RESEARCH

Michael Christoph Thrun

Projection-Based Clustering through Self-Organization and Swarm Intelligence

Combining Cluster Analysis with the Visualization of High-Dimensional Data

OPEN

Springer Vieweg

Projection-Based Clustering through Self-Organization and Swarm Intelligence

Michael Christoph Thrun

Projection-Based Clustering through Self-Organization and Swarm Intelligence

Combining Cluster Analysis with the Visualization of High-Dimensional Data

OPEN

Springer Vieweg

Michael Christoph Thrun
Marburg, Germany

Philipps-Universität Marburg 2017, Hochschulkennziffer 1180

ISBN 978-3-658-20539-3 ISBN 978-3-658-20540-9 (eBook)
https://doi.org/10.1007/978-3-658-20540-9

Library of Congress Control Number: 2017963649

Springer Vieweg

Printed on acid-free paper

This Springer Vieweg imprint is published by Springer Nature
The registered company is Springer Fachmedien Wiesbaden GmbH
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Acknowledgments

My gratitude goes to my sister Monika Sikora. You have converted with great devotion my often complex phrases and contexts into intelligible words. I thank you for your intuitive understanding of language.
I would like to thank Prof. Dr. Alfred Ultsch for his demanding scientific guidance and continuing education. His leadership and preparatory work have provided me with the tools to meet the demands of the job. Your suggestions have given me the creativity that was often necessary to redirect my research when a solution was not obvious.
Without my student colleagues, Felix Pape and Florian Lerch, some of the ideas sprouted in this work would not have been feasible. I owe you, Felix, because of your selfless voluntary commitment to the realization of manufactured 3D printing models of the U-matrices. The fundamental code for visualizing the U-matrices in the form of a topographic map would not have been possible without your cooperation, for which I thank you, Florian.
The positive working environment, which allowed me to flourish, I largely owe to my colleague Catharina Lippmann. Your collegiality and constructive cooperation created the atmosphere that was continuously aiding my systematic research.

Table of contents

List of figures

List of tables

Zusammenfassung

Die vorliegende Arbeit befasst sich mit einem neuen Ansatz zur Clusteranalyse hochdimensionaler Daten. Die projektionsbasierte Clusteranalyse verbindet in zwei Dimensionen erhaltenen Strukturen mit zugrunde liegenden hochdimensionalen Strukturen.
Hierbei werden Cluster als natürlich definiert, wenn sie auf hochdimensionalen Daten beruhen, welche Diskontinuitäten aufweisen. Solche distanz- oder dichtebasierte Diskontinuitäten bezeichnen entweder kompakte oder verbundene Strukturen. Natürliche Cluster mit kompakten Strukturen werden hauptsächlich durch Inter- und Intra-Cluster-Distanzen definiert, während verbundene Strukturen auf dem Prinzip von Nachbarschaften zwischen Datenpunkten beruhen.
Mit Hilfe auf der Graphentheorie begründeten Grundprinzipien und den in dieser Arbeit durchgeführten Untersuchungen lässt sich schlussfolgern, dass zum Erreichen einer Visualisierung oder Clusteranalyse die Optimierung einer mathematischen Zielfunktion irreführende Ergebnisse bezüglich der Struktur liefern kann, wenn die zugrunde liegenden Strukturen der verwendeten hochdimensionalen Daten dieser Zielfunktion nicht entsprechen.
Diese Arbeit geht der Fragestellung nach, wie man einen korrekten Typ von Strukturen herausfinden kann, der Cluster in einem hochdimensionalen Datensatz ohne Vorannahmen definiert. Es wird dargelegt, dass Verfahren der Dimensionsreduktion helfen können, dieses Problem zu lösen.
Projektionsverfahren stellen einen gängigen Ansatz zur Dimensionalitätsreduktion hochdimensionaler Daten dar. Sie werden verwendet, um die Größe des Eingaberaumes zu reduzieren um dadurch eine Visualisierung der hochdimensionalen Daten zu ermöglichen. Durch die Beschränkung des Ausgaberaumes auf zwei Dimensionen zu einem Streudiagram (Projektion) repräsentieren niederdimensionale Ähnlichkeiten jedoch nicht notwendigerweise die Distanzen. Die Projektion kann zu einer irreführenden Interpretation der Strukturen führen.
Die Qualitätsmaße (QM) zur Bewertung der Projektion haben Schwierigkeiten Diskontinuitäten in hochdimensionalen Daten korrekt zu erfassen, weil sie unter Umständen auf falschen Annahmen über die zugrunde liegenden hochdimensionalen Strukturen basieren. Andernfalls könnte mittels einer QM eine globale Zielfunktion definiert werden. Es wäre damit immer möglich, eine strukturerhaltende Projektion durch Optimierung dieser Zielfunktion zu erhalten.
Das aus diesen drei Modulen bestehende Verfahren Databionicswarm (DBS) wird in dieser Arbeit vorgestellt. Das erste Modul des hier vorgeschlagenen Ansatzes besteht darin, hochdimensionale Distanzen in der zweidimensionalen Projektion durch eine dreidimensionale topographische Karte mit hypsometrischen Farben zu visualisieren. Die resultierende topographische Karte ist die Weiterentwicklung der „generalisierten U-matrix“.
Im zweiten Modul wird das neue Projektionsverfahren Pswarm vorgeschlagen. Pswarm nutzt die Konzepte der Schwarmintelligenz, Selbstorganisation, Symmetrieüberlegungen der Physik und das Nash-Gleichgewichtskonzept aus der Spieltheorie. Für Pswarm entfällt die Notwendigkeit einer globalen Zielfunktion. Dieses Projektionsverfahren erfordert, abgesehen von der Distanz, keine Eingabeparameter für die Projektion. Durch Selbstorganisation können Strukturen von hochdimensionalen Daten durch einen Prozess abgebildet werden, der als Emergenz bekannt ist. Die Erwartung hat sich bestätigt, dass ein Schwarm aus intelligenten

Agenten für die Visualisierung und Clusteranalyse verwendet werden kann. Pswarm wurde mit den üblichen Projektionsmethoden PCA, CCA, t-SNE, ESOM, NeRV und dem MDS-Technik-Sammon-Mapping verglichen. Hierbei wurde ein neues Qualitätsmaß (Delaunay Classification Error, DCE) eingesetzt. Der DCE ermöglicht durch die Verwendung vorgegebener Klassifikationen eine unvoreingenommene Beurteilung der Projektionsqualität für beide Arten von Strukturen. Die Ergebnisse zeigen, dass es mit Pswarm-Projektionen möglich ist Projektionen resultierend aus der Optimierung einer globalen Zielfunktion zu übertreffen.
Im dritten Modul werden die Ansätze früherer Arbeiten erweitert, indem kürzeste Wege zwischen geodätischen Abständen der abstrakten U-Matrix von projizierten Punkten für die Clusteranalyse verwendet werden.
DBS übertrifft die gängigen Methoden der Clusteranalyse (k-means, PAM, Single-Linkage, Spektralclustering, modellbasierte Clustering und Ward) hinsichtlich Stabilität und Plastizität auf einem künstlichen Benchmark-System von Datensätzen (FCPS). Im Gegensatz zu anderen üblichen Methoden der Clusteranalyse findet DBS keine Cluster, wenn keine natürlichen Cluster vorhanden sind. Die Anzahl der Cluster kann hierbei mit Hilfe einer Visualisierung abgeschätzt werden.
Die Anwendung von DBS auf drei hochdimensionale und multivariate Datensätze für den praktischen Gebrauch (Leukämie, Welt-Bruttoinlandsprodukt, Tetragonula-Bienen) reproduzierten bereits bekannte Erkenntnisse. In zwei aktuellen Anwendungen, Hydrologie und Schmerz-Gene findet DBS plausible und erklärbare Cluster.
Durch die Modularität lässt sich DBS zu einer projektionsbasierten Clusteranalyse verallgemeinern. Sollte Vorwissen gegeben sein, kann die Visualisierung durch die generalisierte U-Matrix und das DBS-Clustering auf jede Projektionsmethode für beide Strukturtypen (kompakt oder verbunden) angewendet werden. Alternativ können durch die verallgemeinerte U- Matrix-Visualisierung die Ergebnisse gängiger Clustermethoden durch die von Pswarm gefundenen Strukturen oder jede andere Projektionsmethode überprüft werden. Darüber hinaus können 3D-Drucke der visualisierten Strukturen von hochdimensionalen Datensätzen mit üblichen 3D-Drucktechniken hergestellt werden.

Abstract

This work introduces a new approach for cluster analysis defined as projection-based clustering. The projection based clustering combines structures preserved in two dimensions with underlying high-dimensional structures, if natural clusters exist in high-dimensional data. Clusters are defined as natural, if they are based on patterns in high-dimensional data characterized by discontinuity. Discontinuous patterns, which can either be based on distance or density, are described in this work as compact or connected structures. Natural clusters with compact structures are defined mainly by inter- versus intracluster distance, whereas the connected structures are based on the idea of neighborhoods present between data points.

With the use of basic principles founded on graph theory, this work demonstrated that the objective functions of clustering and visualization are based on the fundamental distinction between connected and compact structures. The derived conclusion is that in a case when the goal is to achieve a structure-preserving visualization or clustering, the optimization of a mathematical objective function could yield misleading results if the underlying structures of the high-dimensional data do not coincide with the objective function. The question that arises is how to recognize structures that defines clusters in a high-dimensional data set without prior knowledge. The argument here is that dimensionality reduction methods may help solve this problem.

Projections are common dimensionality reduction methods to visualize high-dimensional data in a two-dimensional space. However, when restricting the Output space into two dimensions resulting in a two dimensional scatter plot (projection) of the data, low dimensional similarities do not represent high dimensional distances coercively. This could lead to a misleading interpretation of the underlying structures. Further, it is argued here that the quality measures (QMs), which evaluate this projection, have difficulties to correctly grasp discontinuities in high-dimensional data; this is because they imply assumptions about the underlying high-dimensional structures. Otherwise, a global objective function could be defined using the best QM, and it would always be possible to obtain a structure-preserving projection or clustering by optimizing this objective function.

Therefore, the first module for a solution proposed here is to visualize high-dimensional distances in the projection through a three dimensional topographic map with hypsometric colors, which is a further development of the generalized U-matrix.

After an extensive review of application of artificial intelligence in data science, two interesting concepts are addressed here, called self-organization and swarm intelligence. The irreducible structures of high-dimensional data can emerge through self-organization in a phenomenon called emergence. If properly applied through the use of a swarm of intelligent agents, the data-driven approach presented in this work can outperform the optimization of a global objective function in the tasks of clustering and dimensionality reduction.

Here, the second module called Pswarm, is presented for projecting high-dimensional data. Pswarm exploits the concepts of swarm intelligence, self-organization, symmetry considerations in physics, and the Nash equilibrium concept from game theory. It eliminates the need for a global objective function and does not require any input parameters for projection besides a distance. The data-driven Pswarm was compared to the common projection methods PCA,

CCA, t-SNE, ESOM, NeRV and the MDS technique Sammon mapping. Using the new quality measure (Delaunay classification error) this work showed that the resulting two-dimensional projections of Pswarm are comparable to the state of the art projection methods like NeRV and ESOM. By using prior classifications, the Delaunay classification error allows for an unbiased evaluation of projection quality for both types of structures.

For the third module, the author expands the idea of previous works by using shortest paths between geodesic distances of the abstract U-matrix of projected points in the case of cluster analysis. The whole method is called Databionic swarm (DBS) and it outperforms the common clustering methods (k-means, PAM, single-linkage, spectral clustering, model based clustering and Ward) in terms of stability and plasticity on an artificial benchmark system of data sets (FCPS). Contrary to other common clustering methods, the DBS finds no clusters if no natural clusters exist. The number of clusters can be estimated with the help of the topographic map.

On three different high dimensional and multivariate data sets (types of leukemia, world gross domestic product, Tetragonula bees), the already known insights can be reproduced. In two real world applications of hydrology and pain genes, the DBS retrieves meaningful clusters, which was confirmed by domain experts.

Through the modularization, DBS can be generalized to projection to projection-based clustering. The visualization by the generalized U-matrix and the DBS clustering can be applied to every projection method for both types of structures. Through the use of the topographic map, results of common clustering methods can be verified by the structures found by Pswarm or any other projection method. Additionally, 3D prints of the visualized structures of high dimensional data sets can be manufactured with common 3D printing techniques

1 Introduction

We live in a time when information is cheaply available and saved as data nearly everywhere. The amount of generated data is growing exponentially. By the end of the year 2016 alone, 9000 exabytes of data will have been generated, equal to 9 trillion gigabytes or the capacity of 360 billion Blu-ray Discs [Schiele, 2016]. The goal of the interdisciplinary field of data science is to extract knowledge from these data with the help of statistics, machine learning or data mining. Unlike in physics, a data scientist hardly ever starts with a hypothesis; he also is not interested in the source of the data or how they were collected. The data must be mined to gain knowledge through the identification of consistent patterns, and this is usually a very trying task.

Among the various available methods of analyzing data, the focal point of this work is cluster analysis. In contrast to common approaches, the goal here is not merely to group similar information but also to explain why the grouping of information in a certain context is valid, non-trivial and useful. Only then will the clustering of data be helpful to a domain expert. Cluster analysis "is a discipline on the intersection of different fields and can be viewed from different angles, which may be sometimes confusing because different perspectives may contradict each other" [Mirkin, 2005, p. 33]. From the statistical perspective, some assumption regarding the underlying model is required, and data clusters are viewed as probability distributions whose properties can be estimated from the data themselves [Mirkin, 2005, pp. 33-34]. "A trouble with this approach is that in most cases clustering is applied to phenomena of which nothing is known" [Mirkin, 2005, p. 34]. Here, cluster analysis is regarded as the process of generating a classification based on empirical data in a situation in which clear theoretical concepts and definitions are absent and the patterns and laws governing the situation are unknown (see [Mirkin, 2005, p. 36]). The concept of every application (available as open-source code in the R language [R Development Core Team, 2008]) used throughout this thesis is based on this idea.

The goal of this work is to provide an open-source framework for cluster analysis that is founded on a swarm-based projection method and uses a human-understandable visualization approach based on a topographic map of high-dimensional data structures, with the option of 3D printing (see [Thrun et al., 2016a]). This framework should be sufficiently stable while remaining adaptive and exhibiting sufficient plasticity to permit the creation of clusters of various shapes. It should include only a very few non-sensitive parameters that can be visually deduced by a non-professional data miner without any need to understand the theory behind them.

To achieve this goal, expertise on various topics from various areas of research will be required. It is the author's experience that experts in different fields rarely share or exchange practical approaches, and almost nobody is interested in providing and willing to provide easily available and human-understandable solutions to domain experts.

Here, the main hope is to be able to provide reproducible cluster analysis solutions for non-professional data miners and to deliver human-understandable concepts of high-dimensional data structures that are simultaneously able to be processed by machines. In the context of the Databionic swarm (DBS) approach, the author attempts to build, use and explain connections

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_1

among various fields of research; to be precise, the author will illustrate connections between cluster analysis [Hennig et al., 2015; Jain/Dubes, 1988], the imitation of collective behavior [Beni/Wang, 1993; Bonabeau et al., 1999; Reynolds, 1987], the visualization of information [Venna et al., 2010] and its evaluation, machine learning applications [Herrmann/Ultsch, 2008c], game theory [Nash, 1951], symmetry considerations in physics [Feynman et al., 2007, pp. 147-153, 745] and emergence [Ultsch, 2007]. Undoubtedly, making connections between different schools of thought sometimes requires simplifications. For example, with regard to the collective behavior of bees, the fact that bees have a queen who influences their behavior remains unaddressed in this work. Such simplifications are necessary for analytical modeling and applications of cluster analysis.

Chapter 2 addresses most of the necessary definitions and lays the groundwork for all of the mathematical notation used throughout the thesis. The literature reviewed in chapter 3 shows how common clustering methods tend to implicitly assume the patterns or structures sought in data. The reviewed clustering methods are grouped based on their definitions of generalized neighborhoods.

Chapter 4 introduces and classifies common methods of projecting high-dimensional data into two dimensions. Such projections are necessary to cope with the pitfalls of higher dimensions (see, e.g., [Bouveyron/Brunet-Saumard, 2014, pp. 55-57; Verleysen et al., 2003]). Two- or three-dimensional projections will always result in errors; however, gaining a spatial understanding of more than three dimensions is typically an excessively complex task for humans.

Chapter 5 presents examples to depict the typical errors encountered and describes efforts to manage these errors by means of the U-matrix visualization approach [Ultsch, 2003a]. By contrast, chapter 6 demonstrates a more stringent mathematical approach based on quality measures (QMs) presented in the literature. The evaluation of 19 QMs yields a grouping of the QMs based on their implied characterization of structures of high-dimensional data using the definition of neighborhoods introduced in this thesis. Consequently, it is not possible to generalize any of the QMs. If it were possible, the corresponding optimization approaches would not imply any prior assumptions about the structures of high-dimensional data and, consequently, would outperform any other projection methods.

Chapter 7 discusses a nature-inspired and behavior-based system of data science with the goal of using emergence, instead of the optimization of an objective function, for data visualization and clustering.

Building on the insights gained in chapter 7, chapter 8 introduces the DBS concept. Because it relies on the self-organization of data and emergence, DBS does not imply any particular structure that is sought in data. In the context of the projection, visualization and clustering of artificial or high-dimensional data, chapters 10-12 compare DBS with various common methods and apply the DBS framework both to reproduce known insights and to gain new knowledge about various types of data, e.g., multivariate time series or genetic data.

Readers may skip certain chapters depending on their interests. However, the contents of some chapters are based on insights from previous chapters, as indicated by arrows in Figure 1.1, which outlines the organization of this work. Please note, that due to technical limitations the figures and equations are numbered chapter wise.

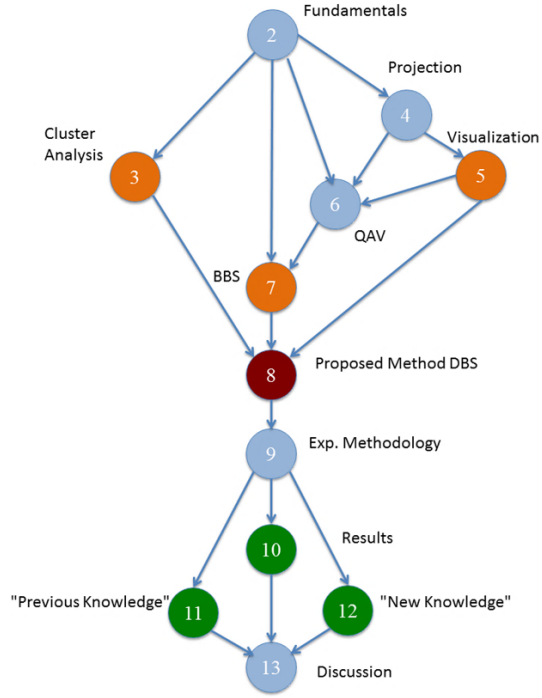

Figure 1.1: Dependency graph of the chapters. BBS: behavior based systems; QAV: Quality Assessments of Visualizations; DBS: Databionic swarm. The underlying concept of DBS is based on insights from chapters 3, 5 and 7 (orange). The evaluation of DBS is performed in three steps (green): general validation in chapter 10, the reproduction of known knowledge in chapter 11, and the generation of new knowledge, as validated by domain experts, in chapter 12.

2 Fundamentals

The first section of this chapter familiarizes the reader with the definitions of the basic notation and terminology used in this thesis. Concepts of graph theory are introduced in the next section. They give rise to a new concept of neighborhoods, which is utilized in several chapters. The last section explains a possible approach to knowledge discovery, which is applied in chapters 11 and 12.

2.1 Basic Definitions

Hilbert space
Let $\mathcal{H}$ be a vector space above a field K with the following properties for every pair of elements $(x, y, z) \in \mathcal{H}$ and $\alpha \in$ K:

1.) $\langle .,. \rangle_{\mathcal{H}} : \mathcal{H}$ x $\mathcal{H} \rightarrow$ K is a non-degenerate symmetric bilinear form:
   a. $\forall\, x \in \mathcal{H} : \langle x, x \rangle_{\mathcal{H}} \geq 0$
   b. $\langle x, y \rangle_{\mathcal{H}} = 0, \forall\, y \in \mathcal{H} \Rightarrow$ x=0
   c. $\langle x, y \rangle_{\mathcal{H}} = \overline{\langle y, x \rangle_{\mathcal{H}}}$ if $\mathrm{K} = \mathbb{C}$, and $\langle x, y \rangle_{\mathcal{H}} = \langle y, x \rangle_{\mathcal{H}}$ if $\mathrm{K} = \mathbb{R}$
   d. $\langle \alpha x, y \rangle_{\mathcal{H}} = \alpha \langle y, x \rangle_{\mathcal{H}}$
   e. $\langle x + y, z \rangle_{\mathcal{H}} = \langle x, z \rangle_{\mathcal{H}} + \langle y, z \rangle_{\mathcal{H}}$
2.) Each Cauchy sequence $\{x_i\}_{i \in \mathbb{N}}$ in $\mathcal{H}$ converges to an element of $\mathcal{H}$, i.e., the space is complete with respect to the norm induced by $\langle .,. \rangle_{\mathcal{H}}$.

Thus, $\mathcal{H}$ is a Hilbert space (for further details, see [Bronstein et al., 2005, pp. 635-636; Nolting, 2001, p. 22]).

Bra-ket notation
Bra-ket notation $\langle . | . \rangle$ is used in physics to describe functions or vectors in a Hilbert space when the coordinate system of the vectors is irrelevant. The left part is called the bra ($\langle . |$), and the right part is the ket ($| . \rangle$). This notation is used to describe physical states (it is also called Dirac notation, as described in [Dirac, 1981, pp. 15-22]; for a formal introduction, see [Nolting, 2001, pp. 147-148]).

Operator
An operator $\hat{A}$ is an unambiguous mapping of each element $|\alpha\rangle$ of the subset $D_\alpha \subseteq \mathcal{H}$ to an element $|\beta\rangle \in W_A \subseteq \mathcal{H}$ such that $|\beta\rangle = \hat{A}\,|\alpha\rangle = |\hat{A}\,\alpha\rangle$, where D_α is the definition range of $\hat{A}$ and the set of all $|\beta\rangle$ is the domain of $\hat{A}$, as defined in [Nolting, 2001, p. 153]; see also [Bronstein et al., 2005, pp. 49,639-640]. An "operator is considered to be completely defined when a result of its application to every ket vector [$|\alpha\rangle$] is given" [Dirac, 1981, p. 23].

Observation
An observation *f* is a set of measured values for the properties of a phenomenon. It is described in the bra-ket notation as the change from one physical state $\langle y|$ to another physical state $|x\rangle$ that results from the measurement of the operator $\hat{f}$, as denoted by $f = \langle y|\hat{f}|x\rangle$ (see [Feynman et al., 2006, pp. 145, 147]). Such an observation *f* is a measurement of a physical process.

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_2

Feature

Each individually measurable property *r* of a phenomenon being observed can be mapped to an operator $\hat{r}$ that can be applied to a physical state $|x\rangle$ [Stöcker et al., 2007, p. 744]. Such an individually measurable property is called a **feature**, **attribute** or **observable**. Here, an approximately continuous distribution of values in the vector space $\mathbb{R}^d$ is additionally assumed for a **variable** (see the definition of the **distribution of a variable**).

