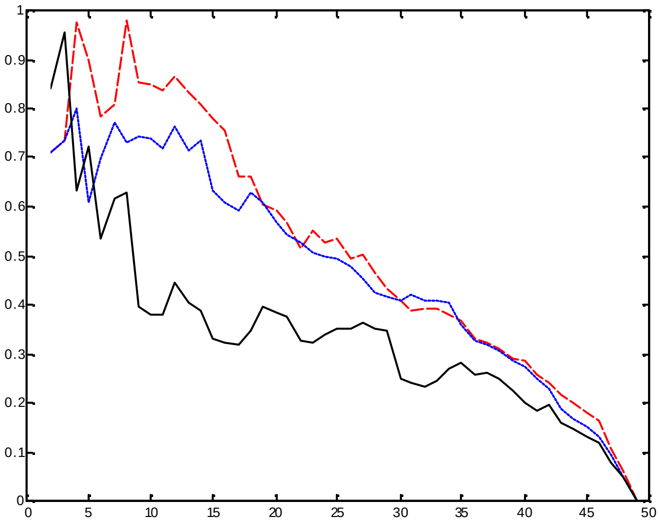

Figure 51: DB-Index for partitionings obtained by hierarchical clustering methods

For k-means clustering, this index is important since it can be used to determine the number of clusters k by computing k-means clusters for a range of k and selecting the partitioning with the highest DB-Index. The results for k between 2 and 15 are depicted in Figure 52a, for each k the best result of 20 runs is given. The peak values are for k = 2, 5 and 13, shown in Figures 52b, 53a and 53b. This is not deterministic and thus not necessarily reproducible. For k = 13, the red cluster on the bottom of the plot is split apart, indicating that there is a slight violation of topology due to interpolating effects.

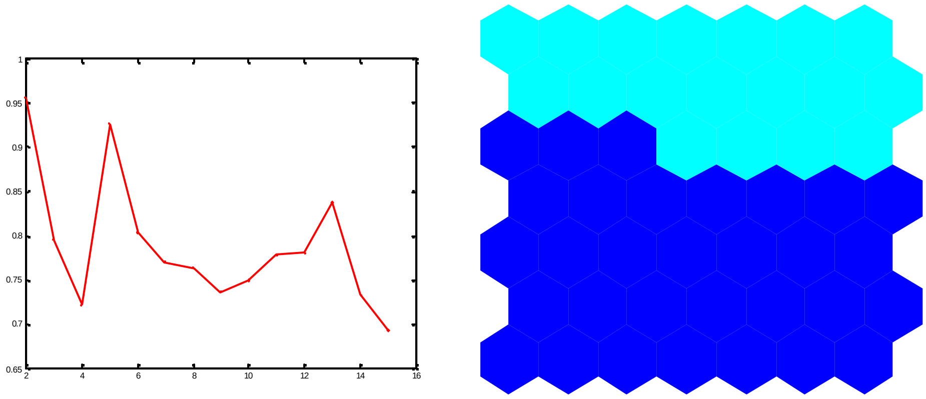

Figure 52a, b: DB-Index of k-means clustering of the Democracy Map by number of clusters (a); k-means of Democracy Map with k=2 (b)

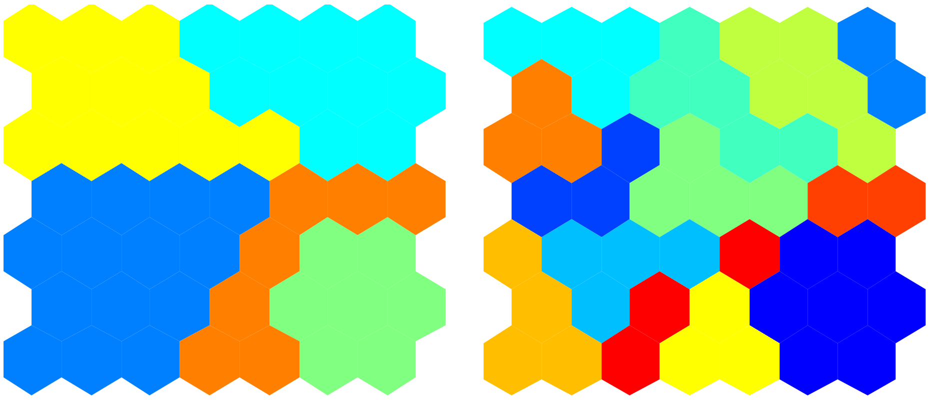

Figure 53a, b: k-means of Democracy Map with k=5 (a); k-means of Democracy Map with k=13 (b)

5.5. Growing Hierarchical SOMs

An extension to the traditional SOM is the Growing Hierarchical Self-Organizing Map (GHSOM) [Dit00, Dit03]. It aims at adapting the net to the data and not vice versa (in a way, the fixed grid size of the traditional SOM is considered a shortcoming). It consists of several layers of rectangular 2-dimensional SOMs that can be arranged and visualized as a quad-tree-like structure. During training, there are two different ways the GHSOM can grow. The first way is that each layer can grow in terms of its prototype units, such that the original 2x2 map size is enlarged by insertion of either a row or a column of new units between existing ones. At a certain point, the map ceases to grow. Then, the units are inspected, and if the samples mapped to one unit are highly different, such that the prototype does not represent the samples precisely enough, another layer of 2x2 units is added below the unit and training is continued as described above. Each of the two growing processes is governed by a parameter. The training and growing procedure is described in more detail in the following paragraphs.

The uppermost layer ("layer 0") holds only a single node, the model of which is the mean of all input samples. Then, the mean quantization error mqe_0 for this prototype vector is computed, which measures the deviation of the samples, formally,

$$mqe_0 := \frac{1}{|X|}\sum_{x_j \in X}\left\|m_0 - x_j\right\| \tag{39}$$

where X is the set of all samples, and m_0 is the single model vector of layer 0. $|X|$ denotes the cardinality of X (the number of samples). Note that this formula is equal to (3), but has to be defined again to match the indexing and notation schemes of the GHSOM. In case of the single unit layer 0, all the sample vectors are mapped to this unit. The value mqe_0 will be referred to later; it denotes how far the data set is spread in input space. Below layer 0, layer 1 with initially 4 (2x2) units is created and trained according to the usual SOM learning rule (as described in Section 3.4.). After a previously defined number of steps l of the training process, the mean quantization errors for all the units are computed,

$$mqe_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} m_i - x_j \,, \tag{40}$$

with C_i the subset of the samples for which unit i is the BMU. Furthermore, the map's mean quantization error *MQE* can be determined, formally written as

$$MQE_m = \frac{1}{|U|} \sum_{i \in U} mqe_i \,. \tag{41}$$

As long as

$$MQE_m < t_1 \cdot mqe_u \tag{42}$$

holds, the training of the current map is continued (mqe_u is the quantization error of the corresponding unit u in the upper hierarchy layer). Here, the first parameter t_1 comes into play that delimits the growth of the map's size. If the stopping criterion is not met, the *error unit e* is determined, namely the unit with the largest mean quantization error, formally

$$e = \underset{i}{\operatorname{argmax}}(mqe_i) \,. \tag{43}$$

Then the *most dissimilar unit d is* computed, that is, one of up to 4 neighbors of e with the largest distance in input space. Between these two units a row or column of units is inserted that are initialized with an interpolated value (i.e. mean) of the existing neighboring units. Figure 54 shows this kind of growth process.