Data

A batch of data is defined as a matrix $\langle i|\hat{A}\,|j\rangle = A_{ij}$, in which **facts**[1] about a physical state are summarized based on observations of the form $\langle y|\hat{A}\,|x\rangle = \sum_{ij}\langle y|i\rangle\langle i|\hat{A}\,|j\rangle\langle j|x\rangle$ of a phenomenon in a Hilbert space, where $\langle i|$, $\langle j|$, $|i\rangle$ and $|j\rangle$ are the basic states relevant to the phenomenon (for further discussion, see [Feynman et al., 2006, pp. 147-150]).

Distribution of a variable

A formal distribution df is defined as the probability density of a feature r: $df(r) = \lim_{\Delta r \to 0} \frac{\langle x_{r,\Delta r}|\, x\rangle}{\sqrt{(\Delta r)}}$ [Nolting, 2001, p. 150]. If the feature *r* is continuous, then it is called a **variable** $z \in \mathbb{R}^d$, and *df* is called its probability density function (**pdf**) (see [Goodfellow et al., 2016, p. 58]). Here, when it describes how the relative probability of a variable z takes on a given value, such a distribution is a pdf that is assumed to be normalized as follows [Walck, 2007, p. 15]: $\int_{-\infty}^{\infty} pdf(z)dz = 1$.

"Statisticians often use the distribution function or as physicists more often call it the cumulative function which is defined as $cdf(z) = \int_{-\infty}^{x} pdf(z)dz$" [Walck, 2007, p. 15].

If not elaborated further, here, the distribution of a variable *z* is regarded as an approximation of its pdf; for further details, see, for example, [Bock, 1974, p. 250; G. Ritter, 2014, p. 275 ff], and for types of pdfs, see [Walck, 2007].

Dirac delta function

The Dirac delta function δ is a function with the following properties [Jackson, 1999, p. 31]:

1.) $\delta(z-a) = 0$ iff $z \neq a$

2.) $\int \delta(z-a) = \begin{cases} 1, & \text{if } z = a \text{ lies in the integration area under the curve} \\ 0, & \text{otherwise} \end{cases}$

Density of data

Let *dn* be the number of observations in an **elementary volume** (see [Bronstein et al., 2005, p. 491]) $d^{\tilde{d}}v = dv_1 * dv_2 * .. dv_{\tilde{d}} = d\vec{v}$ of the Hilbert space $\mathbb{R}^{\tilde{d}}$ (henceforth, $\mathbb{R}^d$); then, the density of the data is defined as $\rho(\vec{v}\,) = \frac{dn}{d\vec{v}}$, where $\rho\colon \mathbb{R}^d \to \mathbb{R}$ is the density field function. Here, ρ is subject to the condition that N is the number of data points defined by $N = \int_{\mathbb{R}^d} \rho(\vec{v})\mathrm{d}\vec{v} = \int_{\mathbb{R}^d} \sum_{i=1}^{N} \delta(\vec{v} - \vec{v}_i)\mathrm{d}\vec{v}$, in analogy to [Jackson, 1999, p. 33], where δ is the Dirac delta function and $\rho(\vec{v}) = \sum_{i=1}^{N} q_i \delta(\vec{v} - \vec{v}_i)$ is the charge density of point charges. Then, the **homogeneity** of the data is defined as $N = \int_{\mathbb{R}^d} \rho(\vec{v})\mathrm{d}\vec{v} = \int_{\mathbb{R}^d} \rho_0 \mathrm{d}\vec{v} = \rho_0 \int_{\mathbb{R}^d} \mathrm{d}\vec{v}$, where $\rho_0 = \text{const.}$

[1] See [Fayyad et al., 1996, p. 6].

Pattern

A "[p]attern is an expression E in a language L describing facts [F] in a subset F_E of F. E is called a pattern if it is simpler than the enumeration of all facts in F_E" [Fayyad et al., 1996, p. 7]. Here, the expression E is "simpler" if it describes a group of similar (see the definitions of **metric space** and **distance** below) or homogeneous observations.

In graph theory, a pattern may be described by a neighborhood H (see the graph theory section for details). If the observations are not directly comprehensible, such a pattern is called a *hidden pattern*.

Discontinuity in data

A set of data can exhibit discontinuity if

$$\int_{\mathbb{R}^d} \rho(\vec{v}) \mathrm{d}\vec{v} \neq \rho_0 \int_{\mathbb{R}^d} \mathrm{d}\vec{v},$$

which means that the density of data ρ depends on its location $\vec{v}$ in the Hilbert space $\mathbb{R}^d$; Discontinuities can occur when interruptions or distortions exist in the homogeneity of the data, or in the continuity of the distribution of the data, in $\mathbb{R}^d$. Thus, there are elementary volumes $\mathrm{d}\vec{v}$ with high density and elementary volumes $\mathrm{d}\vec{v}$ with low density or even empty elementary volumes. In the one-dimensional case, such a discontinuity can be mathematically defined as an essential or jump discontinuity. In two or three dimensions, a discontinuity may manifest as a spatial separation (see, e.g., Figure 2.1 or chapter 5 and 9, the Hepta data set).

In a higher-dimensional case, a discontinuity represents a change in the characteristics of facts, resulting in multiple patterns (see, for example, the leukemia data set, chapter 3, Figure 3.7 and chapter 9).

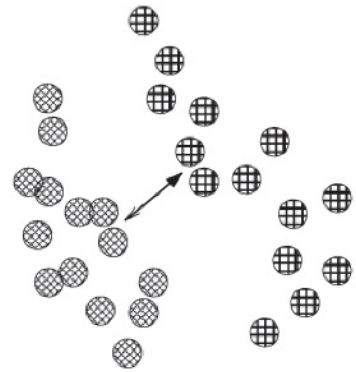

Figure 2.1: Spatial separation of data, after [Handl et al., 2005].

Metric space and distance

Let a metric space be represented by an ordered pair *(M, d)*, where *M* is an arbitrary set and *d* is a metric on *M*, i.e., a function

$$\mathrm{d} : \mathrm{M} \times \mathrm{M} \rightarrow \mathbb{R}$$

such that for any $l, j, m \in M$,

$$d(l,j) = d(j,l)$$
$$d(l,j) \geq 0$$
$$d(l,j) = 0, \text{iff } l = j$$

and the triangle inequality is satisfied as follows:

$$d(l,j) + d(j,m) \geq d(l,m)$$

Then, the metric *d* is also called a **distance** (see [Bronstein et al., 2005, pp. 624-625]). By contrast, for a **dissimilarity**, denoted by $\hat{d}$, the triangle inequality may not apply ([Bock, 1974, pp. 25-26]. The distance between two **similar** points $l, j \in M$ is small, whereas that between two **dissimilar** points $l, j \in M$ is large. Transformations exist between a dissimilarity $\hat{d}$ and a distance *d* (e.g., [Bock, 1974, pp. 77-79]).

If the distance is defined in an output space O, it is denoted by *d(l, j)*, whereas a distance defined in an input space *I* is denoted by *D(l, j).* An example of a metric space is a Hilbert space that is a real-numbered vector space $\mathbb{R}^{\mathrm{d}}$ of *d* dimensions. If the distances in a space are defined as Euclidean distances, then the corresponding space is called a Euclidean space.

Data set

A data set consists of a finite set of observations $f \in \mathrm{F} \subset \mathcal{H}^{\tilde{\mathrm{d}}}$ of $\tilde{d}$ observed features.

In this work, observations f are assumed to be vectors *l* in a metric space *M*, and features are assumed to be variables, if not stated otherwise.

Input space

An input space $I \subset \mathbb{R}^d$ is the d-dimensional space consisting of $d \leq \tilde{\mathrm{d}}$ variables in a data set that have been selected for a given task and contains *n* data points: $I = \{l_1, \ldots, l_n, n \in \mathbb{N}\}$. The properties of an input space are as follows (see [Lee/Verleysen, 2007, p. 243]):

I. The input space is considered to be *high dimensional* if it contains more than five variables, which makes direct visualization very difficult.
II. If the number of data points is greater than 2000, then the input space is considered to be *large*[2].
III. If the number of data points is fewer than 200, then the input space is considered to be *small*.

Data point

A data point $l \in I$ is a numeric vector consisting of one observation for each of the *d* variables in the input space, where a vector is an array of numbers arranged in a specific order defined with respect to the d variables.

[2] Note that, in general, the number of data points has greatly increased over time [Goodfellow et al., 2016, p. 21 , Fig. 1.8] and therefore the precise number may change with time

Object
When the data of interest are a set of facts F consisting of numerical, ordinal or nominal scaled entries, each fact $f \in F$, such that f $\notin \mathbb{R}^{\mathrm{d}}$, is called an object or **case**.
An object can be regarded as a generalization of a data point. If an object can be interpreted (has a meaning within itself), then it contains **information** ([Ultsch, 2016c]; see also [Ultsch, 1994, p. 2]).

Output space
An output space $O \subset \mathbb{R}^m$ is the m-dimensional space such that $m<d$ in which, for each point $j \in O$, a mapping to a data point l of the input space I$\subset \mathbb{R}^d$ exists.

Machine learning
The field of machine learning concerns computer programs that can imitate learning behavior [Natarajan, 2014] (see also [Goodfellow et al., 2016, p. 99]). Machine learning comes in two general forms[3] (see [Murphy, 2012, p. 2]). *Unsupervised learning* refers to the task of finding patterns in unlabeled data. Since the data are unlabeled, no reward function exists that can be used to evaluate potential results. If the data set is labeled, then *supervised learning* is possible. A typical supervised learning task is classification or regression. A typical unsupervised learning task is cluster analysis.

Label
A label is a tag $g \in \{1, \ldots, k\} \subset \mathbb{N}$ attached to an object $f \in F$ that identifies the object via a mapping $f: \{1, \ldots, k\} \to F$. The labels of such a set of objects range from *one* to *k* [Hennig et al., 2015, p. 2], where k is the number of groups of objects. Here, it is assumed that a label exists for every object.

Classification
A classification $C = \{G_1, G_2, \ldots\}$ is a system of subsets [Bock, 1974, p. 22] such that $C \subset \mathcal{H}^{\tilde{\mathrm{d}}}$. A subset $G_i = \{l_1, \ldots l_k\} i \in \mathbb{N}$, , is a set of k observations. In an exclusive classification, the subsets are disjunct, denoted by $G_1 \cap G_2 = \emptyset$; in a non-exclusive classification, elements that overlap between two subsets may exist, denoted by $G_j \cap G_k \neq \emptyset$. However, overlapping classification is not considered here (for various types of classification, see Figure 2.2 or [Hennig et al., 2015, p. 45]). Supervised and unsupervised classifications are defined as in the context of machine learning.

[3] Reinforcement learning is not considered in this context; semi-supervised learning (e.g. active learning) uses labeled data as well as unlabeled data.

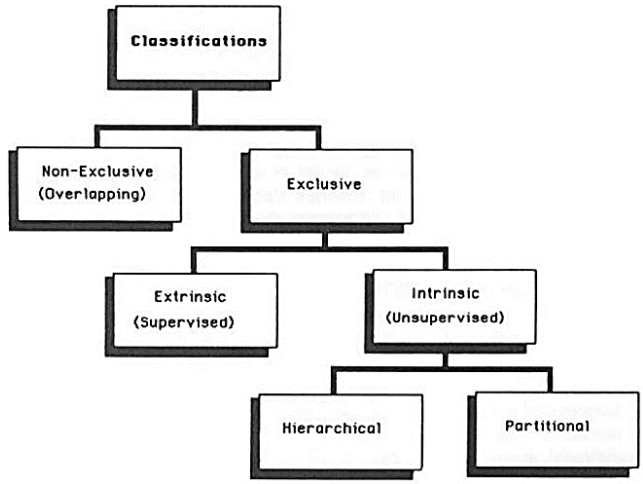

Figure 2.2: Tree of classification types, after [Jain/Dubes, 1988, p. 56]. This work concentrates on unsupervised classification (see unsupervised machine learning).

Classifier

A classifier is an algorithm that constructs a function $Cls: F \to \{1, \ldots, k\} \subset \mathbb{N}$ that maps objects $f \in F$ to class labels $g_i \in \mathbb{N}$.

In terms of understandability, a distinction can be drawn between symbolic and sub-symbolic classifiers [Ultsch/Korus, 1993]. Symbolic classifiers are able to acquire knowledge (for a detailed description, see the last section of this chapter). By contrast, sub-symbolic classifiers (e.g., KNN classifiers) are only able to integrate knowledge [Ultsch, 1994], because a characteristic property of a sub-symbolic representation of data is that a single object alone does not contain information (see [Ultsch, 1994, p. 2]).

Projected point

A projected point $j(x_1, .., x_m) = \vec{j}$ is a vector of m scalars x_i in the output space $O \subset \mathbb{R}^m$, where a vector is an array of numbers arranged in a specific order such that each individual number can be identified by its index.

Projection

Let $j \in I$ denote data points in the input space $I \subset \mathbb{R}^d$, and let $l \in O$ denote projected points in the output space $O \subset \mathbb{R}^m$. Then, a mapping $\mathrm{proj}: I \to O,\ j \mapsto l$ is called a projection iff $m = const \wedge m \ll d$.

Note that unlike for a projection method, for a manifold learning method, the dimensionality of the output space m depends on the data set (see, e.g., [Lee/Verleysen, 2007, pp. 14-15]).

2.2 Concepts of Graph Theory Applied to Patterns

This section uses graph theory to describe patterns found in data.

Graph

"A graph [Γ] is a pair [$\Gamma = (V, E)$] consisting of a finite set $V \neq \emptyset$ and a set E of two-element subsets of V. The elements of V are called vertices. An element e = (a, b) of E is called an edge with end vertices a and b. […] [In such a case,] *a* and *b* are adjacent or neighbors of each other" [Jungnickel, 2013, p. 2].

A graph Γ is called undirected if, for every edge $e(a, b)$ in E, the edge $e(b, a)$ is also in E. A graph is called a weighted graph if a number (weight) is assigned to each edge.

Directed graph

A "directed graph or, for short, a *digraph* is a pair $\Gamma = (V, E)$ consisting of a finite set *V* and a set *E* of ordered pairs *(a, b),* where $a \neq b$ are elements of V" [Jungnickel, 2013, pp. 25-26].

Direct adjacency

Let Γ be a graph, and let j be a point in a metric space M; then,

$$\mathcal{H}(j, \Gamma, M) = \{l \in M \mid v_l \in V \wedge \exists\, e(v_l, v_j) \in E\}$$

is the set of points that are directly adjacent to j. The direct adjacency is defined by the specified graph.

Adjacency matrix

A digraph Γ with a vertex set $\{1, \ldots, n\}$ is specified by an $n \times n$ matrix $A = (a_{ij})$, where $a_{ij} = 1$ if and only if (i, j) is an edge of Γ, and $a_{ij} = 0$ otherwise. *A* is called the adjacency matrix of Γ [Jungnickel, 2013, p. 40].

Path

Let $(e_1, \ldots, e_n)$ be a sequence of edges in a graph Γ. If there exist vertices $v_0, \ldots, v_n$ such that $e_i = v_{i-1}v_i$ for $i = 1, \ldots, n$, then the sequence is called a walk; if $v_0 = v_n$, one speaks of a closed walk (Figure 2.3). A walk for which the e_i are distinct is called a trail (Figure 2.3), and a closed walk with distinct edges is a closed trail. If, in addition, the v_j are distinct, then the trail is a path [Jungnickel, 2013, p. 5].

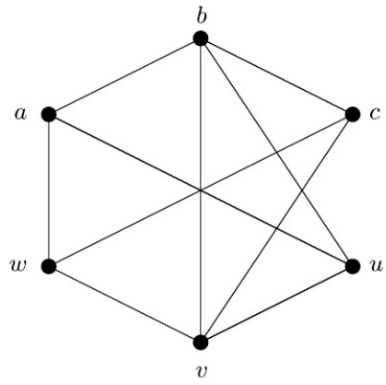

Figure 2.3: Examples of trails, walks and paths [Jungnickel, 2013, p. 6 Fig. 1.5]: (a, b, c, v, b, c) is a walk but not a trail, and (a, b, c, v, b, u) is a trail but not a path [Jungnickel, 2013, p. 5].

Connected Graph

Two vertices *a* and *b* of a graph *Γ* are called connected vertices if a walk exists with start vertex *a* and end vertex *b*. If all pairs of vertices of *Γ* are connected, then *Γ* itself is called a connected graph. For any vertex a, we consider *a* to be a trivial walk of length 0, such that any vertex is connected with itself. Thus, connectedness is an equivalence relation on the vertex set of *Γ*. The equivalence classes of this relation are called the connected components of *Γ*. Thus, *Γ* is connected if and only if its vertex set *V* is its unique connected component [Jungnickel, 2013, p. 6].

Lattice

A connected graph Γ with a particular well-defined two-dimensional tiling (tessellation) is defined as a lattice. A nxm lattice has n vertices on the x-axis and m vertices on the y- axis. If the tiling is rectangular (every vertex has exactly four perpendicular edges) it will be called a **lattice** (tiling) in this work, if the tiling is hexagonal (every vertex has exactly three edges) this will be called a **grid** (tiling) in this work.

Shortest path

For a connected graph Γ, there exists a distance *D(a, b)* between two vertices a and b that can be defined as the shortest path between these vertices [Jungnickel, 2013, pp. 65-66] as follows: For each path $P = (e_1, \dots, e_n)$, let the length of P be $p(P) := p(e_1) + \cdots + p(e_n)$; then, the distance between two vertices a and b in *(Γ, p)* is defined by

$$G(a, b, \Gamma) = \begin{cases} \infty, & \text{if } b \text{ is not accessible from } a \\ \min\{p(P): P \text{ is a path from } a \text{ to } b \text{ in } \Gamma\}, & \text{otherwise} \end{cases}$$

Let the vertices be denoted by points $l, j \in M$ in the metric space *M*; then, G(l, j, *Γ*) is the notation if the points l and j lie in the input space *I*, and $g(l, j, \Gamma)$ is the notation if they lie in the output space *O*.

Note that $d(a, a) = 0$ always holds because an empty sum is considered to have a value of 0, as usual. If no explicit length function is given, then the shortest paths and distances in a graph are defined using a length function that assigns a length of $p(e) = 1$ to each edge *e* [Jungnickel, 2013, p. 66]. An algorithm for calculating the shortest paths in a graph is described in [Jungnickel, 2013, pp. 83-87]. The authors Lee and Verleyson have claimed that graph distances outperform the traditional Euclidean metric in terms of dimensionality reduction [Lee/Verleysen, 2007, p. 227].

Acyclic graph

Let $(M, \preccurlyeq)$ be a partially ordered set (a poset, for short), which consists of the set *M* together with a reflexive, antisymmetric and transitive relation $\preccurlyeq$, and let *M* correspond to a digraph *Γ* with the vertex set *M* and with edges defined by pairs *(a, b)* such that $a \prec b$; then, because of the transitive property, *Γ* is acyclic [Jungnickel, 2013, p. 49].

Tree

A tree is a graph *Γ* that satisfies the following three conditions [Jungnickel, 2013, pp. 7-8]:

I. *Γ* is connected.
II. *Γ* is acyclic.
III. *Γ* contains *n-1* edges and *n* vertices.

The vertices in a tree are often called nodes. If (a, b) is an edge in a tree, then a is called the parent of b, and b is a child of a. If a path exists from a to b ($a \neq b$), then a is a proper ancestor of b and b is a proper descendant of a [Safavian/ Landgrebe, 1990, p. 2]. If a node has no descendant, it is called a leaf; if a node has no ancestor, it is called a root.

Directed acyclic graph (DAG)
A DAG is a directed tree (see above) that contains no cycles and one vertex, defined as the root, into which no edges enter. There is a unique path from the root to every vertex [Safavian/Landgrebe, 1990, p. 3]. Every vertex has a descendant called a child, except for the leaf vertices, which do not.

Decision tree
Let G_i be a subset of a classification $C = \{G_1, \dots, G_i, \dots\} \subseteq \mathcal{H}^{\tilde{\mathrm{d}}}$; then, a decision tree is a tree with the following properties:

I. Each node that is not a leaf is mapped to a feature $f \in F \subset \mathcal{H}^{\tilde{\mathrm{d}}}$.
II. Every edge (a, b), where a is the parent and b is the child, is mapped to a condition that matches the feature mapped to the parent a (see I.).
III. Every leaf is mapped to a subset G_i.

Decision tree learning
Decision tree learning refers to a type of supervised machine learning in which decision trees are used (see [Safavian/Landgrebe, 1990]).

Binary tree
A binary tree is an ordered tree such that [Safavian/Landgrebe, 1990, p. 3] (see also the definition of a DAG)

I. each child of a vertex is designated as either a left child or as a right child, and
II. no vertex has more than one left child nor more than one right child.

Lemma 1
Let $\Gamma = (V, E)$ be a connected graph with a positive length function p. Then, *(V, D)* is a finite metric space, where the distance function is defined as $D = G(a, b)$ [Jungnickel, 2013, p. 68].

Proposition 1
Any finite metric space can be represented by a pair (Γ, p) (network) with a positive length function *p* [Jungnickel, 2013, p. 68].

Ultrametric space
Note that a metric space can be represented by a tree if and only if the following condition holds for any four vertices *x*, *y*, *z*, and *t* of the given metric space [Jungnickel, 2013, p. 69]:

$$d(x, y) + d(z, t) \leq max(d(x, z) + d(y, t), d(x, t) + d(y, z))$$

Changing the triangle inequality to this condition implies an ultrametric space.

2.2.1 Patterns Defined as a Generalization of Neighbourhoods

Here, it is argued that by using shortest paths and direct adjacency, the patterns that exist in data can be generalized to neighborhoods H of an extent *k*.

Let k ∈ ℕ, *k>0*, let Γ be a connected graph, let *j* be a point in a metric space M, and let $G(j, l, \Gamma)$ be the shortest path between $j \in M$ and an arbitrary point $l \in M$; then (1),

$$H_j(k, \Gamma, M) = \{l \in M | \, G(l, j, \Gamma) \leq k\} \qquad (1)$$

is the neighborhood set of the point *j* and *k* the neighborhood extent. The neighborhood *H* can define a pattern in the input space[4].

The easiest example is a neighborhood defined by distances in a Euclidean graph. In the context of graph theory, a Euclidean graph is an undirected weighted graph of the highest order with respect to all other graphs discussed here, because every vertex is connected to every other vertex. Note that the weights of the vertices in a Euclidean graph need not necessarily be defined by the Euclidean metric. Another representation of a neighborhood H is a Delaunay graph $\mathcal{D}(V, E)$, which is a subgraph of a Euclidean graph. A Delaunay graph $\mathcal{D}(V, E)$ is based on Voronoi cells [Toussaint, 1980]. Each cell is assigned to one data point, and the size of a cell is characterized in terms of the nearest data points surrounding the point assigned to that cell. Within the borders of one Voronoi cell, there is no position that is nearer to any outer data point than to the data point within the cell. Thus, a neighborhood of data points is defined in terms of direct links between borders of Voronoi cells that induce an edge E in the corresponding Delaunay graph [Delaunay, 1934]. In short, a Delaunay graph represents a graph for a neighborhood $H(1, \mathcal{D}, M)$. A neighborhood H can also be represented by a Gabriel graph $G(V, E)$ [Gabriel/Sokal, 1969], which is a subgraph of a Delaunay graph $\mathcal{D}(V, E)$ in which two points are connected if the line segment between the two points is the diameter of a closed disc that contains no other points within it (empty ball condition). A Gabriel graph represents a graph for a neighborhood $H(1, G, M)$. Another case that is often considered is that of a neighborhood H_j*(knn, K, M),* where the number of nearest neighbors of a point *j* is defined by the number of vertices connected to this point in the *K*-nearest-neighbor graph (KNN graph), e.g., [Brito et al., 1997]. Here, we will use the shorter notation *H(knn, M).*

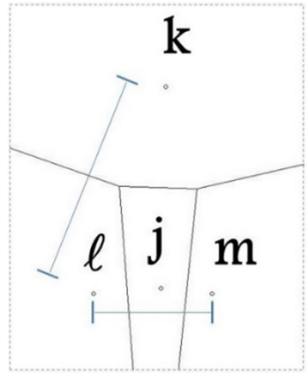

Figure 2.4: Four points and their Voronoi cells: *D(l, k)>D(l, m)* illustrate the different types of neighborhoods: unidirectional versus direction-based.

[4] Such neighborhoods *H* will prove useful for various evaluation steps, as summarized in Fig. 2.5.

Neighborhoods of points can be divided into two types, namely, *unidirectional* and *direction-based* neighborhoods. Consider the four points shown in Figure 2.4. The points *l*, *k*, *j*, and *l* are in the same neighborhood $H_l(1, \mathcal{D}, M)$ in the corresponding Delaunay graph, but the points *l* and *m* are never neighbors in this graph, even if the distance *D(l, m)* is smaller than *D(l, k)*. Thus, in this neighborhood definition, the direction information is more important than the real arrangement of the points in space as characterized by the distances *D*.
However, if a neighborhood is defined in terms of a KNN graph, then the points *l* and *m* could be in the same neighborhood $H_l(knn, K, M)$, and the points *l* and *k* could be in different neighborhoods, depending on the value of knn and on the ranking of the distances between these points. Therefore, this type of neighborhood is called unidirectional. In other words, it can be said that the points *l*, *j*, and *m* are more *dense* with respect to each other than they are with respect to *k*. Thus, unidirectional neighborhoods defined in terms of KNN graphs or unit disk graphs [Clark et al., 1990] can be used to define neighborhoods based on density.

2.3 Overview of Knowledge Discovery

> *"The term knowledge discovery in databases [...] was coined in 1989 to refer to the general process of finding knowledge in data and to emphasize the 'high-level' application of particular data mining methods" [Fayyad et al., 1996, p. 3].*

In 1996, Fayyad et al. used this term in his introduction to "From Data Mining to Knowledge Discovery" as follows:

> *"Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" [Fayyad et al., 1996, p. 6].*

Dropping the suffix *in databases*, the term *knowledge discovery* was extensively discussed in [Mörchen, 2006, pp. 6-7]. According to the definition used in that work, *knowledge discovery* is "data mining with the goal of finding knowledge, i.e., novel useful, interesting, understandable, and automatically interpretable patterns" [Mörchen, 2006, p. 7]. The definition of *data mining* as given in [Mörchen, 2006, p. 7] is

> *"The process of finding hidden information or structure in a data [...] [set.] This includes extraction, selection, preprocessing, and transformation of features describing different aspects of the data".*

The following overview in Figure 2.5 presents a possible approach to knowledge discovery, as applied in chapters 11 and 12. It is not claimed here that this view is the only approach available in this research field. The remainder of this chapter will describe the various tasks involved in knowledge discovery which are shown in Figure 2.5.

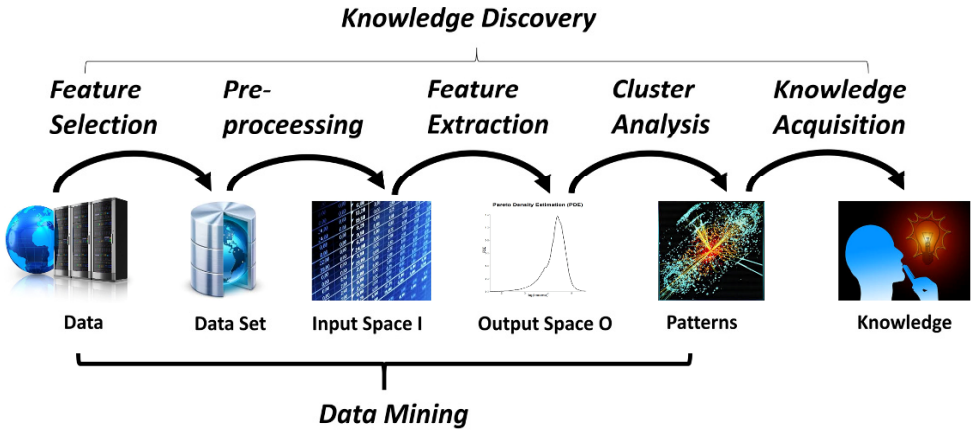

Figure 2.5: The step-wise process of knowledge discovery, as inspired by [Fayyad et al., 1996, p. 10; Ultsch, 2000b]. The systematic process may contain loops between any steps [Behnisch/Ultsch, 2015, p. 52]. This work focuses on Clustering analysis which will be separately discussed in the next chapter, but in general applying Machine learning algorithms would be the 4th step.

2.3.1 Feature Selection

In the first step, the "features must be properly selected so as to encode as much information as possible concerning the task of interest. […] minimum information redundancy among the features is a major goal" [Theodoridis/Koutroumbas, 2009, pp. 596-597] (see also [Lee/Verleysen, 2007, p. 230]). Redundancy refers to a case in which certain features of a data set are not independent of each other [Lee/Verleysen, 2007, pp. 1-2]. For example, if the two variables l and j are correlated, then $D(l,j) = \sqrt{\sum_i l_i - j_i}$ is no longer a Euclidean distance [Cormack, 1971, p. 326].

2.3.2 Preprocessing

> *"Preprocessing the data to be mined is utterly important for a successful outcome of the analysis. If the data is not cleansed and normalized, there is a high danger of getting spurious and meaningless results. Cleansing includes the removal of outliers, i.e., data objects with extreme values, replacement of missing values, or the removal of erroneous corresponding data sets" [Mörchen, 2006, pp. 7-8].*

Sometimes, this first step is already referred to as feature extraction [Bishop, 2006, p. 2]. Many data mining methods rely on the concept of (dis-)similarity between pieces of information encoded in data. For example, for Euclidean distances, "normalization of the data needs to be considered to avoid undesired emphasis of features with large ranges and variances" [Mörchen, 2006, p. 8] (see also [Jain/Dubes, 1988, p. 38]). This process of creating such "syntetic" data features that retain the most important information of a pattern in question is here called feature extraction (consistent with [Mirkin, 2005, p. 208]).

2.3.3 Feature Extraction

The first step of feature extraction is to determine the distribution of each individual variable.

> *"Important tools for this inspection are the quantile-quantile plot (QQ-plot) and kernel estimators for the probability density function (pdf). Here we use the PDE method for pdf estimation [Ultsch, 2003b] as it is specially designed to uncover subsets in the variables" [Behnisch/Ultsch, 2015, p. 54].*

A QQ-plot makes it possible to compare the given distribution of a variable to standard distributions. Additionally, box-whisker diagrams (boxplots) may be used to visualize the quartiles of a variable.

2.3.3.1 Transformations

> *"Real valued data often comes from domains where variables have greatly varying variances because of different scales. Variables with large variances are likely to dominate the obtained distance structure, e.g. when using Minkowski metrics. To overcome this problem, each variable is linearly transformed (standardized) such that the estimated variance is the same on all variables. The Z-score scheme transforms a variable's values* $x \leftarrow (x - m)/\sigma$ *with mean m and standard deviation* σ*"* [Herrmann, 2011, p. 28].

If a variable can be non-linearly transformed to a normal distribution, the Box-Cox algorithm (see [Asar et al., 2014]) is often used to estimate the factor of the transformation. With an approximation of the factor obtained from the ladder of powers [Tukey, 1977], an "understandable" transformation, e.g., "log" or "sqrt," can be applied that is as near as possible to the factor of the Box-Cox algorithm. "These allow for hypotheses on why the distribution is shaped in a particular way" [Behnisch/Ultsch, 2015, p. 56].
For non-normally distributed variables (e.g., a variable with a multimodal distribution), a meaningful variance σ^2 may be difficult to estimate. "Instead, a (robust) min/max-standardization transforms a variable's values $x \leftarrow \frac{x-min(x)}{max(x)-min(x)}$ with robust estimates $min(x)$, $max(x)$ for minimum and maximum values. There is empirical evidence by Milligan and Cooper [Milligan/Cooper, 1988] that min/max standardization is to be preferred over Z-score, especially if variances of underlying distributions is [sic] hard to estimate" [Herrmann, 2011, p. 28]. In this context, $max(x)$ and $min(x)$ are estimated as the 95th and 5th percentiles, respectively, of the distribution [Herrmann, 2011, p. 127].

2.3.3.2 Dimensionality Reduction

A common approach to feature extraction is dimensionality reduction (DR). To cope with the "curse of high dimensionality" (for further details, see [Verleysen et al., 2003]), dimensionality reduction reduces an input space I$\subset \mathbb{R}^d$ to an output space O$\subset \mathbb{R}^m$ such that $m < d$ [Lee/Verleysen, 2007].

> *"All difficulties that occur when dealing with high-dimensional data are often referred to as the 'curse of dimensionality'. When data dimensionality grows, the good and well-known properties of the usual 2D or 3D Euclidean spaces make way for strange and annoying phenomena" [Lee/Verleysen, 2007, p. 3].*

The various phenomena related to this concept are explained in [Lee/Verleysen, 2007, pp. 4-9] (see also [Bellman, 1957]). A DR method is usually either a manifold learning method or a projection method. DR methods such as autoencoders [Hinton/Salakhutdinov, 2006], Isomap [Tenenbaum et al., 2000] or local linear embedding (LLE) [Roweis/Saul, 2000] that are designed to find a manifold[5] that represents a given set of high-dimensional data[6] are called *manifold learning* methods. Such methods are disregarded here because these manifolds usually have more than two dimensions. DR methods of the type known as projection methods are

[5] "A manifold is a connected region. Mathematically, it is a set manifold of points, associated with a neighborhood around each point. From any given point, the manifold locally appears to be a Euclidean space." [Goodfellow et al., 2016, p. 160]

[6] Often described using the term *intrinsic dimension* (e.g., [Lee/Verleysen, 2007, pp. 18-24, 41, 47ff]).

separately introduced in chapter 4. There, the focus is placed on methods that attempt to visualize information by means of projections that are restricted to visualizing high-dimensional data in a two-dimensional space while preserving their structure (for details, see chapter 5). The quality of a projection critically depends on the concept of dissimilarity that is chosen to be applied to the input space *I*. This concept could be a definition based on either distance or local proximity. An index used to evaluate the quality of a projection is called a quality measure (QM), and 19 QMs are introduced in chapter 6.

2.3.4 Cluster Analysis

Many data mining methods rely on some concept of the dissimilarity between pieces of information encoded in the data of interest. These methods are used for cluster analysis, and common approaches will be described in the next chapter. Cluster analysis is the task of unsupervised classification that results in a clustering. Given a data set *I* that contains *n* data points, the objective of cluster analysis is to group the data points into *K* disjoint subsets of *I*, denoted by $c_1, \dots, c_K$ [Hennig et al., 2015, p. 2]. "A clustering is […] the partition obtained" with $\mathrm{K} = \{c_1, \dots c_K\}$. If a data point *l* belongs to a cluster c_g, then it has the class label $g \in \mathbb{N}$. In the literature, this process is often called hard clustering to distinguish it from methods such as fuzzy clustering, in which a fractional degree of membership is assigned to each $l \in I$ [Jain et al., 1999].

Cluster

No generally accepted definition of clusters exists in the literature [Hennig et al., 2015, p. 705]. When describing clusters, the term *pattern* is often used (e.g., [Theodoridis/Koutroumbas, 2009]).

Here, consistent with Bouveyron et al., it is assumed that a cluster is a group of similar objects [Bouveyron et al., 2012]. Chapter 3 will elaborate on this statement while presenting the definition of *natural* clusters.

Intracluster Distance

Let $c_p \subset I$ be a cluster such that $\forall c_q \subset I$, where $p, q \in \{1, \dots, k\}$ and $p \neq q$, $c_p \cap c_q = \{\}$; then, the distance $Intra(c_p) \coloneqq D(l, j)$ between two data points $j, l \in c_{p,}$ is called an intracluster distance.

Intercluster Distance

Let $c_p \subset I$ and $c_q \subset I$ be two clusters such that $p, q \in \{1, \dots, k\}$, $c_p \cap c_q = \{\}$, and $p \neq q$; then, the distance $Inter(c_p, c_q) = \mathrm{D}(j, l)$ between two data points j and l in the two clusters, $j \in c_p$ and $l \in c_q$, is called an intercluster distance.

Compact Structures

Compact structures in a data set are mainly defined by distances *d* if discontinuity in data exist such that the intracluster distances are small and the intercluster distances are large. Note, that the distance distribution is often bimodal if the data structures are compact. This type of structures leads to natural clusters (see chapter 3).

Connected Structures

Connected structures in a data set are mainly defined by density $\rho(\vec{v})$ if discontinuity in data exist. If a connected graph Γ is chosen appropriately regarding the data set, these data structures are based on neighborhoods $H_j\ (k, \Gamma, M)$. This type of structures leads to natural clusters (see chapter 3).

2.3.5 An Approach to Knowledge Acquisition

If, for a given data set, there exist labels defined by a clustering or a domain expert, the next step may be to determine what each cluster means [Behnisch/Ultsch, 2015, p. 65] or what kind of knowledge can be acquired from it[7].

> *"Under knowledge we understand a symbolic representation of objects, facts and rules for an interpreter with symbol processing capability, e.g. a human[8]. In particular, knowledge is communicable by word or writing" [Ultsch, 1994, p. 1] (see also [Ultsch, 1987, p. 22]).*

Knowledge has the properties of being valid, comprehensible, nontrivial, potentially innovative and useful in practice [Behnisch/ Ultsch, 2015, p. 52]. It can be stored in a knowledge base, which "is an organized collection of knowledge together with operations for accessing and manipulating knowledge" [Ultsch, 1987, p. 22]. One example of a representation of knowledge is a rule [Ultsch, 2016c], which is defined as a prescription regarding how to generate, interpret and manipulate facts [Ultsch, 1987, p. 22].

In the context of knowledge discovery, knowledge acquisition can be defined "as the encoding of knowledge into the formal representation scheme of a knowledge-based system [KBS]" [Ultsch, 1987, p. 23]); here, a KBS is defined as "a computer program that contains an explicit, formal representation of knowledge in a knowledge base and is capable of [drawing conclusions[9]]" [Ultsch, 1987, p. 23]. In another context, researchers may interview domain experts "to become educated about the domain and to elicit the required knowledge, in a process called knowledge acquisition" [Russell et al., 2003, p. 217]. In short, knowledge acquisition can be described as a process that leads to a formal representation of knowledge (see [Aikins, 1983]), for example, a process leading to the generation of rules required for a computer program, e.g., DENDRAL [Russell et al., 2003, p. 22] or MYCIN [Aikins, 1983]. One possible approach to knowledge acquisition is to use machine learning [Russell et al., 2003, p. 687]. With regard to understandability, the machine learning methods used for this purpose can be classified as either symbolic or sub-symbolic methods [Ultsch/Korus, 1993].

> *"Sub-symbolic methods model the structure of data using many numerical parameters. They are usually aimed at prediction or classification. The output of sub-symbolic methods often depends on the values and interactions of most or all model parameters. They fail to explain the prediction or classification. There are certainly areas of data mining where it is sufficient to build such black-box models that can approximately reproduce a classification or predict future data. An important requirement for knowledge discovery is the interpretability of the results. In many domains the expert wants to know why a decision was made or what a [...] pattern describes. Comprehensible descriptions of the models are crucial for success in this case"* [Mörchen, 2006, p. 120].

For the acquisition of knowledge through cluster analysis, symbolic methods are preferable, as described in chapters 11 and 12 (see also [Ultsch, 1994]). In chapter 12, decision tree learning

---

[7] In another context one would like to explain a prediction done by a machine learning algorithm.

[8] For humans 7±2 rules appear to be the optimum [Miller 1956].

[9] Formally defined as *inference* in [Ultsch, 1987, p. 22].

is used in a knowledge acquisition approach called Classification And Regression Tree (CART) analysis [Breiman et al., 1984]). This method relies on a binary tree in which the splitting criteria (decisions) for the vertices are expressed in terms of the Gini index (for further details, see [Safavian/Landgrebe, 1990, p. 15]).

"A class is described by a number of conditions" [Ultsch/Korus, 1993, p. 3] that lead to the generation of a subset $G_i \subset C$ defined by a previously identified clustering. Additionally, for each class, a unique class label $g \in \mathbb{N}$ exists for all $o \in G_i$. Every observation $o \in G_i$ can be unambiguously described by one or more properties that are shared among all observations of G_i. Here, the conclusion that an observation can be correctly assigned to a class G_i is reached based on the conditions defining a path (rule) from the corresponding leaf to the root of the binary tree, and this conclusion is called the decision to place o in G_i. Therefore, the class G_i has a semantic characterization because it is characterized by the rules governing the decision tree, which allow this class to be distinguished from other classes. Here, it is assumed that the last step in the evaluation of a clustering is to ask domain experts to validate the identified classes.

3 Approaches to Cluster Analysis

Many data mining methods rely on some concept of the similarity between pieces of information encoded in the data of interest. Various names have been applied to these clustering methods, depending largely on the field of application in data science. For example, in biology the term "numerical taxonomy" is used [Thorel et al., 1990], in psychology the term Q analysis is sometimes employed, market researchers often talk about "segmentation" [Arimond/Elfessi, 2001] and in the artificial intelligence literature, unsupervised pattern recognition is the favored label [Everitt et al., 2001, p. 4]. The corresponding methods can be either data-driven or need-driven. The latter, called also constraint clustering [Tung et al., 2001] aims at organizing the true structure to meet certain application requirements such as energy aware sensor networks, privacy preservation, and market segmentation [Ge et al., 2007, p. 320]. An overview of constrained clustering algorithms can be found in [Basu et al., 2008].

Here, however, the focus is placed on data-driven[10] methods, in which patterns present in the data are used to identify homogeneous groups of objects [Arabie et al., 1996, p. 8 ff.]. Consequently, the term *cluster analysis* is used to refer to a step in the knowledge discovery process (chapter 2, Figure 2.5.). Let it be assumed that in Figure 3.1 (top left), the first data set (I) contains two variables[11]. The division of this homogeneous data set into different patterns would be called dissection [Everitt et al., 2001, p. 7]. By contrast, *natural clusters* do not require dissection; instead, they are clearly separated in the data [Duda et al., 2001, p. 539; Theodoridis/Koutroumbas, 2009, pp. 579, 600], as shown in the second data set (II) in Figure 3.1 (top right).

No generally accepted definition of clusters exists in the literature [Hennig et al., 2015, p. 705]. Additionally, Kleinberg showed for a set of three simple properties (scale-invariance, consistency and richness), that there is no clustering function[12] satisfying all three [Kleinberg, 2003]. By concentrating on distance and density based *structures*[13], this work restricts clusters to "natural" clusters (see section 2) and therefore omits the axiom of richness where all partitions should be achievable. Consequently, only natural clusters, in which objects are similar within clusters and dissimilar between clusters [Bouveyron et al., 2012], are considered here. For example, the distance distribution in the input space can be bimodal, indicating a distinction between the inter- versus intracluster distances: in data set I in Figure 3.1 (bottom left), no large intercluster distances exist and the distribution of the distances is unimodal, whereas in data set II in Figure 3.1 (bottom right), the distribution of the distances is bimodal because data set II contains two natural clusters with a large intercluster distance. Another example is the case in which the number of data points in one *elementary volume* ($d\vec{v}$) of the input space is higher than that in another elementary volume $d\vec{v}$, which can be estimated using a nonparametric technique for density estimation (e.g., kernel density estimation). In a third example, local proximities can be defined as structures based on neighborhoods $H_j(k, \Gamma, M)$ (see chapter 2.2.1).

[10] The progress in an "algorithmic activity" is enforced by data w.r.t. patterns (as opposite to intuition or personal experience, e.g. through the setting of parameters).

[11] In fact, this figure shows a CCA projection of the leukemia data set (see chapter 9).

[12] "[A]ny function f that takes a set S of n points with pairwise distances between them, and returns a partition of S" [Kleinberg, 2003, p 2].

[13] They can be described as patterns identified based on discontinuity.

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_3

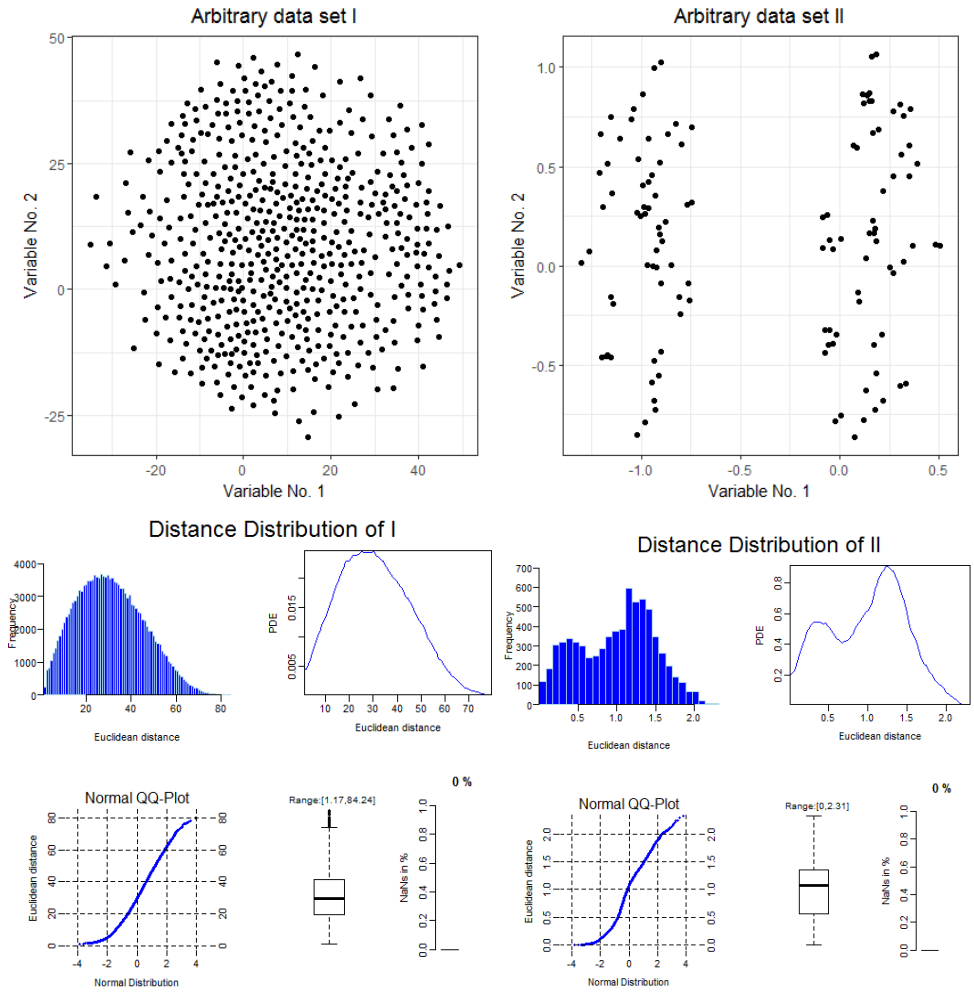

Figure 3.1: Data set I is an approximately homogeneous data set with patterns that form no natural clusters (left, top). The distance distribution in this case is not bimodal (left, bottom). Data set II contains two natural clusters with a large intercluster distance (right, top). The distance distribution is bimodal here (right, bottom). See Figure 12.2 or supplement B for a high-dimensional example. Distance distributions was generated using the AdaptGauss CRAN package [Thrun/Ultsch, 2015; Ultsch et al., 2015].