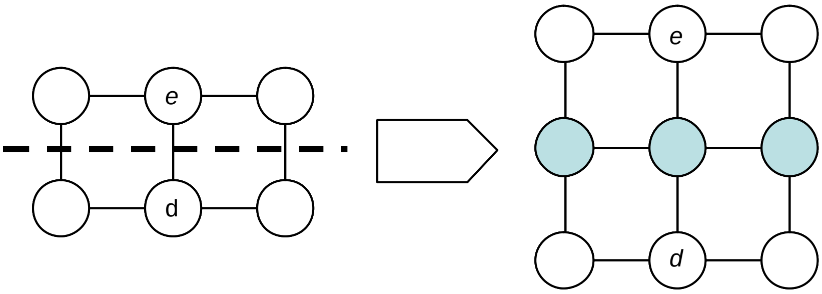

Figure 54:Insertion of a row between error unit e and most dissimilar unit d

After that, the standard SOM training process is continued for another l steps, and when the rule in formula 42 does not hold anymore, training of the current layer is finished. Then the hierarchical growing is applied, if the criterion

$$mqe_i < t_2 \cdot mqe_0 \quad \textbf{(44)}$$

is met, where t_2 is the second parameter. All units refer to the layer 0 unit's quantization error regardless on which layer the current node is located. Note that this growing process does not occur always, only if the unit still requires a more detailed representation. Also, it does not occur evenly across one layer, it is for example possible that a node is finished with training while its neighboring unit requires one (or even more) layers of fine-tuning. If this is the case, another SOM of initially 2x2 nodes is created on the next layer (see Figure 55), and trained with the subset of the samples for which the upper unit is the BMU.

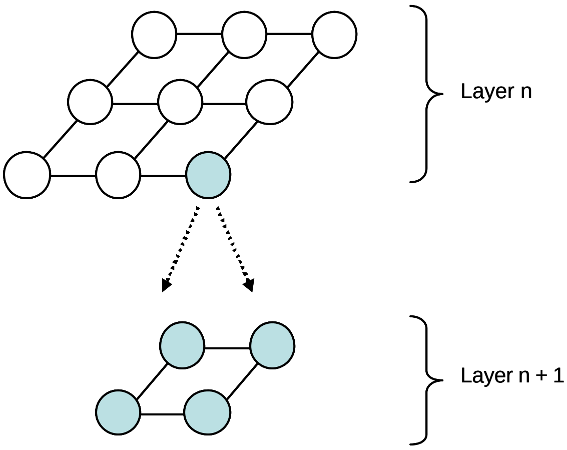

Figure 55: Hierarchical growth process of the GHSOM

Thus, the parameters t_1 and t_2 define the thresholds for the two growth processes. Both parameters have to be between 0 and 1. Relatively small values of t_1 lead to a lengthy growth of each layer and big maps, while large values lead to a shorter training of each map and thus to smaller map sizes. If parameter t_2 is relatively small, units tend to be expanded on the next layer more easily, while large values result in flat hierarchies.

Trained GHSOMs can be visualized in a quad-tree like way, with deeper layers nested in their respective preceding unit. Most of the visualization techniques described in the previous chapters can not be applied to the GHSOM or are more difficult to apply, like the U-Matrix. However, hit-histogram based visualization and component planes can be visualized as with traditional SOMs.

Figure 56 shows a GHSOM with t_1 = 0.8 and t_2 = 0.0001. The plot has been labeled such that the lowest level units are labeled with the countries they are closest. It is apparent that the GHSOM leads to a very low number of interpolating units (at least at upper layers, the lowest layers do actually contain units that are not BMU of any sample). The map's layer 1 has 4 units, the growth process has not enlarged the grid of this map. Also, it is very similar to the traditional SOM with regard to the position of the countries of the map. Figure 57 shows a component plane (H7 – Life expectancy at birth) of the same map. At an overall impression, this method results in a very similar distribution as the traditional Democracy SOM.