3.1 Common Clustering Methods

Clustering methods can be broadly divided into two groups: hierarchical and partitional methods [Jain, 2010]. Partitional clustering methods simultaneously divide a set of data points into subsets. Because we are concentrating on *natural clusters,* overlapping clustering is not considered here. It should be remarked that the choice of the clustering algorithm to be used is more important than the choice of the distance calculation [Jain/Dubes, 1988, p. 140].

A prominent example of a partitional clustering method is the well-known *k-means* method of [MacQueen, 1967] (originally from [Steinhaus, 1956]). It proceeds as follows: Once the number

of clusters has been chosen, a random initialization of cluster centers, called centroids, is performed in the input space. Then, the nearest data points to each centroid are assigned to that centroid. After the mapping of the data points, the centroids are moved such that the distances from the assigned points to their corresponding centroids are minimized. This process is performed repeatedly. Figure 3.2 illustrates four iterations of the process. In summary, k-means centroids are average points rather than individual data points. Details about the algorithm can be found in [Hennig et al., 2015, p. 68ff].

By contrast, the clustering method called partitioning around medoids (PAM), introduced in [L. Kaufman/Rousseeuw, 1990], minimizes the sum of the distances from the data points within a cluster to one chosen data point in the same cluster, called the medoid [Mirkin, 2005, p. 181]. In other words, the average distance between a medoid and a subset of data points in the same cluster is minimized. Aside from the change from centroids to medoids, the algorithm can be formulated analogously to k-means [Mirkin, 2005, p. 182].

Hierarchical clustering algorithms are based on the "representation of data as a hierarchy of clusters nested over set-theoretic inclusion" [Mirkin, 2005, p. 112]. In the agglomerative approach, such an algorithm begins with each data point in its own cluster and successively merges the most similar pairs of clusters to form a cluster hierarchy[14].

A typical visual representation of this process is called a dendrogram (Figure 3.3). A dendrogram is a tree showing a hierarchical structure of distance-based connections between subsets of points. The similarity between points or groups of points depends on the algorithm. [Bock, 1974] demonstrated (see chapter 2 for details) that for every dendrogram, an ultrametric space can be constructed in which the triangle inequality is redefined as

$D(l,j) \leq max\,(D(l,m), D(m,j))$.

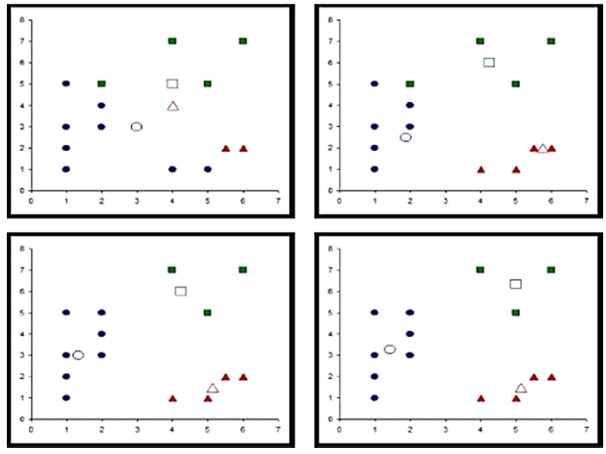

Figure 3.2: Steps of iteration using the k-means algorithm. After a random initialization of three centroids the nearest data points are assigned to each centroid. Then the centroids are moved to minimalize the distances.

[14] The divisive approach is not considered here (see [Mirkin, 2005, p. 113 ff] for details).

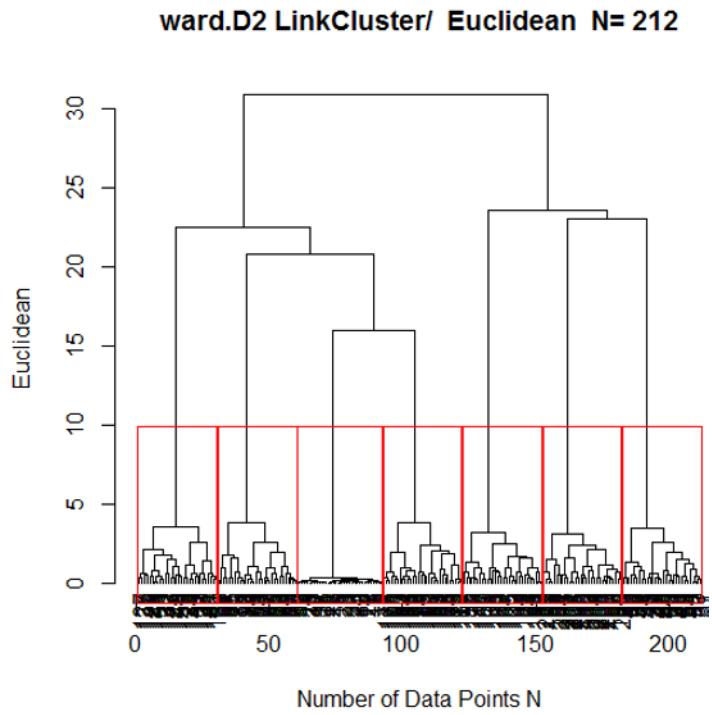

Figure 3.3: Dendrogram of the Hepta data set based on the Ward algorithm. Large changes in fusion levels of the ultrametric portion of the Euclidean distance in the Ward algorithm (y-axis) indicate the best cut. Seven clusters are indicated by red boxes at the y-axis value of 10. If only small changes in the fusion levels exist, it indicates that the algorithm is not able to find a cluster structure.

One of the most common hierarchical clustering algorithms is called *single linkage* (SL) [Florek et al., 1951; Sokal/Sneath, 1963], in which the clustering process is agglomerative [Jain et al., 1999]. In SL, the similarity between two subsets of data points is defined as the minimum distance between data points in these subsets [Duda et al., 2001, p. 553].

Let $\widetilde{D}$ be the distance between two clusters $c_1 \subset I$ and $c_2 \subset I$, and let $D(l,j)$ be the distance between two data points in the input space I; then, SL is defined based on (see [Hennig et al., 2015, p. 9]) $\widetilde{D}(c_1, c_2) = \min_{l \in c_1, j \in c_2} D(l,j)$.

In graph theory terminology, this process generates a tree [Duda et al., 2001, p. 553]. If it is allowed to continue until all subsets of points are linked, the result is a (minimal) spanning tree (MST) [Duda et al., 2001, pp. 553, 554; Jain/Dubes, 1988, p. 70]. Of all common algorithms developed before 1968, only SL satisfies all conditions of a "theoretically valid" clustering (see [Jardine/Sibson, 1968] for details).

Another hierarchical clustering algorithm that will be used here is called the *Ward* algorithm [Ward Jr, 1963]. In the Ward algorithm, the similarity between two subsets of points is based on an optimal value of an objective function, which commonly is the sum of squared errors (*SE)*.

Let $c_r \subset I$ and $c_q \subset I$ be two clusters such that $r, q \in \{1, \dots, k\}$ and $c_r \cap c_q = \{\}$ for $r \neq q$, and let the data points in the clusters be denoted by $j_i \in c_q$ and $l_i \in c_r$, with the cardinality of the sets being $k = |c_q|$ and $p = |c_r|$ and with

$$m(c_q) = \frac{1}{k}\sum_{i=1}^{k} j_i \text{ and } m(c_r) = \frac{1}{p}\sum_{i=1}^{p} l_i;$$

then, the *SE* is defined as (see [Theodoridis/Koutroumbas, 2009, pp. 661-663])

$$SE = \frac{k * p}{k + p} \sum_{i=1}^{n} \left(m(c_k) - m(c_p)\right)^2$$

In Figure 3.3, the ultrametric property of the Ward algorithm is represented in a dendrogram (for further details, see [Duda et al., 2001, p. 557; Everitt et al., 2001, p. 68ff; Jain/Dubes, 1988]). If the values on the y axis "for the levels are roughly evenly distributed throughout the range of possible values, then there is no principled argument that any particular number of clusters is better or more natural than another" [Duda et al., 2001, p. 551]. "Large changes in fusion levels are taken to indicate the best cut" [Everitt et al., 2001, p. 76]. The cut depicted in Figure 3.3 generates a clustering consisting of seven clusters of roughly equal size.
The next clustering method used in this work is called spectral clustering.

> *"[It] is a class of graph-based techniques that unravel the structure properties of a graph using information conveyed by the spectral decomposition [eigendecomposition [see [Goodfellow et al., 2016, pp. 42-44]]] of an associated [Laplacian] matrix. The elements of this matrix code the underlying similarities among nodes [data points] of the graph" [Theodoridis/Koutroumbas, 2009, p. 772].*

> *"The K principal eigenvectors of the Laplacian matrix provide a mapping of the objects into K dimensions. To obtain clusters, the resulting K-dimensional vectors are clustered by standard methods, usually K-means. There are various interpretations of this. […]. For these [Euclidean] data, spectral clustering acts as a remarkably robust linkage method." [Hennig et al., 2015, p. 10].*

There is a close resemblance between spectral clustering and manifold learning methods [Theodoridis/Koutroumbas, 2009, p. 779]. Here, the clustering algorithm of [Ng et al., 2002] is used to take advantage of the open-source implementation of this method that is available in the R language [R Development Core Team, 2008].
"Clustering via mixtures of parametric probability models is sometimes in the literature referred to as 'model-based clustering'" [Hennig et al., 2015, p. 10]. With the clustering algorithm of [Fraley/Raftery, 2006] in mind, here, this clustering method is called the *mixture of Gaussians* (MoG) method. The MoG method uses the *expectation maximization* (EM) algorithm (for further details on the EM algorithm, see [Bishop, 2006]).

> *The EM algorithm is "an algorithm of alternating maximization applied to the likelihood function for a mixture of distributions model. At each iteration, EM is performed according to the following steps: (1) Expectation: Given parameters of the mixture P_k and individual density functions a_k, find posterior probabilities for observations to belong to individual clusters g_{ik} […]. (2) Maximization: given posterior probabilities g_{ik}, find parameters P_k, a_k maximizing the likelihood function" [Mirkin, 2005, p. 178].*

The MoG method suffers "from the well-known curse of dimensionality [Bellman, 1957], which is mainly due to the fact that model-based clustering methods are over-parametrized in high-dimensional spaces" [Bouveyron/Brunet-Saumard, 2014, p. 53]. To solve this problem, "for model based clustering, variable selection can be tackled within a Bayesian framework" [Bouveyron et al., 2012]. In the case of the MoG clustering method, the optimal model can be

calculated according to the Bayesian information criterion [Aho et al., 2014] for parameterized Gaussian mixtures that are EM initialized using hierarchical agglomeration [Fraley/Raftery, 2002, pp. 10-12].

> *"In each hierarchical agglomeration, each stage of merging corresponds to a unique number of clusters, and a unique partition of data. A given partition can be transformed into indicator variables […] which can then be used as conditional probabilities in an M-step of EM for parameter estimation, initializing an EM iteration" [Fraley/Raftery, 2002, p. 11]. Here, the R package mclust is used [Fraley/Raftery, 2006].*

3.2 Structure of Natural Clusters

> *"Clusters can be of arbitrary shapes (structures) and sizes in a multidimensional pattern space. Each clustering criterion imposes a certain structure on the data, and if the data happen to conform to the requirements of a particular criterion, the true clusters are recovered. Only a small number of independent clustering criteria can be understood both mathematically and intuitively. Thus the hundreds of criterion functions proposed in the literature are related and the same criterion appears in several disguises"* [Jain/Dubes, 1988, p. 91].

This section analyzes common clustering algorithms from the perspective of structures, whereas in various other sources, the clustering criterion or objective function has been understood only intuitively. Here, it is argued that the main argument of Jain and Dubes has received overall consent from the clustering community: Different clustering methods tend to implicitly assume different structures of clusters [Duda et al., 2001, pp. 537, 542, 551; Everitt et al., 2001, pp. 61, 177; Handl et al., 2005; Theodoridis/Koutroumbas, 2009, pp. 862, 896; Ultsch/Lötsch, 2016].

3.2.1 Types of Structures Sought by Clustering Algorithms

The argument of Handl et al. is partially adopted here, in which natural clusters are considered to exhibit two types of structures, called compact and connected structures [Handl et al., 2005], as depicted in Figure 3.4. Clusters with compact structures show small variations in their intracluster distances; connected structures are based on the idea of neighborhoods of data points [Handl et al., 2005]. Here, a compact structure is considered to be mainly defined by inter- versus intracluster distances, whereas a connected structure is primarily defined by neighborhoods H of data. Using the definitions presented in section 2.2.1, neighborhoods can be identified based on graph theory. This can result in connected structures consisting of either unidirectional or direction-based neighborhoods.

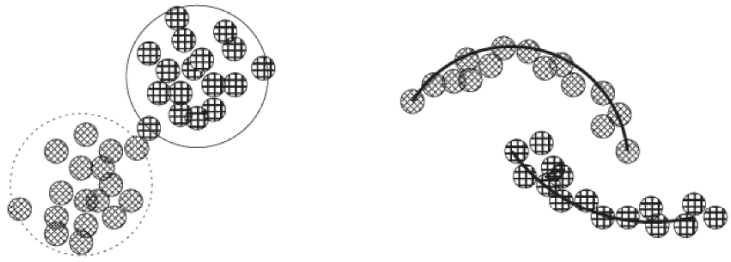

Figure 3.4: Two types of cluster structures, compact (left) and connected (right), taken from [Handl et al., 2005]. Here, a compact structure is considered to be mainly defined by intra- versus intercluster distances, whereas a connected structure is primarily defined based on neighborhoods $H_j\ (k, \Gamma, M)$ and the density of the data.

An example of an algorithm that seeks compact clusters is the k-means clustering algorithm, which imposes a spherical cluster structure [Duda et al., 2001, p. 542; Handl et al., 2005, p. 3202; Hennig et al., 2015, p. 61; Mirkin, 2005, p. 108; Theodoridis/Koutroumbas, 2009, p. 742] such that the clusters cannot be too elongated [L. R. Kaufman/Rousseeuw, 2005, p. 117]. This cluster structure can be found in a data set if "the data points are actually normally distributed" (…) because "the sample mean tends to fall in the region where the samples are most densely concentrated" [Duda et al., 2001, p. 537]. The k-means algorithm is sensitive to noise and outliers [Theodoridis/Koutroumbas, 2009, p. 744]. "This drawback […] gave rise to the k-medoids algorithms […]." The PAM algorithm is less sensitive to outliers. Because of its strong similarity to the k-means algorithm, it is assumed here that PAM also yields a compact spherical cluster structure.

Examples of algorithms that seek connected clusters include density-based methods such as DBscan [Ester et al., 1996] and SL [Handl et al., 2005]. Because SL searches for nearest neighbors [Cormack, 1971, p. 331], it tends to produce connected and chain-like structures [Duda et al., 2001, p. 554; Everitt et al., 2001, p. 67; Hartigan, 1981; Jain/Dubes, 1988, pp. 64-65; Theodoridis/Koutroumbas, 2009, p. 660]. A nearest neighbor is also a Delaunay neighbor (Figure 3.4), leading to a direction-based connected structure of clusters. Spectral clustering is based on graph theory and consequently searches for connected structures [Ng et al., 2002, p. 5] of clusters with "chain-like or other intricate structures" [Duda et al., 2001, p. 582]. This indicates that such an algorithm also searches for direction-based connected clusters (see also [Hennig et al., 2015, p. 10]). "They [spectral clustering methods] are well-suited for the detection of arbitrarily shaped clusters, but can lack robustness when there is little spatial separation between the clusters" [Handl et al., 2005, p. 3202].

The Ward algorithm is sensitive to outliers and tends to find compact clusters of equal size [Everitt et al., 2001, p. 61, Tab. 1] that are ellipsoidal in structure [Ultsch/Lötsch, 2016]. The MoG method uses a mixture-of-distributions approach, which leads to connected clusters. Contrary to [Handl et al., 2005], it is argued here that the MoG method should be able to separate clusters that are non-linear separable (e.g., Chainlink [Ultsch/Vetter, 1995]). Jains and Dubes report that "fitting a mixture density model to patterns" creates clusters with hyper-ellipsoidal shapes [Jain/Dubes, 1988, p. 92]. [Handl et al.] report that the MoG method is very effective for well-separated clusters [Handl et al., 2005, p. 3202].

In the case of self-organizing mapping (SOM)[15], the structures have been reported to be of "very general shapes" [Duda et al., 2001, p. 582; Ultsch/Lötsch, 2016]. Similarly to the emergent SOM (ESOM)/U-matrix clustering method [Ultsch et al., 2016a], the Databionic swarm (DBS) method that is discussed later in this work also uses the concept of emergence[16], through which novel properties can arise in a system. Emergence leads to clusters whose structures are not predefined.

To summarize, the cluster structures that are theoretically sought by various methods are visualized in Figure 3.5. It should be noted that clustering methods that search for clusters with connected structures should also be able to find compact clusters as long as the distance between

[15] However, for k-means-SOM of the batch type, spherical or well-separated structures have been reported [Handl et al., 2005, p. 3202] (see the SOM section in chapter 4 for the differences between ESOM and k-means-SOM).

[16] Definition, see chapter 7.3, p. 81-82

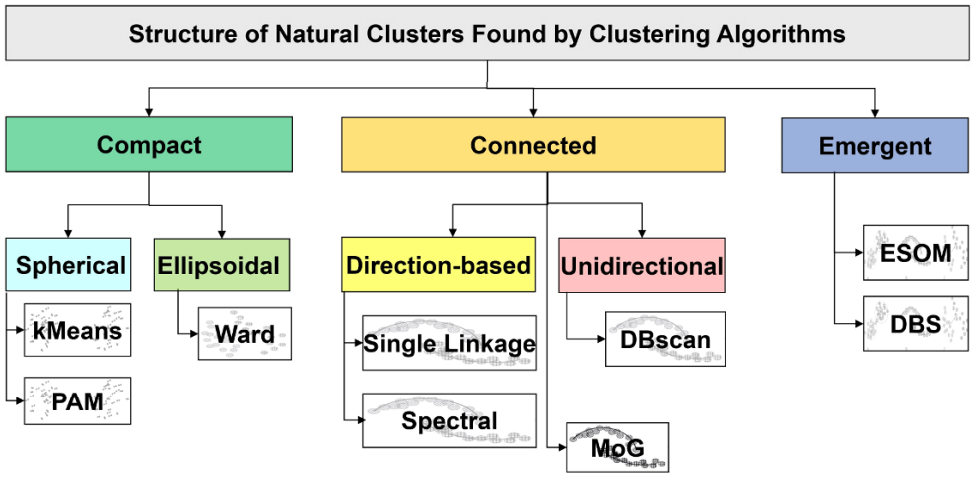

Figure 3.5: Overview of the cluster structures that common clustering algorithms tend to find. It is based on the literature, except for the MoG algorithm[17], for which an educated guess is made. The subgroup of DBscan clustering is characterized based on arguments presented in section 3.2.1, for the definition of emergent see chapter 7.3.

clusters is large or the density between clusters is very low (see also [Handl et al., 2005, p. 3202]); e.g., "single-linkage clusters detect high-density clusters if there is a low enough valley separating them" [Hartigan, 1981]. However, methods that search for compact and spherical structures cannot be expected to find connected structures.

3.2.2 *Quality of Clustering*

> *"[The quality of clustering is measured using a] "procedure for validating a cluster structure […]. This can be based on an internal index, an external index or resampling. An internal index scores the degree of correspondence between the data and the cluster structure. An external index compares the cluster structure with a structure given externally. A resampling is used to see whether the cluster structure is stable with respect to data change" [Mirkin, 2005, p. 205] (see also [Jain/Dubes, 1988, p. 161ff]).*

Internal and external indices are also often called *intrinsic* or *extrinsic* indices, respectively; here, they are referred to as *supervised* or *unsupervised* indices, respectively.
The simplest example of a supervised index is the accuracy, which is defined as follows:

$$\text{Accuracy } [\%] = \frac{[\text{No. of true positives}]}{[\text{No. of cases}]} \qquad (3.1)$$

In Eq. 3.1, the number of true positives is the number of labeled data points for which the label defined by a prior classification is identical to the label defined after the clustering process.
To determine either the number of clusters or the clustering quality, two approaches are generally possible. Covariance matrices can be calculated, or the intra- versus intercluster distances can be compared to evaluate the homogeneity versus heterogeneity of the clusters. In the literature, a sufficient overview of 15-30 indices has already been provided [Charrad et al., 2012; Dimitriadou et al., 2002], and these indices will not be further discussed here. A special type of unsupervised indices, referred to as quality measures for projection methods, will be separately

[17] Also known as model-based clustering.

introduced in chapter 6. Two unsupervised indices and corresponding visualizations are presented in the following sections.

3.2.2.1 Heatmaps

A heatmap is an example of an unsupervised index. For the ordering of the data points in heatmaps, dendrograms are often used. They enable the visualization of high-dimensional information and dissimilarity matrices without projecting them into a lower-dimensional space. Their use strongly depends on the sequence of the observations. For cluster validation, it is desirable to plot observations that are in the same cluster together [Hennig et al., 2015].

> *"[A heatmap] consists of a rectangular tiling, with each tile shaded on a color scale to represent the value of the corresponding element of the data set. The rows (columns) of the tiling are ordered such that similar rows (columns) [in the sense that they are in the same cluster] are near each other" [Wilkinson/Friendly, 2012]. "The cluster heat map is a rectangular tiling of a data matrix with cluster trees appended to its margins. Within a relatively compact display area, it facilitates inspection of joint cluster structure" [Wilkinson/Friendly, 2009].*

Unlike in [Wilkinson/Friendly, 2009; Fig. 1], in Figure 3.7, the dendrogram between the variables is disregarded and only the nxn heat map of the distance matrix is shown.

3.2.2.2 Silhouette plots

The Silhouette plot is a common unsupervised index for visual evaluation of a clustering [L. R. Kaufman/Rousseeuw, 2005].

> *"A score function s: $X \rightarrow [-1, 1]$ evaluates the positioning of data objects inside their assigned cluster. Let a(x) denote the average distance between x and all other objects of the same cluster, and b(x) denotes the smallest average distance between x and all objects of another cluster. The silhouette score follows as $(x) = \frac{b(x)-a(x)}{max\{a(x),b(x)\}}$. Silhouette scores similar to 1 indicate objects that have been assigned to an appropriate cluster, whereas −1 indicates objects that have been badly classified. Silhouette scores similar to 0 indicate objects that lie in between clusters. Each cluster is represented by one silhouette, showing which objects lie within the cluster and which objects merely hold an intermediate position. The entire clustering is displayed by plotting all silhouettes into a single diagram, from which the quality of the clusters can be compared" [Herrmann, 2011, pp. 91-92].*

A reasonable clustering is characterized by a silhouette width of greater than 0.5, and an average width below 0.2 should be interpreted as indicating a lack of any substantial cluster structure [Everitt et al., 2001, p. 105]. However, it is evident that silhouette scores assume clusters that are spherical or Gaussian in shape [Herrmann, 2011, pp. 91-92].