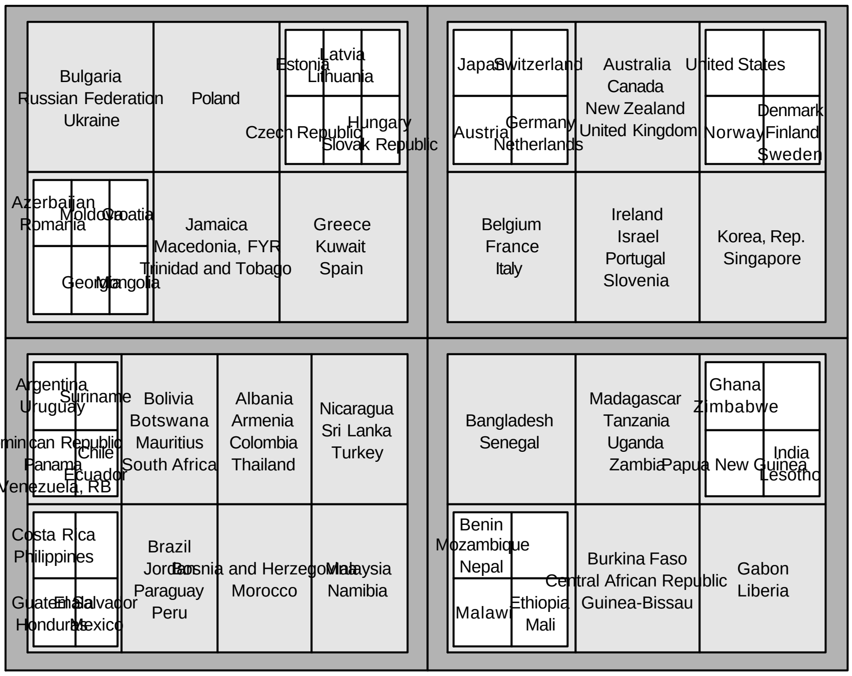

Figure 56: Labeled Democracy GHSOM

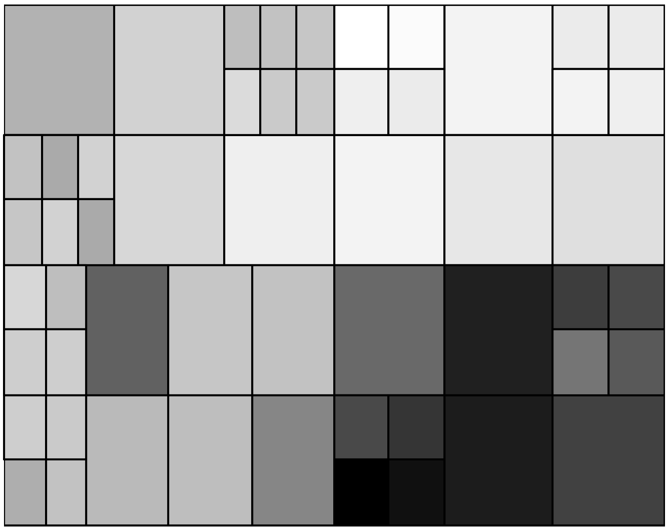

Figure 57: Component plane H7 (grayscale)

Another way to label the map is the LabelSOM method as described in Section 4.6. It emphasizes on showing the characteristic dimensions for which the samples mapped to a node are similar. Figure 58 shows a plot of the GHSOM labeled with this method (note that only units with at least 2 samples are shown). Obviously, there are two dimensions that dominate this visualization type, P5 and P6. This is due to the fact that both of these indicators are binary valued, which hold the value "1" for most of the countries (approximately 95%).

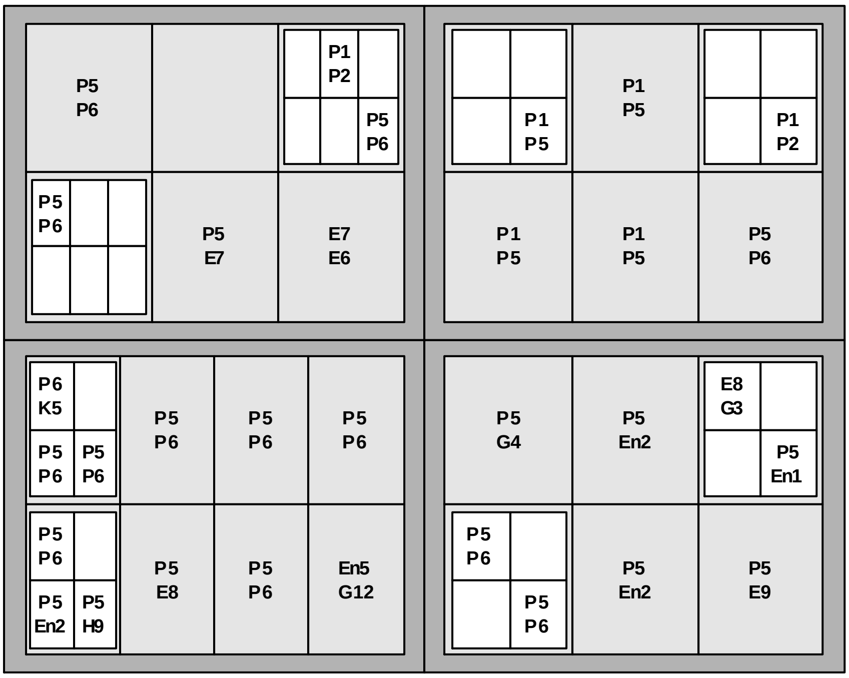

Figure 58: LabelSOM method applied to GHSOM

6. In-depth discussion of the Democracy SOM

6.1. Overview

In this chapter, specific aspects of the Democracy SOM will be investigated. The visualization techniques and clustering methods will be exploited to provide interesting insights into the Democracy data manifold and what can be learnt from it using the SOM, and how these results can be presented. To achieve this, several plots will be combined to reflect different findings simultaneously and to maximize the amount of information that is transported in a single plot. Also, a-priori knowledge will be taken into account, like membership to the NATO, or the fact that dimensions can be grouped to their indicator categories like "Health" or "Political System". These real-world properties will be compared to the Democracy SOM.

The rest of this chapter is organized as follows:

Section 6.2. investigates labeling and coloring schemes according to geographic location (continents) of the countries. Section 6.3. presents hit histograms by countries according to membership of treaties and political unions. In section 6.4., the 60 component planes of the Democracy data are reduced to the 6 main categories, and thus a "score" is determined for each of them; several ways to visualize this will be shown.

6.2. Visualizing the map according to continent distribution

In this section, the countries are assigned labels according to the continent they are located on. Figure 59a shows a map where these labels have been substituted by colors: The map shows the results from the voting procedure, where only the label with the most occurrences is kept. The colors have the following meaning:

- Red: Europe
- Green: Africa
- Yellow: South America
- Blue: Asia
- Orange: North America
- Light blue: Australia

- Black: no samples assigned to this unit

Note that the colors are not distributed evenly at all, this is partly due to the fact that the continents are not represented by the same number of countries each (i.e. there are 34 samples from European countries, but only 2 from Australia). It can be seen that from the upper right to the upper center part of the map European countries clearly dominate. North American and Australian countries are present only in this area. The center left is occupied mainly by South American and the bottom right by African countries.

A similar approach with a different visualization type is shown in Figure 59b. Here, pie charts are shown that reflect the relevance of each continent to a unit. Empty units indicate that no country has been assigned to it. The colors refer to the continents as with the previous plot.

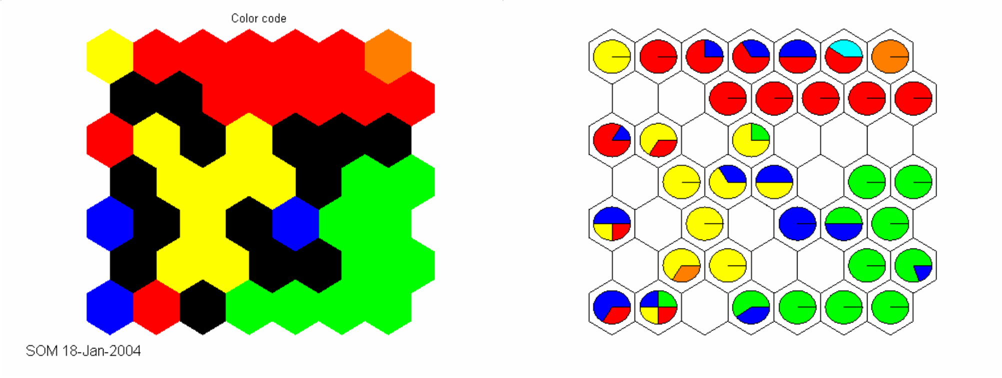

Figure 59a, b: Most dominant continent (a); relative distribution of continents on Democracy SOM as pie chart visualization (b)

However, both plots do not show how many countries are actually mapped to any of the units, except for the black patches and the missing pie charts for interpolating units. Figure 60 shows the pie chart plot again where the size of the unit corresponds to its total number of hits, according to the concept introduced in Section 4.1.3. Thus, the plot reveals both how many countries are mapped to a unit, and the relative importance of each continent to this unit.

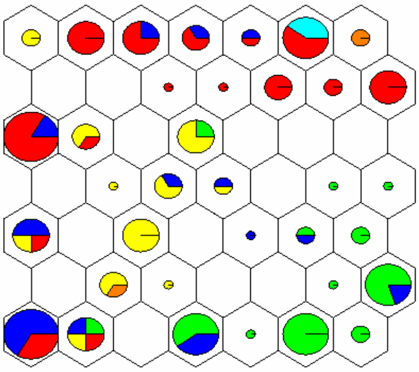

Figure 60: Relative distribution of Democracy Map with scaled pie charts reflecting the absolute number of hits

6.3. Hit histograms by treaties

In case of the Democracy SOM, the hit histogram visualization is used extensively to show the distribution of countries by certain criteria (like membership of a specific treaty). Figure 61 provides interesting insight in the location of the treaties' members and their relation to the distribution according to the Democracy Map, the following observations can be made:

- Most of the treaties' countries are actually very close together in output space, they are obviously close together in terms of democratic value, with some outliers.
- The EU (European Union), OECD (Organization for Economic Co-operation and Development), AU (African Union) and NATO (North-Atlantic Treaty Organization) form the most compact clusters.
- The countries of the OAS (Organization of American States) are mostly on adjacent map units with some minor exceptions.
- Countries of OIC (Organization of the Islamic Conference), APEC (Asia-Pacific Economic Cooperation), OSCE (Organization for Security and Cooperation in Europe) and CE (Council of Europe) are spread between rather distant regions of the map, so there seems to be no correlation between membership of any of these to the democratic score.

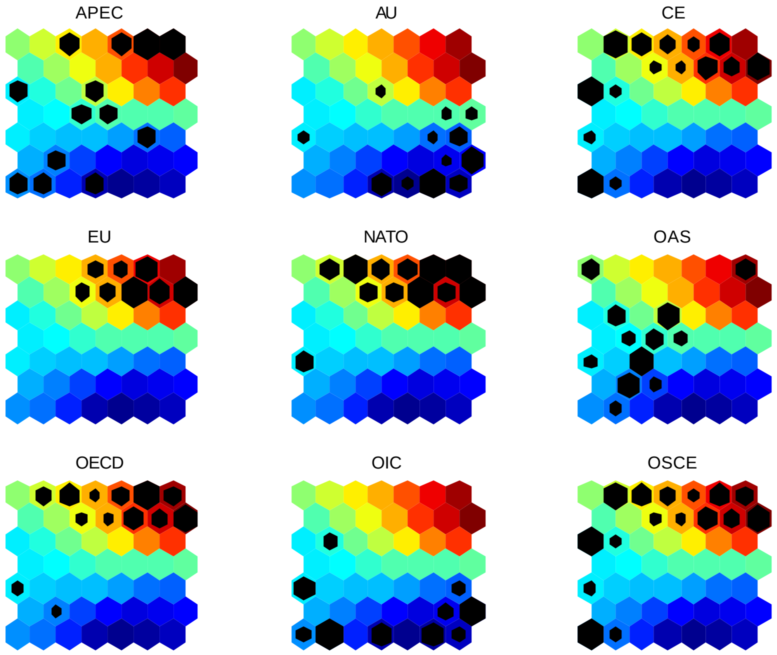

Figure 61: Hit histograms of treaties

6.4. Computing a "score" for qualities within indicator categories

Another approach takes the meaning of the dimensions and the possibility to group them into major categories (like "Health", "Environmental Sustainability") into account. This is a two-step procedure, first the variables have to be reduced to their categories, then the 6 resulting values can be interpreted as if they were component planes. Since the 60 original indicators have been selected in a way that higher values always mean "better", these aggregated values can be interpreted as a "score" of the values in question. This is done by simply calculating the weighted average of each category's indicators (see Chapter 2 for the categories and relative weights). This score can be written formally as

$$score(x) = \frac{1}{\sum_{i \in I} w_i} \sum_{i \in I} w_i \cdot x_i \, , \qquad \textbf{(45)}$$

where x is the data vector for which the score has to be determined, and I is a set of indices that denotes which dimensions should be regarded.

The resulting score can be visualized for the prototype vectors similar to a component plane. For the Democracy data, 7 scores have been calculated: one for each category (Figure 62) and a total score (Figure 63) that averages all of the variables (in this case, I holds indices of all dimensions). This overall score is an attempt to measure the quality of a democracy very similar to the evaluating scheme of the Democracy Award. These quality measures allow a series of conclusions:

- **Political System**: very similar to overall score, not surprisingly since this category is as influential as the rest of the categories combined.
- **Knowledge**: similar to overall score and politics dimension, but the lowest values are on the rightmost part of the bottom instead of the middle part.
- **Health**: high values in this category are spread towards regions with a medium score, and similar to the Knowledge dimension, the worst score is assigned to the bottom right corner.
- **Economy**: very bad score along the left border (i.e. Russian Federation), apart from that, similar to Knowledge
- **Gender Equality**: very similar to Health, but the countries in the bottom part of the map receive slightly below average values
- **Environmental Sustainability**: almost directly opposed to other ratings; good values for lower right (Liberia, Malawi, mostly developing countries), bad values for upper right (US, Canada, EU) and center of left border (Russian Federation, less developed former Eastern Bloc). This is partly due to the fact that this category consists of only 5 indicators, 2 of which measure CO_2 emissions, and several samples have many MVs.