3.3 Problems with Clustering Methods

To illustrate several problems encountered when using common clustering methods, a domain expert measured genetic data for subjects who were known either to be healthy or to have one of 3 subtypes of leukemia. Here, a typical knowledge discovery task could be to identify patterns in the cancer subtypes based on the four diagnoses leading to the prior classification**.**

> *"[I]t is a common practice among researchers to employ a variety of different clustering techniques to analyse a dataset, and to use visual inspection[18] and prior biological knowledge to select what is considered the most 'appropriate' result" [Handl et al., 2005, pp. 3202-3203].*

Consequently, the first step would be to confirm that the structure defined by the classification distinguishing the healthy patients from the non-healthy ones does indeed exist in this data set.

[18] The application of visual inspection will be reported in chapter 6, Fig. 1, resulting in arbitrary projections.

The data set used as an example to illustrate the general problem described above contains data representing 7747 variables for 554 subjects (see chapter 9 for details). Of the subjects, 109 are healthy, 15 have acute promyelocytic leukemia (APL), 266 have chronic lymphocytic leukemia (CLL), and 164 have acute myeloid leukemia (AML). There is a possibility that some subjects might be misclassified, but a future publication will address this diagnostic.

The heatmap and the silhouette plot presented in Figure 3.7 and 3.6 show that this data set is defined by discontinuities because the intracluster distances are small and the intercluster distances large. Hence, the leukemia data set is a high-dimensional data set with natural clusters that are specified by the illness status and defined by discontinuities[19].

Table 3.1 shows the accuracies of common clustering algorithms computed by comparing the clustering results with the prior classification made available by the domain expert. The default settings were used for all algorithms, and the number of clusters was assumed to be four. The MoG algorithm cannot be applied without first using dimensionality reduction methods because the dimensionality of the data set is too high. Only one algorithm (Ward) is able to fully reproduce the prior classification. However, a classification should typically be reproduced using more than one algorithm, and the reproduction of a classification with 100% accuracy is unusual.

This example illustrates that "Clustering algorithms will create clusters whether the data are naturally clustered or purely random" [Jain/Dubes, 1988, p. 201] and "By imposing a predefined shape on the clusters, classical algorithms occasionally suggest a cluster structure in homogenously distributed data or assign points to incorrect clusters" [Ultsch/Lötsch, 2016].

To summarize, the unsupervised indices, namely, the heatmap and the silhouette plot, agree with the prior classification provided by the domain expert, whereas the external index of accuracy and the projections of the data[5] disagree with the domain expert. The question arises whether this data set contains natural clusters and, if so, how the structure of these natural clusters can be correctly identified or how the optimal clustering (or projection) algorithm can be chosen for the knowledge discovery task. This work will propose approaches and solutions to these problems.

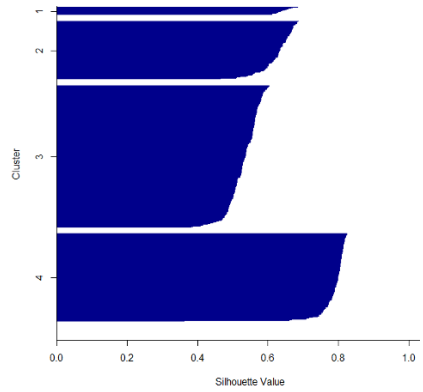

Figure 3.6: Silhouette plot of the leukemia data set indicates a cluster structure.

[19] It should be remarked that common data-driven methods as well as the heatmap and Silhouette plot do not reproduce the (sub) classification(s) of AML (like FAB subtypes) or CLL of research in this area, e.g. [Bene et al., 1995; Bennett et al., 1985; Vardiman et al., 2009; Haferlach et al., 2010], for CLL [Rosenwald et al., 2001].

Table 3.1: Accuracy results for common clustering algorithms.
No result could be calculated for the MoG algorithm (also known as model-based clustering).

Algorithm	Ward	SL	k-means	MoG	PAM	Spectral
Accuracy in %	100	80.1	76.53	*Not Computable*	78.3	59.0

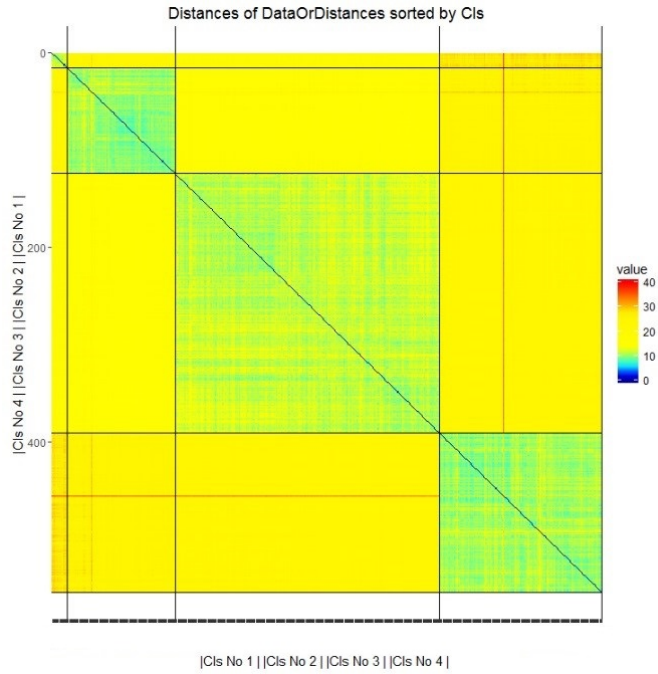

Figure 3.7: The heatmap of the leukemia data set with at least one outlier (red line). The intracluster distances are distinctively smaller than the intercluster distances. Cls1 =APL, Cls2= healthy, Cls3=CLL, Cls4=AML.

4 Methods of Projection

Dimensionality reduction techniques reduce the dimensions of the input space to facilitate the exploration of structures in high-dimensional data. Two general dimensionality reduction approaches exist: manifold learning and projection. Manifold-learning methods attempt to find a sub-space in which the high-dimensional distances can be preserved. These sub-spaces may have a dimensionality of greater than two. However, only two- or three-dimensional representations of high-dimensional data are easily graspable for to the human observer.
The goal of this chapter is the visualization of structures in high-dimensional data. Venna et al. argued that "manifold learning methods are not necessarily good for […] visualization […] since they have been designed to find a manifold, not compress it into a lower dimensionality" [Venna et al., 2010, p. 452], and it has been shown by van der Maaten et al that they do not outperform classical principal component analysis (PCA) for real-world tasks [L. J. van der Maaten et al., 2009].
Therefore, this chapter focuses on common projection methods. Many projection methods are characterized by an objective function that is optimized using gradient descent or a corresponding learning algorithm. The quality of the projection and, consequently, of the visualization will critically depend on the similarity concept chosen as the basis of the objective function, which may be based on either distance or local proximity; thus, the methods will be categorized on this basis. This chapter will attempt to relate the various projection approaches to the compact and connected structure types introduced in the previous chapter.

4.1 Common Approaches

Here, projection is used as a method for visualizing high-dimensional data in a two-dimensional space such that the discontinuities in the data are captured. Thus, the quality of a projection critically depends on the chosen similarity concept. This concept may be defined based on either distance or local proximity. The former type of similarity describes the arrangement of all given points in space and is sometimes called topography; the latter compares local neighborhoods and is sometimes called topology. Here, projections are called focusing if they are constructed using an iterative learning process that first adapts to the global intercluster distances and then focuses on more local intracluster distances.

4.1.1 Principal Component Analysis (PCA)

PCA assumes that the directions in the input space that show the highest variance contain the most information about the data set [Hotelling, 1933]. The coordinate system of the input space is replaced with a (principal) coordinate system in which the variance of the data is maximized. This is achieved by finding a set of weighted linear combinations of the original variables, where the weights are found through eigendecomposition (for a definition, see [Goodfellow et al., 2016, pp. 42-44]).
Pearson proposed an equivalent definition based on an objective function in which the average projection cost is minimized [Pearson, 1901]. The projection cost is defined in terms of the mean squared distances between the points $i \in I$ and the projected points $j \in O$:

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_4

$$E = \frac{1}{n}\sum_{l\in I}^{n} D(l,\hat{j}) \qquad (4.1)$$

where $\hat{j} = j + \sum_{i=m+1}^{n} b_i * \hat{u_i} = \sum_{i=1}^{m} b_i * u_i + \sum_{i=m+1}^{n} b_i * \hat{u_i}$ has the same dimension as $l \in I$. Here, *n* is the dimension of the input space *I*, m is the dimension of the output space *O*, the u_i are the basis vectors, and the b_i are constants. The minimization of *J* is achieved by choosing the basis vectors to be eigenvectors of the covariance matrix constrained by the orthonormality conditions [Duda et al., 2001, pp. 114-117]:

$$Cov(l,j) = \frac{1}{n}\sum_{l\in I}(l - mean_l)\,(j - mean_j) \qquad (4.2)$$

$$Cov(l,j) * u_i = \lambda_i u_i \qquad (4.3)$$

Now, the objective function E can be redefined in (4.4) in terms of the eigenvalues λ_i in (4.1) as

$$E = \sum_{i=m+1}^{n} \lambda_i \qquad (4.4)$$

where *n* is the dimension of the input space I and m is the dimension of the output space O. The largest eigenvalues correspond to the $1, \dots, m$ dimensions with the largest variance. Dimensions of the input space with small variances are discarded. Thus, PCA is an orthogonal projection of the data into a lower-dimensional space. It should be noted that "PCA remains a rather basic method and suffers from many shortcomings" [Lee/Verleysen, 2007, p. 226].

4.1.2 Independent Component Analysis (ICA)

> *"Independent component analysis (ICA) is a method for finding underlying factors or components from multivariate (multi-dimensional) statistical data. What distinguishes ICA from other methods is that it looks for components that are both statistically independent, and nonGaussian" [Hyvärinen et al., 2004].*

Let $I = (l_1, \dots, l_n)$ be defined as the matrix of the data in the input space. ICA assumes that *I* is a linear combination of non-Gaussian independent components *S* as follows:

$$I = S * A \qquad (4.5)$$

where *A* is a linear mixing matrix and $S = (j_1, \dots, j_n), j \in O$. ICA unmixes *I* by estimating a matrix $W = A^{-1}$ such that

$$I * W = S \qquad (4.6)$$

With the goal of estimating *W*, the central limit theorem and matrix search can be used to maximize the non-Gaussianity. In the fastICA algorithm [Hyvärinen, 1997], the non-Gaussianity is defined as the negentropy *F*, and it is approximately maximized by maximizing the objective function in (4.7)

$$E(j) \approx \left[F\{G(j)\} - F\{G(N(m=0, s=1))\}\right]^2 \qquad (4.7)$$

where *N* is a Gaussian and G is a contrast function, e.g., $G(u) = -exp(-\frac{au^2}{2})$.

Constraints on the estimated contrast function *G* include pre-whitening and the centering of the data in the input space [Hyvärinen et al., 2004].

4.1.3 Non-linear metric multidimensional scaling (MDS) techniques

Multidimensional scaling (MDS) was originally proposed by [Torgerson, 1952]. MDS techniques attempt to preserve the pairwise distances *D(l, j)* of the input space in the output space to the greatest possible extent. Therefore, MDS techniques minimize an objective (error) function *E* that is, as given in [Kruskal, 1964b], defined as

$$E(D,d) = \sum_{j,l=1,j<l}^{n} \Big(f\big(D(l,j)\big) - d(l,j)\Big)^2 \qquad (4.8)$$

where *f(D(l, j))* is a *non-metric*, monotonic transformation of the distances in the input space [Kruskal, 1964a, p. 7]. *E* is often called the stress, and *E* is minimized in an attempt to reproduce the general rank ordering of the distances. This minimization is usually performed via gradient descent.

However, the objective function *E* depends on the scale on which the distances are measured. It is preferable to normalize the objective *E* to reduce it to the same units in which the distances are expressed (Eq.4.9). Sammon mapping [Sammon] is one type of MDS technique and uses the error function

$$E(D,d) = \frac{1}{\sum_{j,l=1,j<l}^{n} D(l,j)} \sum_{j,l=1,j<l}^{n} \frac{\big(D(l,j) - d(l,j)\big)^2}{D(l,j)} \qquad (4.9)$$

4.1.4 Curvilinear Component Analysis (CCA)

When a non-linear structure is being analyzed, MDS cannot reproduce all distances. Therefore, [Demartines/Hérault] proposed a projection method that favors local neighborhoods. Curvilinear component analysis (CCA) attempts to reproduce short distances before reproducing long distances [Demartines/Hérault, 1995]. The objective function is defined in (4.10) as

$$E(D,d) = \sum_{j,l=1,j<l}^{n} \big(D(l,j) - d(l,j)\big)^2 * h(D(l,j),r) \qquad (4.10)$$

where h: $\mathbb{R} \rightarrow [0,1]$ is a neighborhood function that depends on a radius R as follows:

$$h(D(l,j),R) = \begin{cases} 1, & if D(l,j) \le R \\ 0, & otherwise \end{cases} \qquad (11)$$

4.1.5 t-Distributed Stochastic Neighbor Embedding (t-SNE)

The t-distributed stochastic neighbor embedding (t-SNE) technique is an enhanced version of SNE [Hinton/Roweis, 2002] in which the Kullback-Leibler divergence (KLD) is symmetrized and the crowding problem solved. The latter is achieved by redefining the conditional probabilities in the output space O through the application of Student's t-distribution with

$$p(l|j) = \begin{cases} \dfrac{(1 + \mathrm{d(l,j)}^2)^{-1}}{\sum_{\mathrm{l,j}\in\mathrm{I}}(1 + \mathrm{d(l,j)}^2)^{-1}}, & l \neq j \\ 0, & l = j \end{cases} \qquad (4.12)$$

In [Van der Maaten/Hinton], the distance between two data points is redefined as the conditional probability that j would pick l, where $l, j \in I$, as follows:

$$P(l|j) = \begin{cases} \dfrac{\exp\left(-\dfrac{D(l,j)^2}{2\sigma(l)^2}\right)}{\sum_{\mathrm{l,j}\in\mathrm{I}} \exp\left(-\dfrac{D(l,j)^2}{2\sigma(l)^2}\right)}, & l \neq j \\ 0, & l = j \end{cases} \qquad (4.13)$$

where $\sigma(l)$ is the variance of a Gaussian that is centered on data point *j*. If the projection is correct, then the conditional probabilities will be equal [Van der Maaten/Hinton]. Therefore, the objective function is defined using the symmetric KLD in (14) as

$$E = \sum_i \sum_j \frac{P(l|j) + P(j|jl)}{2n} * log\left(\frac{\dfrac{P(l|j) + P(j|jl)}{2n}}{p(l|j)}\right) \qquad (4.14)$$

4.1.6 Neighborhood Retrieval Visualizer (NeRV)

[Venna et al., 2010] reintroduced the idea of misses used by [Ultsch/Herrmann, 2005], where misses are similar data points $(l_I, j_I) \in i$ that are mapped onto far separated points $(l_O, j_O) \in O$ [Ultsch/Herrmann, 2005]. Conversely, if a pair of closely neighboring positions (l_O, j_O) represents a pair of distant data points, then this pair is called a false positive. From the information retrieval perspective, this approach allows one to define the precision F_P and the recall F_R for the case in which the neighborhoods are simply binary. However, [Venna et al., 2010] goes a step further by replacing such binary neighborhoods with probabilistic ones, which are loosely inspired by the SNE approach [Hinton/Roweis, 2002]. The neighborhood of the point l is defined in terms of the relevance of the $j \in I$ points around l:

$$p_l(j) = \frac{\exp\left(-\dfrac{D(l,j)^2}{\sigma_l^2}\right)}{\sum_{k\neq j} \exp(-\dfrac{D(l,k)^2}{\sigma_l^2})} \qquad (4.15)$$

where σ_l is set to the value for which the entropy of $p_l(j)$ is equal to log(knn) and knn is a rough upper limit on the number of relevant neighbors that is set by the user [Venna et al., 2010]. The authors propose a default value of 20 effective nearest neighbors. Similarly, the corresponding neighborhood in the output space is defined as

$$q_l(j) = \frac{\exp\left(-\dfrac{d(l,j)^2}{\sigma_l^2}\right)}{\sum_{k\neq j} \exp(-\dfrac{d(l,k)^2}{\sigma_l^2})} \qquad (4.16)$$

These neighborhoods are compared based on the mean of the KLD, which is used to define the precision F_P and recall F_R:

$$F_R = -\frac{1}{N}\sum_l^N \sum_{j\neq l} p_j(l) * \log(\frac{p_j(l)}{q_j(l)}) \qquad (4.17)$$

$$F_P = -\frac{1}{N}\sum_l^N \sum_{j\neq l} q_j(l) * \log(\frac{q_j(l)}{p_j(l)}) \qquad (4.18)$$

The objective function is then defined in (19) as

$$E = \lambda \sum_{ij} p_j(l) * \log\left(\frac{p_j(l)}{q_j(l)}\right) + (1-\lambda) \sum_{ij} q_j(l) * \log\left(\frac{q_j(l)}{p_j(l)}\right) \qquad (4.19)$$

The objective function E is non-linearly optimized via conjugate gradient descent. In the absence of prior knowledge, the neighborhoods p are defined as symmetric Gaussians or heavy-tailed distributions. The weighting between precision and recall must be set by the user using the parameter λ. Weighting precision over recall means that if points are similar to each other in the output space, then they will also be similar to each other in the input space, whereas weighting recall over precision means that if points are similar in the input space, then they will also be similar in the output space. Note that the KLD and the symmetric KLD do not follow the triangle inequality for metric spaces.
The projection approach used in the Neighborhood Retrieval Visualizer (NeRV) method is randomly initialized by default, resulting in stochastic projections (see Figure 4.1). However, there exists an option to use PCA projection for initialization.

4.2 Emergent Self-Organizing Map (ESOM)

Self-organizing (feature) map (SOM) was invented by [Kohonen, 1982a, 1982b] and is a type of unsupervised neural learning algorithm. In contrast to other neural network models[20] a SOM consists of an ordered two-dimensional layer of neurons called units. Neurons are interconnected nerve cells in the human neocortex [H. Ritter et al., 1992, p. 22], and the SOM approach was inspired by somatosensory maps (e.g. see [Hennig et al., 2015, p. 421] cites [Haykin, 1994], see also [Kandel, 2012, p. 335]). There are two types of SOM algorithms: online and batch [Fort et al., 2001]. The first is stochastic, whereas the second is deterministic, which means that it yields reproducible results for a given parameter setting. However, Fort et al. have argued "that randomness could lead to better performances" [Fort et al., 2001, p. 12].
The main differences between batch-SOM [Kohonen/Somervuo, 2002] and online-SOM [Kohonen, 1995] lie in the updating and averaging of the input data. In batch-SOM, prototypes (see Eq. 4.20 below) are assigned to the data points and the influences of all associated data points are calculated simultaneously, in contrast to online-SOM, in which sequential training of the neurons is applied (as described in detail below). The batch-SOM method has been shown to produce topographic mappings of varying quality depending on the pre-defined parametrization [Fort et al., 2001], and "the representation of clusters in the data space on maps trained with batch learning is poor compared to sequential training" [Nöcker et al., 2006]. An important comparison between the batch-SOM approach and ant-based clustering was presented by [Herrmann/Ultsch, 2008c] and will be elaborated upon in chapter 7. No objective function is used in online-SOM [Lee/Verleysen, 2007, p. 241], and SOM remains a reference tool for two-dimensional visualization [Lee/Verleysen, 2007, p. 244].
In one common approach to applying the SOM concept, the algorithm acts as an extension of the k-means algorithm [Cottrell et al., 2016] or is a partitioning method of the k-means type [Murtagh/Hernández-Pajares, 1995]. In such a case, only a few units are used in the SOM algorithm to represent the data [Reutterer, 1998], which results in direct clustering of the data. Here, each neuron can be considered to represent a cluster. For example, Cottrell and de Bodt

[20] For an overview, see [H. Ritter et al., 1992], for deep learning see [Goodfellow et al., 2016].

used 4x4 units to represent the 150 data points in the Iris data set ([Ultsch et al., 2016a] cites [Cottrell, 1996]). Therefore, the conventional SOM algorithm is called k-means-SOM here. This SOM algorithm also has two common extensions called Heskes-SOM [Heskes, 1999] and Cheng-SOM; these two extensions include objective functions [Cheng, 1997] and are not discussed further in this thesis**.** The optimization of objective functions in general will be discussed in chapter 6, where it will be argued that it is not useful for the goal of this thesis. Chapter 7 will show that objective functions are incompatible with self-organization.

The other approach to applying SOM is to exploit its emergent phenomena through self-organization, in which case it is necessary to use a large number of neurons (>4000) [Ultsch, 1999]. This enhancement of the online-SOM approach is called emergent SOM (ESOM). In such a case, the neurons serve as a projection of the high-dimensional input space instead of a clustering, as is the case in k-means-SOM.

Let $M = \{m_1, \dots, m_n\}$ be the positions of neurons on a two dimensional lattice[21] (feature map) and $W = \{w(m_i) = w_i \mid i = 1, \dots n\}$ the corresponding set of weights or prototypes of neurons, then, the SOM training algorithm constructs a non-linear and topology-preserving mapping of the input space by finding the best matching unit (BMU) for each $l \in I$:

$$bmu(l) = \underset{m_i \in M}{\operatorname{argmin}}\{D(l, w_i)\}, \quad i \in \{1, \dots, n\} \qquad (4.20),$$

if in Eq. 4.20 a distance in the input space I between the point *l* and the prototype w_i is denoted. In each step, SOM learning is achieved by modifying the prototypes (weights) in a neighborhood as follows:

$$\Delta w(R) = \eta(R) * h(bmu(l), m_i, R) * (l - w(m_i)) \qquad (4.21)$$

The cooling scheme is defined by the neighborhood function $h: M \times M \times \mathbb{R}^+ \to [-1,1]$ and the learning rate $\eta: \mathbb{R}^+ \to [0,1]$, where the radius R decreases until $R = 1$ in accordance with the definition of the maximum number of epochs. In contrast to all previously introduced projection methods, no objective function is used in the ESOM algorithm. Instead, ESOM uses the concept of self-organization (see chapter 6 for further details) to find the underlying structures in data.

The structure of a (feature) map is **toroidal**; i.e., the borders of the map are cyclically connected [Ultsch, 1999], which allows the problem of neurons on borders and, consequently, boundary effects to be avoided. The positions $m \in M$ of the BMUs exhibit no structure in the input space [Ultsch, 1999]. The structure of the input data emerges only when a SOM visualization technique called U-matrix is exploited [Ultsch/Siemon, 1990].

Let $N(j)$ be the eight immediate neighbors of $m_j \in M$, let $w_j \in W$ be the corresponding prototype to m_j, then the average of all distances between prototypes w_i

$$u(j) = \frac{1}{n} \sum_{i \in N(j)} D(w(m_i), w(m_j)), n = |N(j)| \qquad (4.22)$$

A display of all U-heights in Eq. 4.22 is called a U-matrix [Ultsch/Siemon, 1990].

[21] In general this work uses the term grid if the resulting tiling is hexagonal and lattice if the resulting tiling is rectangular (see connected graph). In the context here the distinction is not important, therefore we use the term (feature) map.

"By formalizing the displayed structures, [Lötsch/Ultsch, 2014] showed that the U-matrix is an approximation of the Voronoi borders of the high-dimensional points in the output space:

Let bmu(l) and bmu(j) be the BMUs of data points l and j, where bmu(j) and bmu(l) have bordering Voronoi cells. On the borderline, there is a vertical plane (AU-height), which is the distance D(l, j) > 0 between the data points in the input space. In sum, the abstract U-matrix (AU-matrix) is the Delaunay graph of the BMUs weighted by the corresponding Euclidean distances in the input space" [Thrun et al., 2016a, p. 9].