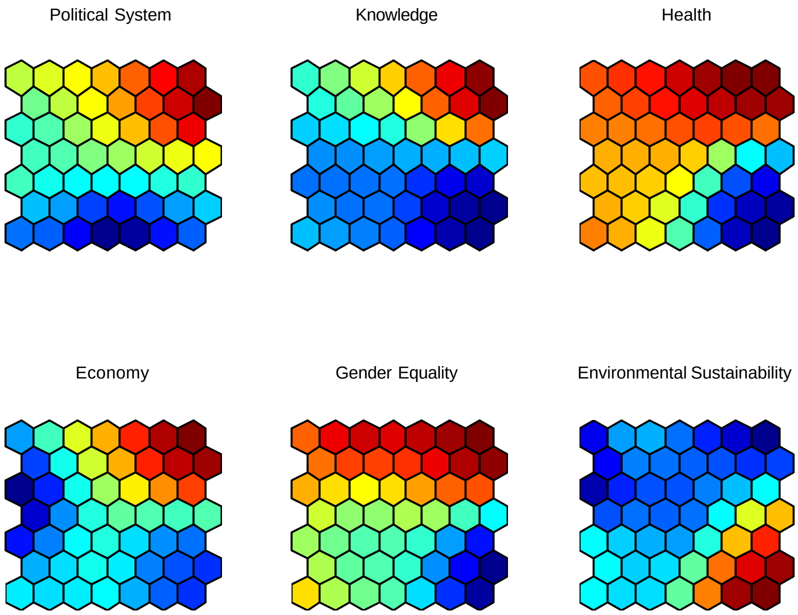

Figure 62: Reduced component planes

Figure 63 shows a color coded map with the total Democracy score. It also shows the labels for the countries. Again, it can be seen from all of these plots that there seems to be a border on the right center of the map. Apart from this gap, the visualizations are mostly very continuous. This leads to the assumption that the data is ordered this way, in a U-shape manner, from the lower right, to the left, and to the upper right according to its quality of democracy.

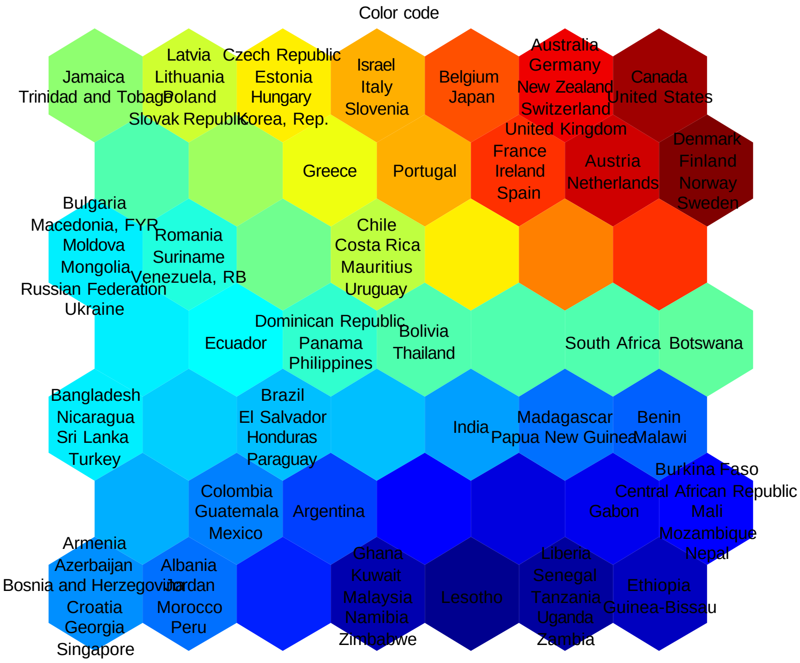

Figure 63: Democratic score

The scores can of course be visualized in a single plot with multiple bar charts, which is especially helpful in discovering correlations between the indicator categories. As opposed to Figure 40 (in section 4.10.) which depicted all of the 60 indicators at once, this visualization type does make sense with only 6 components to be visualized. Figure 64 shows the reduced planes (the order of the bars, from left to right: "Political System" (red), "Knowledge" (yellow), "Health" (green), "Economy" (cyan), "Gender Equality" (blue), and "Environmental Sustainability" (purple)). This figure is, however, not entirely accurate and misleading with respect to the relative importance of the "Political System" dimension, which has a 5 times higher weight than the other dimensions. To reflect the real significance of the first component here, Figure 65 shows the same but with repeating the PS-category due to its true influence, the rest of the categories are shifted to the right. The purpose of this second plot is to explain how the overall score is acquired.

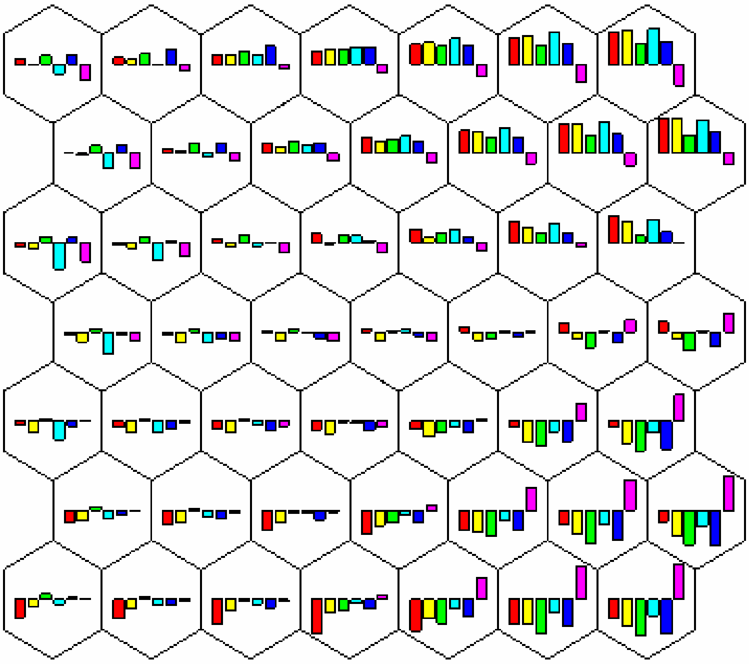

Figure 64: Reduced components to 6 main categories visualized as box plots

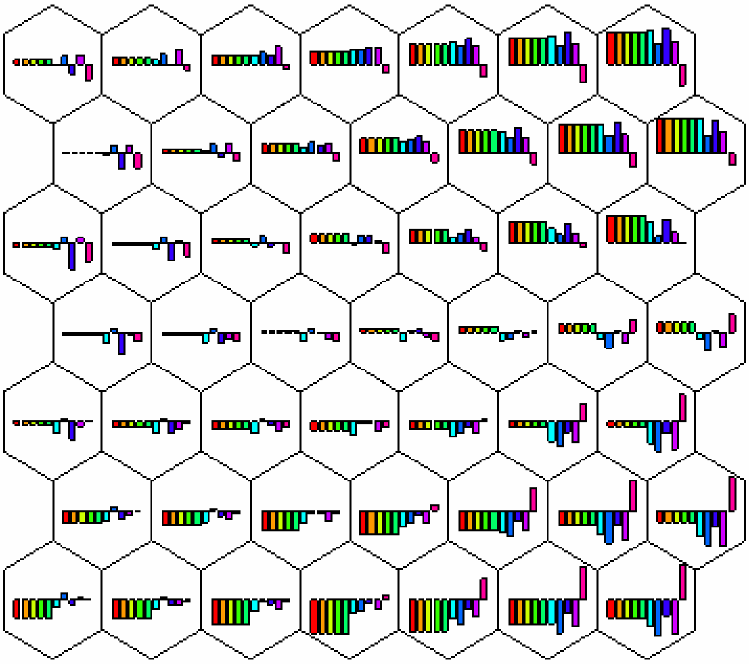

Figure 65: Reduced components to 6 main categories visualized as box plots, repeating "Politics" component 4 times to reflect its importance according to the Pilot Ranking

7. Conclusion

In this thesis, the SOM algorithm and its application to a political data set have been investigated.

In Capter 2, the data set, consisting of countries and political indicators, has been described. Also, a benchmark data set was introduced and several distance metrices between the data points have been investigated.

In Chapter 3, the Self-Organizing Map was discussed, and compared to similar algorithms from the areas of Vector Projection and Quantization. Further, several ways to initialize and configure the SOM has been shown. The Democracy SOM has been described in detail, which served as the basis for experiments in the later chapters. Also, several ways to deal with missing values have been shown, where especially the interpolation method using SOMs has proven very valuable.

In Chapter 4, a range of common visualization methods of the SOM has been described. These visulizations have been applied to the Democracy SOM to provide insight into the distribution of the countries on the map. Other techniques showed the values of individual indicators across the map, labels, local distanced between neighboring units, and multiple components. In particular, the visualizations revealed the rather linear nature of the Democracy data set, and hinted that the countries on the map are distributed in groups that reflect how developed and advanced the countries are. The Democracy SOM is evenly crowded along its upper, left and lower parts, while there is an interpolating gap in the center of the right part that separates the least developed countries from the most developed ones.

In Chapter 5, clustering methods of the SOM have been described. The purpose of clustering is to identify coherent regions, and several partitionings of the map have been shown. Hierarchical clustering methods have been compared to the k-means algorithm, and have been tested by a validity measure. The experiments reaffermed the assumption that the only large gap has been the interpolating region on the right side of the map, while the map was slightly less coherent on the lower side than on the upper side. Further, a variant of the SOM, the Growing Hierarchical SOM, has been discussed.

The combination of the visualization concepts to investigate whether the Democracy SOM (and also the underlying data set) can be compared to real-world categories like political treaties has been perfomed in Chapter 6. First, the geographical distribution was analyzed. It was obvious, that the upper part was largely dominated by European, and the lower part mostly by African countries. Then, hit histograms were used to show the positions of members of certain political and economic treaties on the map. Most of them have been located in contingent areas of the map (with a few exceptions), which indicates that the members of those treaties are on similar levels according to democratic value. Finally, an attempt was made to reduce the indicators to a single "score" that reflects this democratic value on a qualitative scale, similar to the Democracy Award Pilot Ranking. These experiments showed that the highest score was assigned to the upper right area (EU, USA), steadily declining to the lower right area (developing countries).

The SOM has provided several interesting insights into the Democracy data set. It is also very interesting that the SOM's way to handle missing values produces very similar results as the Pilot ranking approach. Regarding the data set, the SOM has revealed that the countries ranked in the upper half in terms of democratic score have a very stable linear correlation regarding the dimensions, while on the lower end of the ranking, the variables are not as related anymore. This is revealed by the reduced compont planes, which partly show the low values in different places, while high values are constantly in the upper right region. The distribution of the countries according to treaties and continents affirmed this hypothesis, since treaties between industrial countries are more compact that i.e. the Organization of the Islamic Conference.

Further research could aim at automatic report generation or finding more diverse visualization methods. The most interesting part of this work was definitely Chapter 6, and efforts could be undertaken to furher simplify SOM visualizations for mainstream use. These visulizations are relatively easy to comprehend even to persons who are not experts with SOMs, so maybe this could be a promising research field.

In the field of political sciences, it will be interesting to see how the quality measurement of democracies evolves. Currently, there are very few projects that try to rank democracies qualitatively, and research in this area is rather limited. The Global Democracy Award could be an important stimulus for this type of academic research.

Appendix A: Countries and Data Collections

List of Countries

Country Name	Abbreviation	Country Name	Abbreviation
Albania	ALB	Liberia	LBR
Argentina	ARG	Lithuania	LTU
Armenia	ARM	Macedonia, FYR	MKD
Australia	AUS	Madagascar	MDG
Austria	AUT	Malawi	MWI
Azerbaijan	AZE	Malaysia	MYS
Bangladesh	BGD	Mali	MLI
Belgium	BEL	Mauritius	MUS
Benin	BEN	Mexico	MEX
Bolivia	BOL	Moldova	MDA
Bosnia and Herzegovina	BIH	Mongolia	MNG
Botswana	BWA	Morocco	MAR
Brazil	BRA	Mozambique	MOZ
Bulgaria	BGR	Namibia	NAM
Burkina Faso	BFA	Nepal	NPL
Canada	CAN	Netherlands	NLD
Central African Republic	CAF	New Zealand	NZL
Chile	CHL	Nicaragua	NIC
Colombia	COL	Norway	NOR
Costa Rica	CRI	Panama	PAN
Croatia	HRV	Papua New Guinea	PNG
Czech Republic	CZE	Paraguay	PRY
Denmark	DNK	Peru	PER
Dominican Republic	DOM	Philippines	PHL
Ecuador	ECU	Poland	POL
El Salvador	SLV	Portugal	PRT
Estonia	EST	Romania	ROM
Ethiopia	ETH	Russian Federation	RUS
Finland	FIN	Senegal	SEN
France	FRA	Singapore	SGP
Gabon	GAB	Slovak Republic	SVK
Georgia	GEO	Slovenia	SVN
Germany	DEU	South Africa	ZAF
Ghana	GHA	Spain	ESP
Greece	GRC	Sri Lanka	LKA
Guatemala	GTM	Suriname	SUR
Guinea-Bissau	GNB	Sweden	SWE
Honduras	HND	Switzerland	CHE
Hungary	HUN	Tanzania	TZA
India	IND	Thailand	THA

Ireland	IRL	Trinidad and Tobago	TTO
Israel	ISR	Turkey	TUR
Italy	ITA	Uganda	UGA
Jamaica	JAM	Ukraine	UKR
Japan	JPN	United Kingdom	GBR
Jordan	JOR	United States	USA
Korea, Rep.	KOR	Uruguay	URY
Kuwait	KWT	Venezuela, RB	VEN
Latvia	LVA	Zambia	ZMB
Lesotho	LSO	Zimbabwe	ZWE