4.2.1 Visualizations of SOMs

This section is reproduced in its entirety from [Thrun et al., 2016a]. The result of every Kohonen SOM algorithm is a set of neurons located on a map where a set W of prototypes corresponds to a set M of positions. In general, the positions on M are restricted to a grid/lattice, but a few approaches exist that change the positions in M, like Adaptive Coordinates [Merkl/Rauber, 1997]. Because these approaches are not grid/lattice based, they are not considered any further. BMUs define the locations of input points on the map. However, they exhibit no structure of the input space for a SOM [Ultsch, 1999]. However, the goal is to grasp the high-dimensional data structure and possibly even visualize cluster boundaries. Therefore, post-processing of the neurons is required for an informative representation of high-dimensional data. Three standard approaches are found in the literature:

The first approach projects the set W of prototypes with MDS [Torgerson, 1952] or some of its variants to a two-dimensional space [Kaski et al., 2000; Sarlin/Rönnqvist, 2013]. The result is mapped into the CIELab color space [Colorimetry, 2004]. In this uniform color space, perceptual differences in colors correspond to Euclidean distances in the map space as precisely as possible [Kaski et al., 2000]. The next two approaches visualize either the distances or density of the prototypes.

The second approach defines receptive fields around each position in M. The unified distance matrix (U-matrix), [Ultsch/Siemon, 1990] or one of its variants [Häkkinen/Koikkalainen, 1997; Hamel/Brown, 2011; Kraaijveld et al., 1995] , represents distances of prototypes (see equations above) by using proportional intensities of gray shades, color hues, shape or size. In [Kraaijveld et al., 1995], every neuron corresponds to a pixel. The gray value of each pixel is determined by the maximum unit distance from the neuron to its four neighbors (up, down, left, right). The larger the distance is, the lighter the gray value is. In [Häkkinen/Koikkalainen, 1997], additional unit distance visualization approaches are explained. The shapes and sizes of the receptive fields describe the dissimilarity of corresponding neurons. Apart from the U-matrix, visualizations of receptive fields in three dimensions or specific components of prototypes with receptive fields in two dimensions have been attempted [Vesanto, 1999]. It is also possible to add SOM quality measures to the receptive fields in a third dimension, e.g., [Vesanto et al., 1998].

The third approach connects the positions M by way of a specific scheme. In [Hamel/Brown, 2011], in addition to a U-matrix approach, neurons are connected with lines along the maximum gradient. The authors claim that clusters are the always-connected components of the graph defined by the U-matrix. [Merkl/Rauber, 1997] omitted the receptive fields approach, merely connecting map positions with lines, where the connection intensities reflect the similarity of the underlying prototypes. [K. Tasdemir/Merenyi, 2009] proposed the CONNvis technique, which visualizes the feature map by connecting neurons whose corresponding prototypes are adjacent in an input space with a dimensionality equal to that of the high-dimensional data. The

width of each connection line is proportional to the strength of the connection [K. Tasdemir/Merenyi, 2009].
In sum, all above described visualizations of large SOMs require an expert in the field for interpretation. To the best of the present author's knowledge, there are no 3D visualizations of ESOMs based on a 2D feature map currently in use[22].

4.2.2 Clustering with ESOM

Combining ESOM with the U*-matrix approach enables an application of [Ultsch et al., 2016a]:

> *"A single wall of AU-matrix represents the true distance information between two points in the data space. Valid density information at the midpoints between a BMU and a second BMU is calculated for [the] P-matrix, since the same volumes, i.e. spheres of a predefined radius, are used. The AU*matrix therefore represents the true distance information between two points weighted by the true density at the midpoint. The representation is such that high densities shorten the distance and low densities stretch this distance. Using transitive closure for these weighted distances allows classical clustering algorithms (AU*clustering) to actually perform distance- and density-based clustering, taking into account the complex structure of partially entwined clusters within the data."*

In contrast to the Databionic swarm approach, in which the shortest paths between AU-distances are calculated[23], this clustering approach uses only the direct neighborhood of the projected points. A computation of the abstract P-matrix is necessary because ESOM itself does not consider density. Overlaying a political map on the U*-matrix map reveals errors made by the ESOM algorithm during the annealing process. The political map shows the Voronoi areas of each cluster, where the color of each cluster area corresponds to the cluster label. The clustering is solid if every cluster consists of only one connected area, of which the borders are mountain ranges. The clustering process is sensitive to the parcel window parameter that is required for estimating the density of the high-dimensional data, and the clustering process is mostly conducted through an interactive approach requiring human intervention[24].

4.3 Types of Projection Methods

In the previous section, it was shown that projection methods such as CCA, MDS and NeRV are characterized by an objective function that is optimized using gradient descent or a corresponding learning algorithm, whereas others, such as ESOM, are not. However, the first obvious difference between types of projection methods is that between linear projection methods such as PCA or ICA and non-linear projection methods. Linear projection methods are only able to rotate the high-dimensional data space and choose the most interesting dimensions, such as the dimensions with the highest variance, as is the case for PCA.
In contrast to this approach, non-linear projection methods are able to disentangle structures, e.g., represent the Chainlink data set[25] in such a way that the two clusters are separated in the output space. The next major distinction between projection methods is the deterministic versus the stochastic approach. Some projection methods will always produce the same projection in the output space if all parameters remain unchanged. However, for many projection methods, such as t-SNE, their projections in the output space will drastically change with different trials

[22] Standard ESOM visualizations using the U-matrix are shown in supplementary D.
[23] See chapter 7 for details.
[24] For this reason, the ESOM/U-matrix clustering approach cannot be compared with other approaches in chapter 10.
[25] See the next chapter for details.

even when all settings of the projection method remain unchanged (see also examples in chapter 5, Figure 5.2). Hence, the results of deterministic methods are always reproducible, whereas stochastic methods may yield irreproducible results and require a statistical approach to assess their quality. Similarly to MDS techniques, deterministic projection methods are often based on Lyapunov functions (for further details, see [Lyapunov, 1992]). Here, it is assumed that linear and MDS techniques should only be able to visualize compact structures, which are based on the intra- versus intercluster distances of natural clusters (see the previous chapter for details).

Stochastic methods are mainly characterized by either a focusing approach or a self-organizing approach. Let k be the neighborhood extent, and let Γ be a graph; then, a projection method is of the focusing type if the result is constructed through an iterative learning process that adapts first to global neighborhoods $H(k_1 > 1, \Gamma, I)$ and later to local neighborhoods $H(k_2, \Gamma, I)$, where $k_1 > k_2$. Therefore, such methods should be capable of visualizing connected structures (see the previous chapter for details) if the annealing process is correctly chosen.

Self-organization is defined as spontaneous pattern formation by a system itself, without any central control[26] [Kelso, 1997, p. 8 ff.]. By means of self-organization, some projection methods, such as ESOM or Pswarm, are able to project data without requiring an objective function. Thus, self-organizing methods do not implicitly predefine the structures that are sought in the data of interest. The Pswarm projection method will be introduced in chapter 8 as part of the Databionic swarm clustering approach. An overview of the various types of projection methods is shown in Figure 4.1.

Assumptions regarding the types of structures that the projection methods in Figure 4.1 are able to visualize will be either disproven or verified in chapter 10 based on 100 trials per projection method (with the exception of ICA due to technical difficulties) of five artificial three-dimensional data sets.

[26] Further explained in chapter 7, p.79 ff.

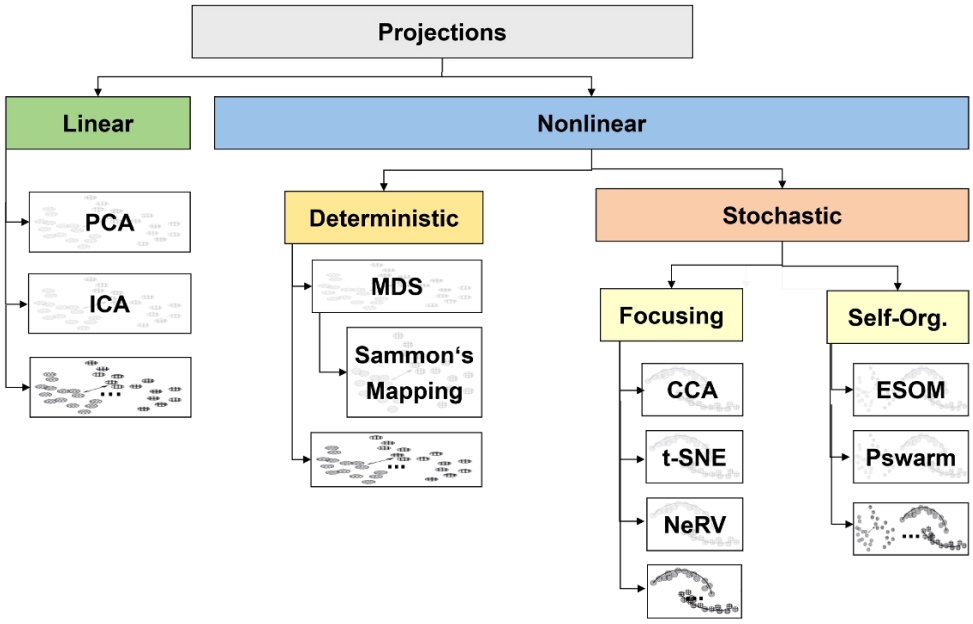

Figure 4.1: Overview of different types of projection methods. Here, it is argued that linear methods and MDS techniques are only able to visualize compact structures (shaded with the first pattern), whereas focusing projection methods should be able to visualize connected structures (shaded with the second pattern) if the annealing scheme is correctly chosen. For self-organizing methods, the structures that are sought in the data are not implicitly predefined. The ellipses indicate that this overview includes only common projection methods. Pswarm will be introduced in chapter 8 as a new approach based on swarm intelligence.

5 Visualizing the Output Space

Projection methods are a common approach to dimensionality reduction with the aim of transforming high-dimensional data into a low-dimensional space. For data visualization purposes, projections into two dimensions are considered here. However, when the output space is limited to two dimensions, the low-dimensional similarities cannot completely represent the high-dimensional distances, which can result in a misleading interpretation of the underlying structures.

Nonetheless, visualization techniques based on scatter plots produced using a projection method (usually principal component analysis (PCA)) remain the state of the art in cluster analysis (e.g., [Everitt et al., 2001, pp. 31-32; Hennig et al., 2015, pp. 119-120, 683-684; Mirkin, 2005, p. 25; G. Ritter, 2014, p. 223]). Even if one disregards that "PCA remains a rather basic method and suffers from many shortcomings" [Lee/Verleysen, 2007, p. 226], visualization based on such a scatter plot is questionable in principle. Several two-dimensional scatter plots of elementary three-dimensional data sets and one high-dimensional data set (see also Figure 6.1 in the next chapter) will be presented to illustrate this claim.

Thereafter, structure preservation will be defined in this chapter to serve as the basis for a new method of visualization. This new concept with regard to the visualization of projected points in a two-dimensional output space is called the generalized U-matrix approach. In the generalized U-matrix approach, similarities between high-dimensional data are represented as valleys, and dissimilarities are represented as mountains or ridges. For the computation of the generalized U-matrix, the generation of the topographic map (see chapter 5.3) and island visualization the CRAN R package GeneralizedUmatrix was used [Thrun/Ultsch, 2017b].

5.1 Examples

In Figure 5.1, the Hepta data set is shown. The Hepta data set [Moutarde/Ultsch, 2005] consists of 7 clusters that are clearly separated by distance, which means that the intracluster distances are small and the intercluster distances are large (for details, see chapter 9). This gives rise to structures that are clearly defined by discontinuity and consequently can be characterized as natural clusters.

Projections of the Hepta data set obtained by applying three of the projection methods introduced in the previous chapter are shown in Figure 5.2: PCA, curvilinear component analysis (CCA) and t-distributed stochastic neighbor embedding (t-SNE). In total, four projections are evaluated, including two t-SNE projections, denoted by *t-SNE (1)* and *t-SNE (2).* PCA yields the best representation of the clusters. With the default parameters, CCA adds excessive gaps around three points. In t-SNE (1), generated using the default parameter settings of t-SNE, the density of the data is overestimated, and wide gaps are added between two points and their corresponding cluster. When one parameter of the t-SNE algorithm is changed, resulting in t-SNE (2), the data clusters are not preserved because many random gaps are added.

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_5

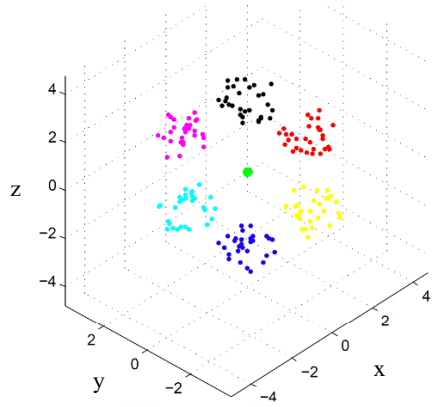

Figure 5.1: The three-dimensional Hepta data set consists of 7 clusters that are clearly separated by distance. One cluster (green) has a higher density. Every cluster is ball-like in shape.

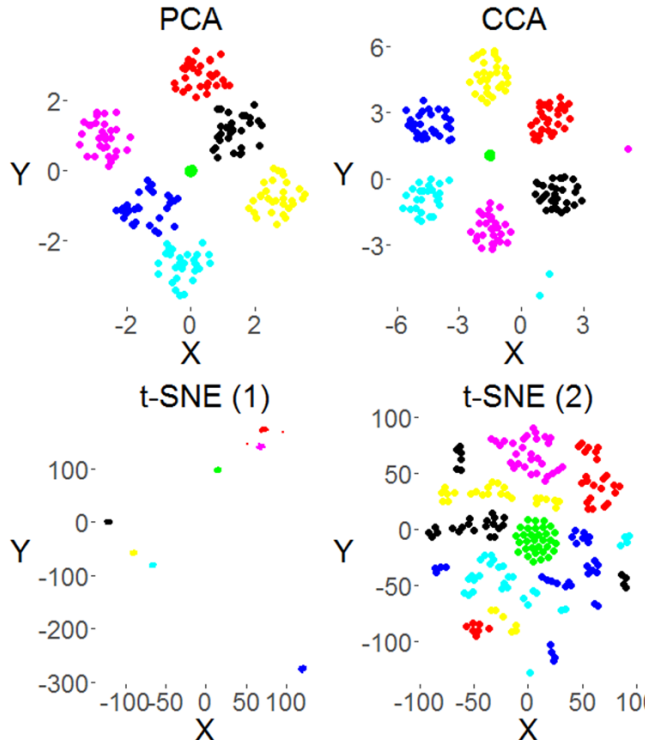

Figure 5.2: Visualizations of four cases of the projection of the Hepta data set into a two-dimensional space generated with [Thrun et al, 2017b].
Top left: PCA projects the data without disrupting any clusters. This is the best-case scenario for a projection method. **Top right**: CCA disrupts two clusters by falsely projecting 3 points. This is the standard-case scenario.
Bottom left: t-SNE does not correctly visualize the density of the data set at all, and one cluster is disrupted through the false projection of two points. Projection methods are often unable to correctly capture the density of data**. Bottom right**: When one parameter of the t-SNE algorithm is chosen incorrectly, all clusters are completely disrupted. This is the worst-case scenario for a projection method.

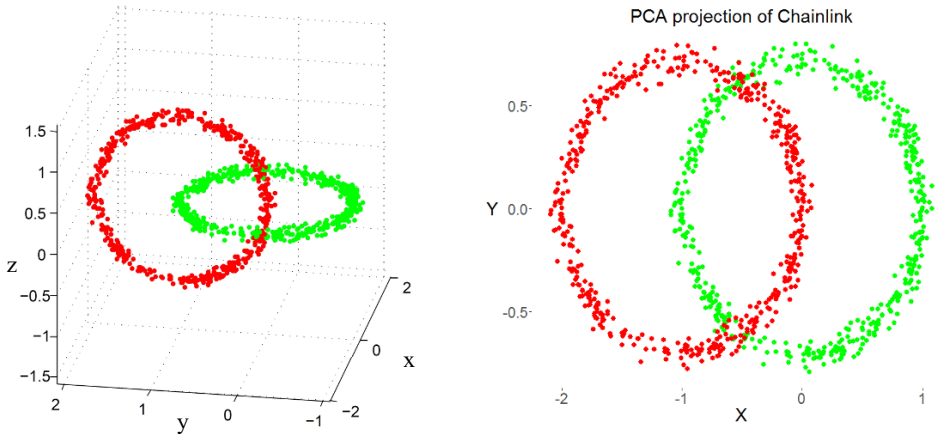

Figure 5.3: Chainlink data set and PCA projection generated with [Thrun et al., 2017]. The projection suffers from local backward projection error (BPE) and forward projection error (FPE) only in two small areas around a low number of points, but the visualization still shows low structure preservation.

The Chainlink data set [Ultsch, 2005c] consists of two clusters in $\mathbb{R}^3$. Together, both clusters form intricate links in a chain and therefore cannot be separated by linear decision boundaries. Both rings are intertwined in $\mathbb{R}^3$ and have the same average distance and density (Figure5.3 left). The data lie on two well-separated manifolds; however, the global proximities contradict the local ones in the sense that the center of each ring is closer to some elements of the other class than it is to elements of its own class (for details, see chapter 9). PCA projection completely fails to preserve the structures in this data set because PCA merely rotates the data set and the discontinuities are not linearly separable.

5.2 Structure Preservation

Let $k > 0$, $k \in \mathbb{N}$, let Γ be a connected graph, and let j be a point in a metric space M; then,

$$H_j(k, \Gamma, M) = \{l \in M | \, G(l, j, \Gamma) \leq k\} \qquad (5.1)$$

is the neighborhood set of j with k as the neighborhood extent, where $G(l, j, \Gamma)$ is the minimum distance among all possible path distances (for details, see chapter 2, Eq. 1).

Suppose that there exists a pair of similar high-dimensional data points $(l_I, j_I) \in I$ such that $(l_I, j_I) \in H(1, \Gamma, I)$. For visualization, the goal of a projection is to match these points to the low-dimensional space R^b; e.g., data points in close proximity should remain in close proximity, and remote data points should stay in remote positions.

Consequently, two kinds of errors exist. The first is forward projection error (FPE), which occurs when similar data points $l \in H_j(1, \Gamma, I)$ are mapped onto far-separated points $l \notin H_j(1, \Gamma, O) \wedge l \in H_j(k > 1, \Gamma, O)$. The second is backward projection error (BPE), which occurs when a pair of closely neighboring positions $l \in H_j(1, \Gamma, O)$ represents a pair of distant data points $l \notin H_j(1, \Gamma, I) \wedge l \in H_j(k > 1, \Gamma, I)$. It should be noted that similar definitions are found in [Ultsch/Herrmann, 2005], for the case of a Euclidean graph; in [Venna et al., 2010], for the case of a KNN graph of binary neighborhoods, where BPE and FPE are referred to as

precision and recall; and in [Aupetit, 2007], for the case of a Delaunay graph, where BPE and FPE are referred to as manifold stretching and manifold compression.
Examples of BPE and FPE are shown in Figure 5.2. The PCA projection of the Hepta data set has a low FPE but a high BPE. The CCA projection has a very low BPE, but three points have high FPEs. The t-SNE (1) projection has a very high FPE, and for the t-SNE (2) projection, both the FPE and BPE are very high.
However, the FPE and BPE are not sufficient measures for evaluating projections if the goal is to estimate the number of clusters or to ensure a sound clustering of the data (e.g., Figure 5.3 right). In such a case, a suitable projection method should be able to preserve discontinuities, which occur in regions of the data space where the probability density function becomes very small. Discontinuities divide a dataset in the input space I into several clusters of similar elements represented by points ([Ultsch/Herrmann, 2005] used a similar definition).
In summary, the quality of structure preservation should be measured based on the preservation of high-dimensional discontinuities as gaps in the two-dimensional output space. Structure preservation refers to the preservation of input-space discontinuities such that no points are allowed to intrude into the corresponding discontinuity regions in the output space.
Let $j \in I$ be an arbitrary point, and let I be projected into O by the function *proj*; then, the projection method *proj* is structure-preserving for a fixed extent $k \in \mathbb{N}$ if

$$proj\colon I \to O, H_j(k, \Gamma, I) \mapsto H_j(k, \Gamma, O)\ \forall j \in I \qquad (5.2)$$

The direct neighborhoods are preserved if

$$\forall j \in I\colon H_j(1, \Gamma, I) \cap H_j(1, \Gamma, O) = \emptyset \qquad (5.3)$$

The BPE and FPE are acceptable if the quality of structure preservation is high (e.g., Figure 5.3). Notably, the preservation of structure critically depends on the chosen concept of similarity. For example, a multidimensional scaling (MDS) technique may be a suitable projection method if the structure preservation depends only on a Euclidean graph. This is the case for the Hepta data set. By contrast, for the Chainlink data set, a KNN graph with a suitably chosen number of nearest neighbors could yield a better result.
In Chapter 6, it will be demonstrated that many quality criteria exist for evaluating visualizations. Given the definition of structure preservation, it is possible to group these quality measures (QMs) into semantic classes based on graph theory.
In the last section of this chapter, a visualization method with the specific aim of structure preservation is proposed.

5.3 Generating a Topographic Map from the Generalized U*-matrix

In this section I introduce an U*-matrix technique that is generally applicable for all projection methods and can be used to visualize both distance- and density-based structures. This visualization technique is the further development of the idea that the U-matrix can be applied to every projection method [Ultsch/Mörchen, 2006].
In this work, the visualization technique results in a topographic 3D landscape. Here, the requirements are a heavily modified emergent self-organizing map (ESOM) algorithm and a

method of high-dimensional density estimation. Contrary to [Ultsch/Mörchen, 2006], the process of computing the resulting topographic map is completely free of parameter dependence and accessible by simply by downloading the corresponding R package [Thrun/Ultsch, 2017b].

5.3.1 Simplified ESOM

To calculate a U*-matrix for any projection method, a modified ESOM algorithm is required. The first step is the computation of the correct lattice size.

On the x axis, let the lattice begin at 1 and end at a maximal number denoted by Columns C (equal to the number of columns in the lattice); similarly, on the y axis, let the lattice begin at a maximal number denoted by Lines L and end at 1. Then, the first condition is expressed as [Ultsch, 2015]

$$\frac{L-1}{C-1} \approx \frac{|\max(y)-\min(y)|}{|\max(x)-\min(x)|} = \frac{dy}{dy} = \Delta \qquad (I.)$$

The second condition is that the lattice size should be larger than NN[27]:

$$L * C \geq \mathrm{NN} \qquad (II.)$$

The first condition (I.) implies that the lattice size should be as close to equal to the size of the coordinate system as possible. The second condition (II.) is required for emergence in our algorithm. For details, see [Ultsch, 1999]. The resulting equation to be solved is

$$L^2 + L(1+\Delta) - \mathrm{NN} * \Delta \geq 0 \qquad (5.4)$$

which yields

$$L \geq -\frac{1+\Delta}{2} + \sqrt{\left(\frac{1+\Delta}{2}\right)^2 + \mathrm{NN} * \Delta} \qquad (5.5)$$

After the transformation from the projected points[28] $p \in O$ to points on a discrete lattice, the points are called the best-matching units (BMUs) $bmu \in B \subset \mathbb{R}^2$ of the high-dimensional data points j, analogous to the case for general SOM algorithms with $fgrid: O \to B, p \mapsto bmu,$ where *fgrid* is surjective when conditions (i) and (ii) are met.

To develop the algorithm illustrated in Listing 5.1, the idea of [Ultsch/Mörchen, 2006], in which it was suggested to "apply Self-Organizing Map training without changing the best match[ing unit] assignment", was adopted. However, in contrast to [Ultsch/Mörchen, 2006], here, the transformation *fgrid* is defined precisely to calculate the BMU positions and the structure of the lattice is toroidal; i.e., the borders of the lattice are cyclically connected [Ultsch, 1999].

Based on the relevant *symmetry considerations*[29], a simplified version of ESOM (sESOM) is introduced here. No epochs or learning rate are required, because the cooling scheme is defined by a special neighborhood function h: $\mathrm{M} \times \mathrm{M} \times \mathbb{R}^+ \to [0,1]$.

Let $M = \{m_1, \ldots, m_n\}$ be a set of neurons (where m_i are the lattice positions) with the corresponding prototype set $W = \{w_1, \ldots, w_n\}$, where dim(W)=dim(I) and #W=#M; then, the neighborhood function h is defined as

[27] In [Ultsch, 1999] the minimum number of 4096 neuros was proposed.
[28] Or DataBot positions on the hexagonal grid of Pswarm (see chapter 8).
[29] See chapter 8 for details.

$$h = \begin{cases} 1 - \frac{d(j,l)^2}{\pi R^2}, & \text{iff } \frac{d(j,l)^2}{\pi R^2} < 1 \\ 0, & \text{else} \end{cases} \qquad (5.6)$$

In sESOM, learning is achieved in each step by modifying the weights in a neighborhood as follows:

$$\Delta w(R) = 1 * h(bmu(j), m_i, R) * (j - w(m_i)) \qquad (5.7)$$

In contrast to [Ultsch/Mörchen, 2006], the algorithm does not require any input parameters, and the resulting visualization is not a two-dimensional gray-scale map but rather a topographic map with hypsometric tints [Thrun et al., 2016a]. The entire algorithm is summarized in Listing 5.1.

```
function (B, I)
    for all bmu(j)∈ B:
        assign the positions m_j ∈ M with random weightings w_j∈ W on the grid
        assign to each bmu(j) = m_j the weighting w_j = j ∈ I
    end for bmu(j)
    for R=Rmax to 1 do
        for all j∈ I:
            bmu(j) = argmin_{m∈M} {D(j, w(m))}
                Δw(R, bmu(j)) = h(bmu(j), m_i, R) * (j − w(m_i))
                for all w(m_k) ∈ h(bmu(l), m_i, R)
                    w´(m_k) = w(m_k) + Δw(R,bmu(l))
                end for w(m_k)
        end for j∈ I
        for all bmu(j)∈ B:
            assign to each bmu(j) = m_j the weighting w_j = j ∈ I
    end for R
end function
```

Listing 5.1: sESOM pseudocode algorithm implements a stepwise iteration from the maximum radius Rmax which is given by the lattice size (Rmax = C/6) stepwise with one per step and down to 1. $w´(m_k)$ indicates that the prototype $w(m_k)$ of neuron m_k is modified by Eq. 5.7
Additionally, the search for a new best matching unit still is used and these prototypes may change during one iteration. The predefined prototypes are reset to the weights of their corresponding high-dimensional data points after each iteration.

5.3.2 U-Matrix Calculation*

After sESOM projection, the structure of the input data emerges when a visualization technique called U-matrix is applied. A U-matrix represents a folding of the high-dimensional space in which each receptive field is called a U-height. Let *N(j)* be the eight immediate neighbors of $m_j \in M$, and let $w_j \in W$ be the prototype corresponding to m_j; then, the average of all distances between w_j and the other prototypes w_i is called the U-height corresponding to the position m_j:

$$u(j) = \frac{1}{n} \sum_{i \in N(j)} D(w_i, w_j), \qquad n = |N(j)| \qquad (5.8)$$

To explain the visualization technique for the sESOM algorithm, in this section and in section 5.3.3 below, [Thrun et al., 2016a] is cited:

> *"The U-matrix is the display of values* $u(j)$ *through proportional intensities of grey shades [Ultsch, 2003a]. By formalizing the displayed structures, [Lötsch/Ultsch, 2014] showed that the U-matrix is an approximation of [the] Voronoi borders of the high-dimensional points in the output space" (see chapter 4.2.0).*

Therefore, the generalized U-matrix can be normalized [using] the generalized abstract U-matrix.

> *"In addition to the U-matrix, [Ultsch, 2003c] introduced the high-dimensional density visualization technique called P-matrix, where P-heights on top of the receptive fields are displayed. The P-height* $p(m_i)$ *for a position* m_i *is a measure of the density of data points in the vicinity of* $w(m_j)$:
>
> $$p(m_j) = |\{i \in I | D(i, w(m_j)) < r > 0, r \in \mathbb{R}\}| \qquad (5.9).$$
>
> *The P-height is the number of data points within a hypersphere of radius r. Here, we choose the interval* ϱ *of the radius with*
>
> $$\varrho \in [median(C(D)), median(A(D))], \qquad (5.10)$$
>
> *where D [represents] all input space distances and A(D) is the group A of distances calculated by [the] ABC analysis [Ultsch/Lötsch, 2015]. ABC analysis[30] tries to identify the optimum information that can be validly retrieved by using concepts developed in economic sciences. In particular, [these] concepts are used in the search for a minimum possible effort that gives the maximum yield [Ultsch/Lötsch, 2015]. The distances are divided into three disjoint subsets A, B and C, with subset A comprising [the] largest values ("outer cluster distances"), subset B comprising values where the yield equals the effort required to obtain it, and the subset C comprising [] the smallest values ("inner cluster distances"). We suggest [choosing] the specific radius r [based on] the [ratio] v of [the] inter- versus intracluster distances[,] estimated [as]*
>
> $$v = \frac{max(C(D))}{min(A(D))} \qquad (5.11)$$
>
> *The radius r is estimated [as]* $r = v * p20(D)$*, where* $p20(D)$ *is [the] 20-th percentile of [the] input distances [Ultsch, 2003b]. From this starting point, the user may search interactively for the empirical Pareto percentile [that] defines the radius r (see [the] R package Umatrix).*
>
> *The combination of a U-matrix and a P-matrix is called [a] U*-matrix [Ultsch et al., 2016a]. It can be formalized as [a] pointwise matrix [product]:* $U^* = U * F(P)$*, where F(P) is a matrix of factors f(p) that are determined through a linear function f on the P-heights p [in] the P-matrix. The function f is calculated so that f(p) = 1 if p is equal to the median and f(p) = 0 if p is equal to the 95-[th] percentile (p95) of the heights in the P-matrix. For p(j) > p95, f(p) = 0, which indicates that j is well within a cluster and results in [a height of zero] in the U*-matrix." [Thrun et al., 2016a]*

5.3.3 Topographic Map with Hypsometric Tints

The U*-matrix visualization technique produces a topographic map with hypsometric tints [Thrun et al., 2016a]. Hypsometric tints are surface colors that represent ranges of elevation [Patterson/Kelso, 2004]. Here, a specific color scale is combined with contour lines.

The color scale is chosen to display various valleys, ridges and basins: blue colors indicate small distances (sea level), green and brown colors indicate middle distances (low hills), and white colors indicate large distances (high mountains covered with snow and ice). Valleys and

[30] For usage see CRAN R package ABCanalysis [Thrun et al. 2015].

basins represent clusters, and the watersheds of hills and mountains represent the borders between clusters (Figure 5.1 and Figure 5.4).

The landscape consists of receptive fields, which correspond to certain U*-height intervals with edges delineated by contours. This work proposes the following approach (see [Thrun et al., 2016a, p. 10]): First, the range of U*-heights is split up into intervals, which are assigned uniformly and continuously to the color scale described above through robust normalization [Milligan/Cooper, 1988]. In the next step, the color scale is interpolated based on the corresponding CIELab color space [Colorimetry, 2004]. The largest possible contiguous areas corresponding to receptive fields in the same U*-height intervals are outlined in black to form contours. Consequently, a receptive field corresponds to one color displayed in one particular location in the U*-matrix visualization within a height-dependent contour. Let u(j) denote the U*-heights, and let q01 and q99 denote the first and 99-th percentiles, respectively, of the U*-heights; then, the robust normalization of the U*-heights u(j) is defined by

$$u(j) = \frac{u(j) - q01}{q99 - q01} \qquad (5.12)$$

The number of intervals in is defined by

$$\frac{1}{in} = \frac{q01}{q99} \qquad (5.13)$$

The resulting visualization consists of a hierarchy of areas of different height levels represented by corresponding colors (see Figure 5.4). To the human eye, the visualization using the generalized U-matrix tool is analogous to a topographic map; therefore, one can visually interpret the presented data structures in an intuitive manner. In contrast to other SOM visualizations, e.g., [K. Tasdemir/Merenyi, 2009], this topographic map presentation enables the layman to interpret sESOM results.

The use of a toroidal map for sESOM computations necessitates a tiled landscape display in the interactive U-matrix tool [Thrun et al., 2015], which means that every receptive field is shown four times. Consequently, in the first step, the visualization consists of four adjoining images of the same U-matrix [Ultsch, 2003a] (the same is true for the U*-matrix). To obtain the 3D landscape (island[31]), [Thrun et al., 2016a, p. 10] proposed to rectangularly cut the tiled U*-matrix visualization as follows.

Let v_{Lines} and $v_{Columns}$ be the vectors of the row and column sums, respectively, of the U*-heights, and let b_{Lines} ($b_{Columns}$) be the number of BMUs in the corresponding row line of v_{Lines} ($v_{Columns}$); then, we define the upper border as up $= \max(v_{Lines}/f(b_{Lines}))$, the left border as lb $= \max(b_{Columns}/f(v_{Columns}))$ and the other two borders based on the length and width of the U*-matrix, where the vector *f(b)* is the sum $f(b) = \hat{b} + b + \check{b}$ with $\hat{b} = (b_n, b_1, \dots, b_{n-1})$ and $\check{b} = (b_2, \dots, b_{n+1})$ for a toroidal lattice. For better comprehensibility, see the axes in [Thrun et al., 2016a, p. 14, Fig. 1], which are defined from one to $max(Lines)$ and from one to $max(Columns)$.

[31] An island can be also cut interactively (or the the cutting may be improved) and thus may not be rectangular

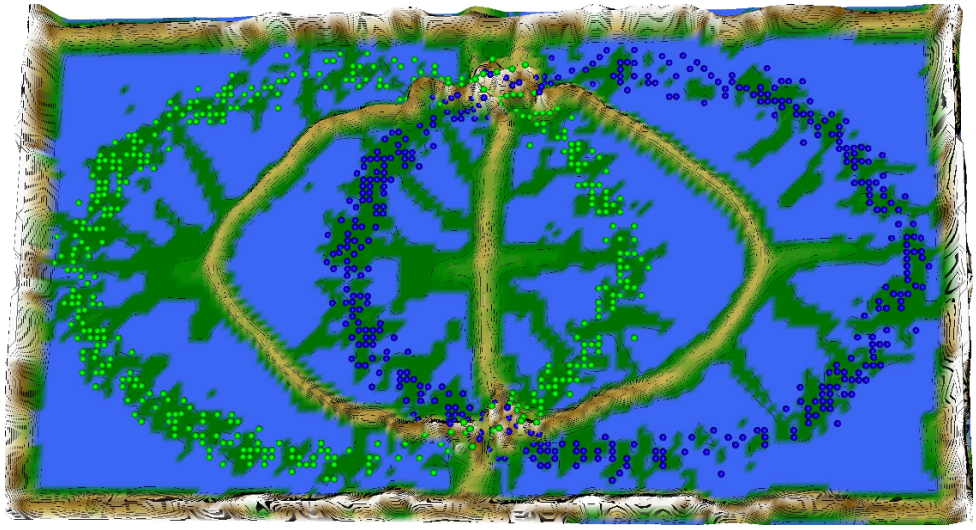

Figure 5.4: Topographic map of the PCA projection of the Chainlink data set. The discontinuities between the clusters are misrepresented.

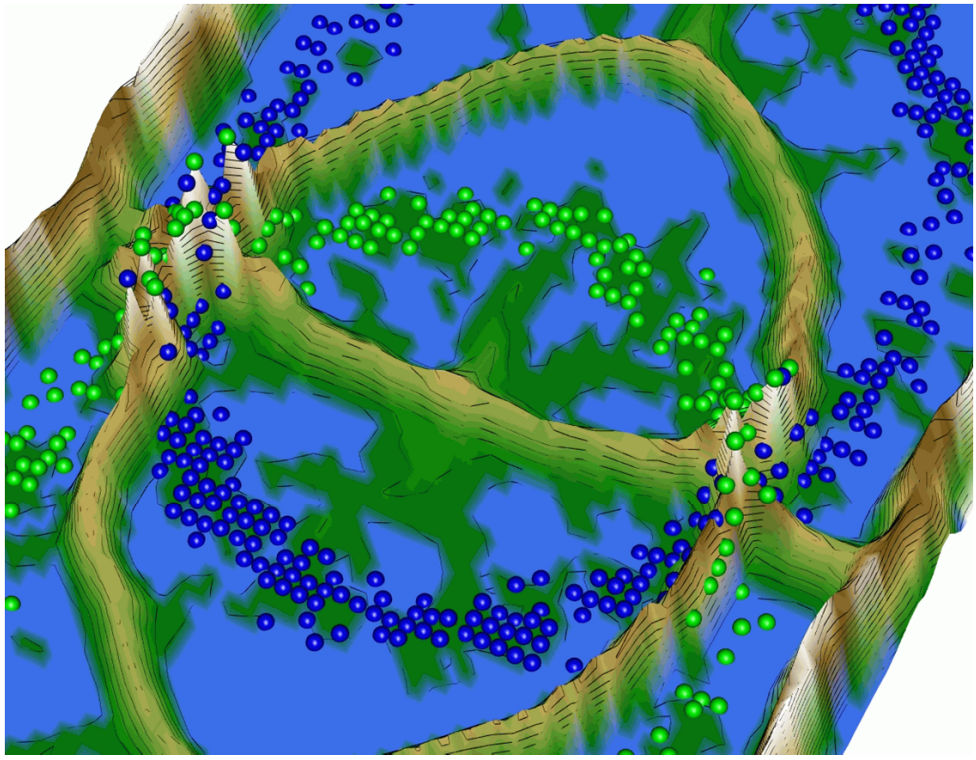

Figure 5.5: Zoomed-in view of the misrepresentation of the discontinuities in the PCA projection of the Chainlink data set to better visualize the BPE and FPE.

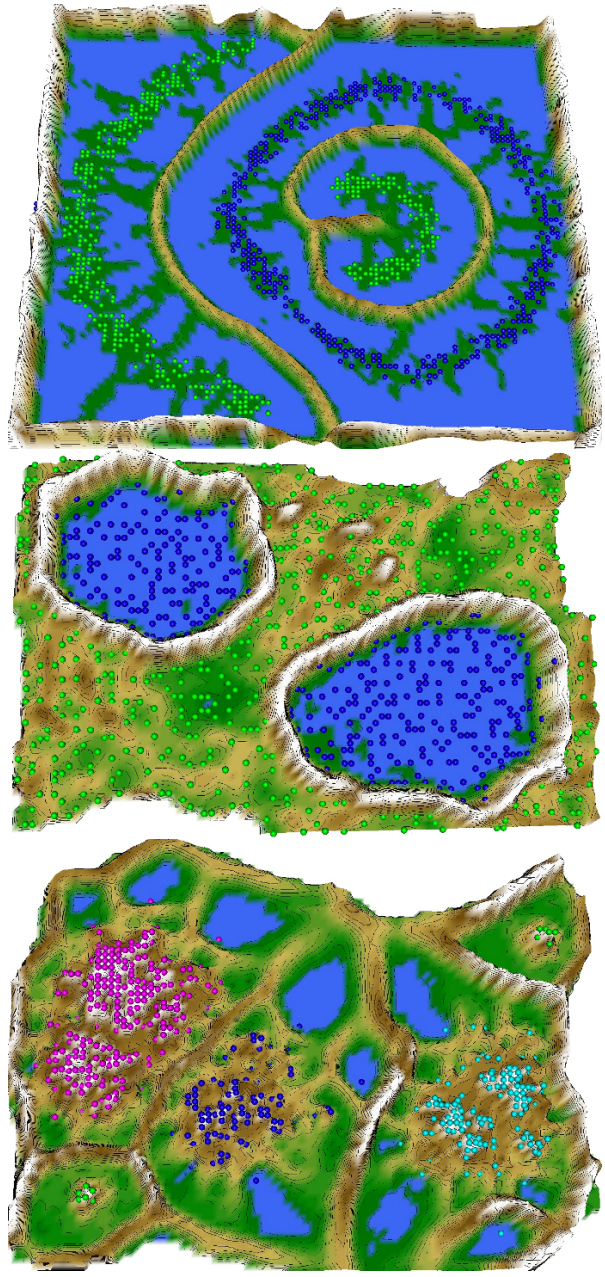

Figure 5.6: Topographic maps can depict the discontinuities in high-dimensional data sets: clusters lie in valleys and are separated by hills. However, the introduction of spurious gaps between projected points (the disruption of clusters) cannot be seen using this approach.
Top: topographic map of CCA projection [Demartines/Hérault, 1995] of the Chainlink data set.
Middle: topographic map of ESOM projection [Ultsch, 1999] of the Atom data set.
Bottom: island of NeRV projection [Venna et al., 2010] of the leukemia data set. All results are trial-dependent because the projection methods are stochastic. Sometimes, the annealing scheme (in CCA or ESOM) or the random initialization process (in NeRV) fails.

5.3.4 Limitations

The generalized U*-matrix visualization by a topographic map is capable of visualizing BPEs and FPEs. For example, this is shown in Figure 5.5. The projected points in the output space with low BPE/FPE values lie in sea regions. If the BPE/FPE around a projected point is high, then the visualization generates a mountain at this point (Figure 5.5). However, the topographic map has certain limitations (Figure 5.6). When the default parameters in CCA are used to analyze the Chainlink data set (see [Thrun et al., 2017]) or when the default ESOM parameters ([Thrun et al., 2016b]) are used to analyze the Atom data set, clusters are sometimes disrupted because additional gaps are added that cause points to intrude into the discontinuity regions between clusters.

Another question that arises in this chapter from the examples of the CCA and ESOM projections of the Chainlink and Atom data sets, respectively, in Figure 5.6 is the question of how to handle stochastic projection methods in which the visualization is trial-dependent. The annealing schemes used in the ESOM and CCA algorithms may be relevant here. The annealing process depends on certain parameters and may not yield structure-preserving projections, as shown in the examples in Figure 5.6 The Neighborhood Retrieval Visualizer (NeRV) projection of the leukemia data set presented in Figure 5.6 further illustrates the problem of the correct choice of parameters, which is typically very challenging. In this case, the NeRV projection is sensitive to the initialization parameters, especially to the seed used for the random number generator. In chapter 9, an additional example will be presented to demonstrate that NeRV requires the weighting between precision and recall to be correctly chosen for high-dimensional structures to be preserved.

Hence, the next chapter will focus on the search for a QM that may be able to measure structure preservation instead of attempting to visualize it.

6 Quality Assessments of Visualizations

Dimensionality reduction techniques reduce the dimensions of the input space to facilitate the exploration of structures in high-dimensional data. Two general dimensionality reduction approaches exist: manifold learning and projection. Manifold learning methods attempt to find sub-spaces in which the high-dimensional distances are preserved. Usually, these sub-spaces have more than two dimensions.

It was argued in [Venna et al., 2010] that manifold learning methods are not very useful for information visualization because they are designed simply to find a manifold, and L. J. van der Maaten et al. demonstrated that they do not outperform classical principal component analysis (PCA) for real-world tasks [L. J. van der Maaten et al., 2009].

This work focuses on two-dimensional visualizations of high-dimensional data, with the intention of making the visualizations easily understandable, because it is difficult for humans to get a spatial sense of more than three dimensions. A valid visualization is possible if a projection method creates an image of the structure of high-dimensional data. The two-dimensional scatter plot remains a state-of-the-art form of visualization used in cluster analysis (e.g., [Everitt et al., 2001, pp. 31-32; Hennig et al., 2015, pp. 119-120, 683-684; Mirkin, 2005, p. 25; G. Ritter, 2014, p. 223]). Consequently, the aim here is to evaluate two-dimensional visualizations of high-dimensional data in which the structures are defined by discontinuities. In short, projection methods should preserve the structures defined by natural clusters.

However, as a consequence of limiting the output space to two dimensions, the low-dimensional similarities cannot completely represent the high-dimensional distances, which can result in a misleading interpretation of the underlying structures; these structures can be evaluated using quality measures (QMs), and the first step in the process of assessing the performance of projection methods is to assess these measures themselves. Here, the QMs are assessed based on the proposed concept of structure preservation, namely, the preservation of high-dimensional discontinuities related to compact or connected structures (see chapter 3, section 3.2.1, for details). Overall, 19 QMs will be categorized into semantic groups in this chapter, and their advantages and disadvantages will be discussed.

To date, QMs have mostly been applied to data sets such as a Swiss roll shape [L. Van der Maaten et al., 2009] [Mokbel et al., 2013], an s-shape [Yin, 2007] or a sphere [Venna et al., 2010], for which the problem lies only in the visual representation of an object that is continuous in more than two dimensions. Recently, [Gracia et al.] conducted a study on a number of QMs based on 12 real-world data sets. The research team's analysis of the QMs concentrated on the correlations between them [Gracia et al., 2014]. This study illustrates the other common evaluation approach: the use of various natural high-dimensional data sets for which prior classifications are available. However, with the exception of the classification error (CE) (see section 2), this information is not used in the evaluation of projection methods, e.g., [Bunte et al., 2012]. Moreover, it is not stated whether the classification is defined based on discontinuities or the prior knowledge of a domain expert. Whether these data sets possess discontinuities is not discussed.

Serving as an illustration of this problem, Figure 6.1 presents projections of a high-dimensional data set called the leukemia data set. In addition, above each plot in Figure 6.1, the CE for 7

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_6

nearest neighbors is provided. The leukemia data set was introduced in chapter 3, where it was shown that common clustering algorithms are unable to reproduce its prior classification.
The question arises of whether the existing QMs are able to distinguish among different projections with regard to their preservation of the discontinuities in this data set (see chapter 3.3, Figure 3.6 and 3.7). As an example, Figure 6.2 shows the often used trustworthiness and discontinuity (T&D) measures [Venna/Kaski, 2001] and precision and recall measures [Venna et al., 2010] for this data set. The distinction among the six projections in terms of quality, based on these measures, is debatable.