List of Indicators

Political System

Indicator name	Description	Relative importance	Overall influence
P1	Political rights	12.5 %	6.25 %
P2	Civil liberties	12.5 %	6.25 %
P3	Freedom of Press	12.5 %	6.25 %
P4	Transparency versus corruption	12.5 %	6.25 %
P5	Change of the government head	12.5 %	6.25 %
P6	Partial or complete change of government parties	12.5 %	6.25 %
P7	Duration of months with female head(s) of government	12.5 %	6.25 %
P8	Average percentage share of female cabinet members	12.5 %	6.25 %

Gender Equality (Educational and Economic)

Indicator name	Description	Relative importance	Overall influence
G1	Employees, agriculture, female/male (% of economically active female/male population)	5 %	0.5 %
G2	Employees, industry, female/male (% of economically active female/male population)	5 %	0.5 %
G3	Employees, services, female/male (% of economically active female/male population)	5 %	0.5 %
G4	Labor force activity rate, female/male (% of female/male population ages 15-64)	5 %	0.5 %
G5	Labor force activity rate, female (% of female population ages 15-64)	10 %	1 %
G6	Unemployment, female/male (% of female/male labor force)	5 %	0.5 %
G7	Unemployment, female (% of female labor force)	10 %	1 %
G8	Primary education, pupils (% female)	10 %	1 %
G9	School enrollment, secondary, female (% gross)	10 %	1 %
G10	School enrollment, secondary, female (% net)	10 %	1 %

G11	Illiteracy rate, adult female/male (% of females/males ages 15 and above)	5 %	0.5 %
G12	Illiteracy rate, adult female (% of females ages 15 and above)	10 %	1 %
G13	Life expectancy at birth, female (years)	10 %	1 %

Economy

Indicator name	Description	Relative importance	Overall influence
E1	Central government debt, total (% of GDP)	11.1 %	1.1 %
E2	GDP per capita, PPP (current international $)	11.1 %	1.1 %
E3	GNI per capita, PPP (current international $)	11.1 %	1.1 %
E4	Overall budget deficit, including grants (% of GDP)	11.1 %	1.1 %
E5	Inflation, consumer prices (annual %)	11.1 %	1.1 %
E6	Food price index (1995 = 100)	11.1 %	1.1 %
E7	Labor force, children 10-14 (% of age group)	11.1 %	1.1 %
E8	Unemployment, total (% of total labor force)	11.1 %	1.1 %
E9	Unemployment, youth total (% of total labor force ages 15-24)	11.1 %	1.1 %

Health

Indicator name	Description	Relative importance	Overall influence
H1	Health expenditure per capita, PPP (current international $)	5.6 %	0.56 %
H2	Health expenditure, private (% of GDP)	5.6 %	0.56 %
H3	Health expenditure, public (% of GDP)	5.6 %	0.56 %
H4	Hospital beds (per 1,000 people)	5.6 %	0.56 %
H5	Immunization, DPT (% of children under 12 months)	5.6 %	0.56 %
H6	Immunization, measles (% of children under 12 months)	5.6 %	0.56 %
H7	Life expectancy at birth, total (years)	50 %	5 %
H8	Mortality rate, infant (per 1,000 live births)	5.6 %	0.56 %
H9	Mortality rate, under-5 (per 1,000 live births)	5.6 %	0.56 %
H10	Physicians (per 1,000 people)	5.6 %	0.56 %

Knowledge

Indicator name	Description	Relative importance	Overall influence
K1	School enrollment, secondary (% gross)	6.7 %	0.67 %
K2	School enrollment, secondary (% net)	6.7 %	0.67 %
K3	School enrollment, tertiary (% gross)	6.7 %	0.67 %
K4	Pupil-teacher ratio, primary	6.7 %	0.67 %
K5	Illiteracy rate, adult total (% of people ages 15 and above)	6.7 %	0.67 %

K6	Daily newspapers (per 1,000 people)	6.7 %	0.67 %
K7	Telephone mainlines (per 1,000 people)	6.7 %	0.67 %
K8	Television sets (per 1,000 people)	6.7 %	0.67 %
K9	Personal computers (per 1,000 people)	6.7 %	0.67 %
K10	Internet hosts (per 10,000 people)	6.7 %	0.67 %
K11	Internet users (per 1,000 people)	6.7 %	0.67 %
K12	Mobile phones (per 1,000 people)	6.7 %	0.67 %
K13	Information and communication technology expenditure (% of GDP)	6.7 %	0.67 %
K14	Research and development expenditure (% of GNI)	6.7 %	0.67 %
K15	Scientists and engineers in R&D (per million people)	6.7 %	0.67 %

Environmental Sustainability

Indicator name	**Description**	**Relative importance**	**Overall influence**
En1	CO2 emissions, industrial (kg per PPP $ of GDP)	20 %	2 %
En2	CO2 emissions, industrial (metric tons per capita)	20 %	2 %
En3	GDP per unit of energy use (PPP $ per kg of oil equivalent)	20 %	2 %
En4	Organic water pollutant (BOD) emissions (kg per day per worker)	20 %	2 %
En5	Organic water pollutant (BOD) emissions (kg per day)	20 %	2 %

Glossary

ANN	artificial neural network
BMU	best-matching unit
CCA	Curvilinear Component Analysis
GDA	Global Democracy Award
KMC	k-means clustering
MSE	mean squared error
MV	missing value
NLM	non-linear mapping
PCA	principle component analysis
SOM	self-organizing map
SVD	Singular Value Decomposition
VP	vector projection
VQ	vector quantization

Bibliography

[Ber02] P. Berkhin. *Survey Of Clustering Data Mining Techniques*. Technical Report, Accrue Software, San Jose, CA, USA, http://www.accrue.com/products/rp_cluster_review.pdf, 2002

[Cam02] D. Campbell, M. Sükösd. *Feasibility Study for a Quality Ranking of Democracies*. Wien, Budapest, http://www.global-democracy-award.org/downloads/feasibility_study.pdf, 2002

[Cam03] D. Campbell, M. Sükösd. *Global Quality Ranking of Democracies: Pilot Ranking 2000*. Wien, Budapest, http://www.global-democracy-award.org/downloads/folder_a4pdf, 2003

[Dav79] D. Davies, D. Bouldin. *A cluster separation measure*. IEEE Transactions on Pattern Analysis & Machine Intelligence 1(4): 224-227, 1979

[Deb98] G. Deboeck, T. Kohonen (eds). *Visual Explorations in Finance with Self-Organizing Maps*. Springer, 1998

[Dem97] P. Demartines, J. Hrault. *Curvilinear component analysis: A selforganizing neural network for nonlinear mapping of data sets*, IEEE Transactions on Neural Networks, 8(1): 148-154, 1997

[Dit00] M. Dittenbach, D. Merkl, A. Rauber, A. *The Growing Hierarchical Self-Organizing Map*. Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000), pages 15-19, IEEE Computer Society, Como, Italy, July 24-27 2000

[Dit03] M. Dittenbach. *The growing hierarchical self-organizing map: Uncovering hierarchical structure in data*. Journal of the Austrian Society for Artificial Intelligence (ÖGAI) 22(3): 25-28, October 2003

[Eck80] T. Eckes, H. Roßbach. *Clusteranalysen*, Kohlhammer Standards Psychologie, Teilgebiet: Methoden. Verlag W. Kohlhammer, Stuttgart, Berlin, Köln, Mainz, 1980

[FH01] Freedom House. *Freedom in the World. The Annual Survey of Political Rights and Civil Liberties 2000-2001*. Washington, D.C. 2001

[Fis36] R. Fisher. *The use of multiple measurements in taxonomic problems*. The Annals of Eugenics, 7: 179-188, 1936

[Fle97] A. Flexer. *Limitations of self-organizing maps for vector quantization and multidimensional scaling*. Advances in Neural Information Processing Systems 9, pages 445-451, 1997

[Fri94] B. Fritzke. *Growing Cell Structures - a Self-organizing Network for Unsupervised and Supervised Learning*. Pergamon Press, Neural Networks, 7(9): 1441-1460, 1994

[Gra84] R. Gray. *Vector Quantization*. IEEE ASSP Magazine, pages 4-29, April 1984

[Hal01] M. Halkidi, Y. Batistakis, M. Vazirgiannis, *On Clustering Validation Techniques*. Intelligent Information Systems Journal, Kluwer Pulishers, 17(2-3): 107-145, 2001

[Jai99] A. Jain, M. Murty, P. Flynn. *Data Clustering: A Review*. ACM Computing Surveys 31(3): 264-323, 1999

[Kas98] S. Kaski, J. Kangas, T. Kohonen. *Bibliography of Self-Organizing Map (SOM) Papers: 1981-1997*. Neural Computing Surveys, 1: 102-350, 1998

[Koh01] T. Kohonen. *Self-Organizing Maps*. Volume 30 of Springer Series in Information Sciences. Springer, Berlin, Heidelberg, 3rd edition 2001

[Koh91a] T. Kohonen, K. Mäkisara, O. Simula, J. Kangas, editors. *Artificial Neural Networks*. Elsevier Science Publishers, 1991

[Koh91b] T. Kohonen. *Self-organizing maps: optimization approaches*. In Kohonen, Mäkisara, Simula, Kangas, eds, Artificial Neural Networks. Proceedings of ICANN'91, International Conference on Artificial Neural Networks, North-Holland, Amsterdam, 2: 981-990, 1991

[Koi94] P. Koikkalainen. *Progress with the tree-structured self-organizing map*. Cohn, ed, 11th European Conference on Artificial Intelligence. European Committee for Artificial Intelligence (ECCAI), John Wiley & Sons, Ltd., August 1994

[Lam00] J. Lampinen, T. Kostiainen. *Self-Organizing Map in data analysis – notes on overfitting and overinterpretation*. Proceedings of ESANN' 2000, Bruges, Belgium, pages 239-244, April 2000

[Lan67] G. Lance and W. Willams. *A General theory of classification sorting strategies*. 1st Hierarchical Systems, Computer Journal, 9: 373-380, 1967

[Mac67] J. MacQueen. *Some Methods for Classification and Analysis of Multivariate Observations*. Proceedings of 5th Berkley Symposium on Mathematical Statistics and Probability, Volume I: Statistics, 281-297, 1967

[Mar93] T. Martinetz, S. Berkovich, K. Schulten. *"Neural Gas" Network for vector quantization and its application to time-series prediction*. IEEE Transactions on Neural Networks, 4(4): 558-569, 1993

[Mer98] D. Merkl and A. Rauber. *CIA's view of the world and what neural networks learn from it: A comparison of geographical document space representation metaphors*. Proceedings of the 9th International Conference on Database and Expert Systems Applications (DEXA'98), Vienna, Austria, Lecture Notes In Computer Science. Springer, pages 816-825, 1998

[Oja03] M. Oja, S. Kaski, and T. Kohonen, *Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum*. Neural Computing Surveys, 3: 1-156, 2003

[Pam02] E. Pampalk, A. Rauber, and D. Merkl. *Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps*. Proceedings of the International Conference on Artifical Neural Networks (ICANN), pages 871-876, 2002

[Rau99] A. Rauber. *LabelSOM: On the Labeling of Self-Organizing Maps.* Proceedings of the International Joint Conference on Neural Networks (IJCNN'99), Washington, DC, July 10 - 16, 1999

[Ric38] M. Richardson. *Psychological Bulletin*. 35: 659-660, 1938

[Sam69] J. Sammon. *A Non-Linear Mapping for Data Structure Analysis.* IEEE Transactions on Computers, C-18(5): 401-409, May 1969

[Sip01] M. Siponen, J. Vesanto, O. Simula, P. Vasara. *An Approach to Automated Interpretation of SOM*. Proceedings of the Workshop on Self-Organizing Map 2001(WSOM2001), Springer, pages 89-94, 2001

[Tor52] W. Torgerson. *Multidimensional scaling. I. Theroy and method*. Psychometrika 17:401-419, 1952

[Ult90] A. Ultsch and H. Siemon. *Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis*. In Proceedings of International Neural Network Conference (INNC'90), pages 305–308, Dordrecht, Netherlands. Kluwer, 1990

[Ves99] J. Vesanto, J. Himberg, E. Alhoniemi, Juha Parhankangas. *Self-Organizing Map in Matlab: the SOM Toolbox.* In Proceedings of the Matlab DSP Conference 1999, pages 35-40, Espoo, Finland, November 1999

[Ves00] J. Vesanto and E. Alhoniemi. *Clustering of the self-organizing map*. IEEE Transactions on Neural Networks, 11: 586-600, 2000

[Ves02] J. Vesanto. *Data Exploration Process Based on the Self-Organizing Map*, Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 115, Espoo 2002, 96 pages, Finnish Academies of Technology, 2002

[Vil97] T. Villmann, R. Der, M. Herrmann, and R. Martinetz. *Topology Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement*, IEEE Transactions on Neural Networks, 8(2): 256-266, 1997

[WBDI01] World Bank. *The 2001 World Development Indicators,* (1960-1999). Washington, D.C. (CD-ROM Version), http://www.worldbank.org, 2000