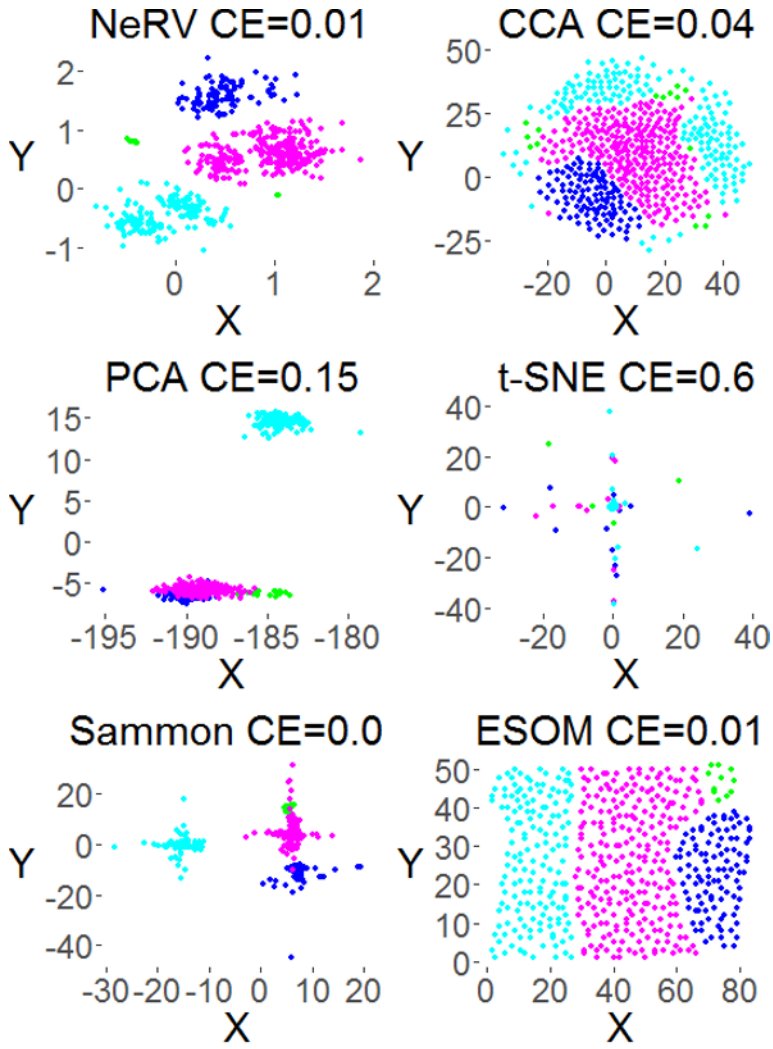

Figure 6.1: Projections of the leukemia data set generated using common methods and the corresponding classification errors (CEs, see 6.1.1 for def.) for 7 nearest neighbors CE(k=7). The colors represent the predefined illness cluster labels. The clusters are separated by discontinuities in the high-dimensional space (see chapter 3). Emergent self-organizing map (ESOM) is the projection method that best preserves the discontinuities in this data set. The Neighborhood Retrieval Visualizer (NeRV) algorithm splits the smallest cluster into two roughly equal parts.

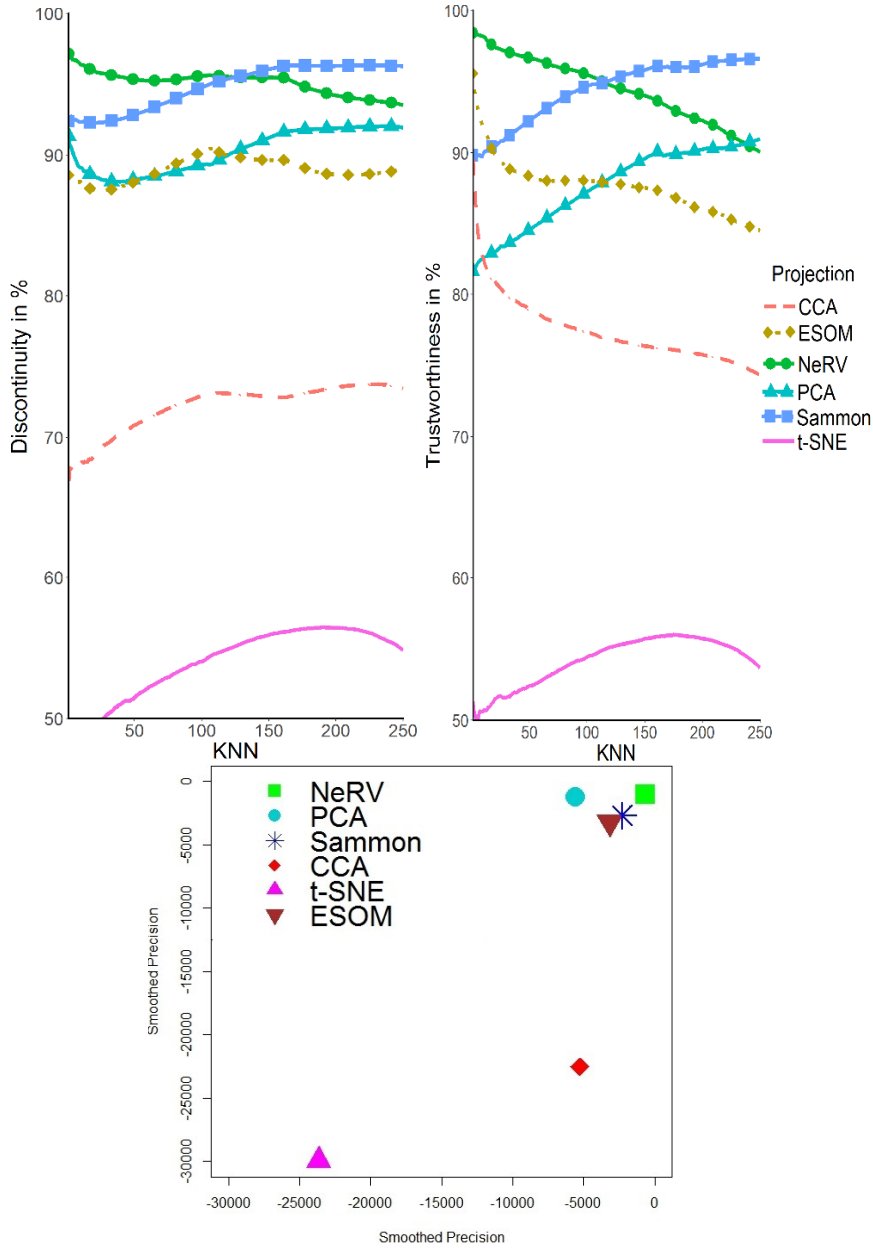

Figure 6.2: Trustworthiness and discontinuity (T&D) measures (def. see 6.1.13 on p. 65) and precision and recall measures (def. see 6.1.8 on p. 68) for the six projections shown in Figure 6.1 of the leukemia data set. The discontinuity is highest for Sammon mapping and NeRV (top left), as is the trustworthiness (top right). However, in the case of the trustworthiness, the outcome depends on the number of nearest neighbors considered, k; for a low value, ESOM is superior to Sammon mapping, and for a high value, principal component analysis (PCA) overtakes NeRV. In terms of the smoothed precision and recall [Venna et al., 2010], NeRV and PCA achieve the best values. Without the scatter plots in Figure 6.1, interpretation of the results of this figure is difficult.

This example illustrates that the evaluation of projections of real-world, high-dimensional data sets, and consequently the evaluation of QMs, is a challenging task. To simplify the problem, two elementary artificial three-dimensional data set[32] will be used to aid in assessing QMs (results in supplement A). Both data sets are clearly defined based on the discontinuities, which some projection methods fail to project into two dimensions (see supplement A). In the second section of this chapter, the definitions of neighborhoods from the perspective of graph theory (chapter 2) will enable a deeper understanding of the various types of QMs.
In the last section, a new QM called the Delaunay classification error (DCE) will be introduced, which requires a prior classification of the data set of interest and is inspired by recent SOM research [Lötsch/Ultsch, 2014] on the structures of the U-matrix. In the previous chapter, a method that allows the U-matrix to be computed for any projection method was proposed.

6.1 Common Quality Measures (QMs)

In this section, the well-known measures for assessing the quality of projections are introduced in alphabetical order. Some QMs use the ranks of distances $R(j,l)$) instead of the actual distances $D(j,l)$ between points. In this case, the following shorthand notation will be used.
Let $D(j,l)$ be an entry in the matrix $D_{N\times N}$ of the distances between all *N* points in a metric space *M*, where $j,l \in M$; then, the rank $R(D(j,l)) = y \in \{1,\dots,n\}$ denotes the y^{th} position in the consecutive sequence of all entries of this matrix arranged in value from smallest to greatest. In short, the ranks of the distances are the relative positions of the distances, where *R* denotes the ranks of the distances in the input space and *r* denotes the ranks of the distances in the output space. Occasionally, ranks are represented by a vector in which the entries are the ranks of the distances between one specific point and all other points. Typically, the matrix or vector of ranks is normalized such that the values of its entries lie between zero and one.

6.1.1 Classification Error (CE)

This type of error is often used to compare projection methods when a prior classification is given [Bunte et al., 2012; Gracia et al., 2014; L. J. van der Maaten et al., 2009; Venna et al., 2010].
Each point $l \in O$ in the output space is classified by a majority vote among its k nearest neighbors in the visualization [Venna et al., 2010], although sometimes simply the cluster of the nearest neighbor is chosen. This classification is compared with the prior classification as follows: Let $c \in C$ denote the classification of the points $j \in I$ in the input space, where $C_k(I)$ denotes a cluster of the classification in I. Let $l \in O$ denote the projected points in the output space that map to I. Let $H_j(\text{knn}, K, O)$ be the neighborhood of j in a KNN graph in the output space. Then, the clusters are sorted and the clusters with the largest number of points is chosen: If $\{l \in H_j(\text{knn}, K, O) | \forall\, l_1, \dots, l_{knn}, |C_{k_1}(l)| < |C_{k_2}(l)| < \dots < |C_{k_k}(l)|\}$, then $C_j(O) = \{C_{k_k}(l)\}$. The label $C_j(O)$ is then compared with $C_j(I)$. This yields the error

$$F = \frac{1}{N}\sum_{j=1}^{N} |C_j(O) \neq C_j(I)| \qquad (6.1)$$

[32] One with compact structures, one with connected structures.

6.1.2 C Measure

The C measure is a product of the input and output spaces in terms of similarity functions [Goodhill et al., 1995]. For ease of comparison, in (6.4), the similarity function is redefined as the distance between two points. Consequently, the C measure is defined based on a Euclidean graph.
In the equation below, C is replaced with the capital letter F.

$$F = \sum_{j} \sum_{l} D(j,l) \cdot d(j,l) \qquad (6.2)$$

A high value of the C measure indicates good neighborhood preservation. It is evident from Eq. 6.2 that F is at a maximum when the ranks of the distances in the spaces I and O are equivalent. No normalization of the F value is given.

6.1.3 Two Variants of the C Measure: Minimal Path Length and Minimal Wiring

Eq. 6.3 presents the definition of the minimal path length [Durbin/Mitchison, 1990], and Eq. 6.4 gives the definition of the minimal wiring [Mitchison, 1995]:

$$F = \sum_{j,l} D(j,l) \cdot s(j,l) \qquad (6.3)$$

$$F = \sum_{j,l} d(j,l) \cdot s(j,l) \qquad (6.4)$$

Where Eq. (*I)* with $s(k,j)$ defines the k nearest neighbors. Thus, it is analogous to a KNN graph**:**

$$s(j,l) = \begin{cases} 1, & if\ j \in H(knn = 1, M) \\ 0, & otherwise \end{cases} \qquad (I)$$

where in (Eq. 6.3), M=I to define the set of the nearest spatial neighbors in the input space I, and in (Eq 6.4), $M = O$ to serve the same purpose for the output space. A smaller value of the error F indicates a better projection.

6.1.4 Force Approach Error

According to the force approach concept presented in [Tejada et al., 2003], the relation between the distances $D(j,l)$ and d(j, l) should be constant for each pair of adjacent data points. The force approach attempts to separate data points that are projected too close to one another and to bring together those that are too scattered. In [Tejada et al., 2003], it was suggested that it is possible to improve any projection method by the following means.
First, for each pair of projected points (w_j, w_l), the vector $\overrightarrow{v_{jl}} = w_j - w_l$ is calculated if w_j is a direct neighbor of w_l; then, a perturbation in the direction of $\overrightarrow{v_{jl}}$ is applied. Consequently, w_j is moved in the direction of $\overrightarrow{v_{jk}}$ by the fraction defined in (5a). When all points w_j have thus been improved, a new iteration begins.

$$\Delta_l = \frac{D(j,l) - D_{min}}{D_{max} - D_{min}} - \mathrm{d}'(\mathrm{j},\mathrm{l}) \qquad (6.5')$$

Note that all distances $D(j,l)$ are normalized only once. For performance reasons, the projected points are normalized in every iteration instead of the d(j, l). The error on the projected points is defined as

$$F = \frac{1}{M}\sum_{l=1}^{N}|\Delta_l| \qquad (6.5)$$

Thus, as shown in Eq. 6.5′, the force approach error is defined with respect to a Euclidean graph, and an F value of zero suggests optimal neighborhood preservation, as seen from Eq. 6.5. A similar approach, referred to as point compression and point stretching, was proposed in [Aupetit, 2007], where it was used for the visualization of errors with the aid of Voronoi cells.

6.1.5 König's Measure

König's measure is a rank-based measure introduced in [König et al., 1994]:

$$F(knn) = \frac{1}{3knn * N}\sum_{j=1}^{N} q_c(j, knn) \qquad (6.6)$$

with $q_c$as in Eq. I

$$q_c(j,knn) = \begin{cases} 3, & if\ R(j,l) = r(j,l)\ and\ \ l \in H_j(knn,I) \cap H_l(knn,O) \\ 2, & if\ l \in H_j(knn,I) \cap H_j(knn,O) \\ 1, & if\ l \in H_j(knn,I) \cap H_j(c,O), knn < c \\ 0, & otherwise \end{cases} \qquad (I)$$

König's measure is controlled by the following parameters: a constant parameter c and a variable parameter representing the neighborhood size, $knn \in \{1,..,knn|knn < c\}$, which must be smaller than c.

In the first case, the ranks place l in the same knn neighborhood with respect to j in both the input and output spaces. In the second case, the sequence in the neighborhood may be different, but $l \in O$ is still within the first knn ranks relative to j in the current neighborhood defined by the value of knn. In the third case, the point l lies in a larger, constant neighborhood of $H_j(c,O)$. The range of F is between zero and one, where a value of one indicates perfect structure preservation and a value of zero indicates poor structure preservation [König, 2000]. The parameters knn and c were investigated by [Karbauskaitė/Dzemyda, 2009]. The results indicated that c does not have a strong influence on the value of F; F changes only for large knn values. Moreover, [Karbauskaitė/Dzemyda, 2009] showed that the parameter k_1 influences only the magnitude of the F value, whereas the form of F(knn) remains approximately the same.

6.1.6 Local Continuity Meta-Criterion (LCMC)

The local continuity meta-criterion (LCMC) was introduced in [Chen/Buja, 2006]; note that a similar idea was independently adopted by [Akkucuk/Carroll, 2006]. Because the correlation between these two measures is very high [Gracia et al., 2014]), only the LCMC is introduced here. The LCMC is defined as the average size of the overlap between neighborhoods consisting of k nearest neighbors in I and O [Chen/Buja, 2009]. For each $x_j \in I$ and $w_j \in O$, there exist corresponding sets of points in the neighborhoods $H(knn,I)$ and $H(knn,O)$, which are calculated using a given knn in a KNN graph. The overlap is measured in a pointwise manner:

$$A(j) = \left| H_j(knn, I) \cap H_j(knn, O) \right|, \qquad \overline{A_{knn}} = \frac{1}{N}\sum_{j=1}^{N} A(j) \qquad (6.7')$$

In Eq. 6.7′, a global measure is obtained by averaging all N cases [Chen/Buja, 2009]. The mean $\overline{A_{knn}}$ is normalized with respect to *knn* because this value is the upper bound on $\overline{A_{knn}}$. Eq. 6.7 is also adjusted by means of a baseline term representing a random neighborhood overlap, which is obtained by modeling a hypergeometric distribution with *knn* defectives out of N-1 items, from which *knn* items are drawn:

$$F(knn) = \frac{1}{knn}\overline{A_{knn}} - \frac{knn}{N-1} \qquad (6.7)$$

In contrast to the T&D measures and the mean relative rank error (MRRE; see the next section), the LCMC is calculated based on desired behavior [Lee/Verleysen, 2009]. The cited authors also showed that the LCMC can be expressed as a special case of the co-ranking matrix.

6.1.7 Mean Relative Rank Error (MRRE) and the Co-ranking Matrix

The MRRE was introduced in [Lee/Verleysen, 2007, p. 214] and is defined as follows:

$$F_1(knn) = \frac{1}{N(knn)} * \sum_j \sum_{l \in \mathrm{H(knn,O)}} \frac{|R(j,l) - r(\mathrm{j,l})|}{R(j,l)} \qquad (6.8a)$$

$$F_2(knn) = \frac{1}{N(knn)} * \sum_j \sum_{j \in \mathrm{H(knn,I)}} \frac{|R(j,l) - r(\mathrm{j,l})|}{r(j,l)} \qquad (6.8b)$$

The normalization is given by $N(knn) = N\sum_{n=1}^{knn} \frac{|N-2n+1|}{n}$, which represents the worst case. There are notable similarities between the MRRE and the T&D measures: both types of measures use the ranks of the distances and KNN graphs to calculate overlaps, but, in addition to the different weightings, the MRRE also measures changes in the order of positions in a neighborhood *H(knn, I)* or *H(knn, O)*. Both position changes and intruding/extruding points are considered, but position changes are weighted more heavily than intrusion/extrusion. The MRRE (and T&D and LCMC, as well) can be abstracted using the co-ranking matrix framework as follows.

As introduced in [Lee/Verleysen, 2008], $Q = q_{ik,1\le i,k\le N-1}$ is a matrix in which each element is equal to the number of pairs of points that lie in neighborhoods defined by the same or different values of knn. For example, $q_{ik} = \left| H(i, knn, I) \cap H(k, knn, O) \right|$ represents the upper left block of the co-ranking matrix for a specific knn. Formally, Q is a sum of N permutation matrices; hence, $\sum_{i=1}^{N-1} q_{ik} = \sum_{k=1}^{N-1} q_{ik} = N$. It was shown in [Lee/Verleysen, 2009] that the MRRE can be rewritten as two alternative quantities characterizing a projection

$Q_{MRRE}(K) = 1 - \frac{F_1+F_2}{2}$, which the authors call the quality of the projection, and

$B_{MRRE}(K) = F_1 - F_2$, called the behavior (for details, see [Lee/Verleysen, 2009]).

6.1.8 Precision and Recall

[Venna et al., 2010] reintroduced the idea of misses used by [Ultsch/Herrmann, 2005], where misses are similar data points $(l_I, j_I) \in i$ that are mapped to far-separated points $(\mathrm{l}_O, \mathrm{j}_O) \in O$

[Ultsch/Herrmann, 2005]. Conversely, if a pair of closely neighboring positions (l_O, j_O) represents a pair of distant data points, then this pair is called a false positive. From the information retrieval perspective, this approach allows one to define the precision and recall for the case in which the neighborhoods are merely binary. However, [Venna et al., 2010] goes a step further by replacing such binary neighborhoods with probabilistic ones, which are loosely inspired by stochastic neighbor embedding [Hinton/Roweis, 2002]. The neighborhood of the point l is defined with respect to the relevance of the points $j \in I$ around l:

$$p_l(j) = \frac{exp(-\frac{D(l,j)^2}{\sigma_l^2})}{\sum_{k \neq j} exp(-\frac{D(l,k)^2}{\sigma_l^2})} \qquad (I)$$

where σ_l is set to the value for which the entropy of $p_l(j)$ is equal to log(knn) and knn is a rough upper limit on the number of relevant neighbors and is set by the user [Venna et al., 2010]. The authors propose a default value of 20 effective nearest neighbors. Similarly, the corresponding neighborhood in the output space is defined as

$$q_l(j) = \frac{\exp\left(-\frac{d(l,j)^2}{\sigma_l^2}\right)}{\sum_{k \neq j} \exp(-\frac{d(l,k)^2}{\sigma_l^2})} \qquad (II)$$

These neighborhoods are compared based on the Kullback-Leibler divergence (KLD). Applying *(I)* and *(II)* KLD is used to define the precision F_P and recall F_R:

$$F_R = -\frac{1}{N}\sum_{l}^{N}\sum_{j \neq l} p_j(l) \log\left(\frac{p_j(l)}{q_j(l)}\right) \qquad (6.9a)$$

$$F_P = -\frac{1}{N}\sum_{l}^{N}\sum_{j \neq l} q_j(l) \log\left(\frac{q_j(l)}{p_j(l)}\right) \qquad (6.9b)$$

The precision and recall are plotted using a receiver operating characteristic (ROC)-like approach, in which the negative definition of the values results in the best projection method being displayed in the top right corner. The authors call this measure smoothed because it is not normalized, and theyalso propose a normalized version, with values lying between zero and one, based on ranks instead of distances. Note that the KLD and the symmetric KLD do not follow the triangle inequality for metric spaces.

6.1.9 Rescaled Average Agreement Rate (RAAR)

The average agreement rate is defined in Eq. *I* as

$$Q(knn) = \frac{1}{N}\sum_{j=1}^{N} \frac{\left|H_j(knn, I) \cap H_j(knn, O)\right|}{knn} \qquad (6.10)$$

in [Lee et al., 2014], analogously to the LCMC, using the unified co-ranking framework [Lee/Verleysen, 2008], in which the T&D, MRRE, and LCMC measures can all be summarized mathematically (for further details, see [Lee/Verleysen, 2009]). [Lee et al., 2014] argues that to enable fair comparisons or combinations of values of Q(knn) for different neighborhood sizes, the measure in Eq. 6.10 must be rescaled to

$$F(knn) = \frac{(N-1)Q(knn) - knn}{N-1-knn}, 1 \leq knn \leq N-2 \qquad (6.10')$$

This quantity is called the rescaled average agreement rate (RAAR). The values of F lie in the interval between zero and one, with a logarithmic knn scale and a scalar value that can be obtained by calculating the area under the curve (AUC).

6.1.10 Stress and the Shepard Diagram

The original multidimensional scaling (MDS) measure has various limitations, such as difficulties with handling non-linearities (see [Shepard, 1980] for a review); moreover, the underlying metric must be Euclidean, and Sammon mapping is simply a normalized version of MDS. Therefore, only non-metric MDS is considered here. The calculated evaluation measure is known as the stress and was first introduced in [Kruskal, 1964a]. Here, the stress F is defined as shown in Eq. 6.11. The disparities $\xi_{i,j}$ are the target values for each $d(j,l)$, meaning that if the distances in the output space achieve these values, then the ordering of the distances is preserved between the input and output spaces [Goodhill et al., 1995, pp. 8-9].

$$F = \sqrt{\frac{\sum_{j \neq l}\left(D(j,l) - \xi_{i,j}\right)^2}{\sum_{j \neq l} D(j,l)^2}} \qquad (6.11)$$

The input-space distances are used to define this measure based on a Euclidean graph. Several algorithms exist for calculating $\xi_{i,j}$. [Kruskal, 1964a] himself regarded *F* as a sort of residual sum of squares. A smaller value of *F* indicates a better fit. Therefore, perfect neighborhood preservation is achieved when *F* is equal to zero [Kruskal, 1964a]. The author describes *F* in terms of percentages, where values below 5% imply good neighborhood preservation. *F* can be described as the deviation from a perfect scatter plot of the distances in I versus the distances in O. This scatter plot is known as the Shepard diagram [Shepard, 1980 Fig 1C].

Here, the use of a density plot based on Pareto density estimation (PDE) [Ultsch, 2005b], instead of a scatter plot, is proposed. The author also proposes calculating Kendall's τ for these density plots.

6.1.11 Topographic Product

The topographic product [Bauer/Pawelzik, 1992] and an improved version thereof [Revuelta et al., 2004] were originally defined for neural maps, but in contrast to the quantization error [Uriarte/Martín, 2005] and the topographic error [Kiviluoto, 1996], it is possible to generalize the idea of the topographic product to all projection methods. Let the points $l_M \in H(knn(j), M)$ constitute the neighborhood of a point j in a metric space M defined based on a KNN graph and sorted in ascending order of knn; then,

$$q(j, knn) = \frac{d(j, l_I)}{d(j, l_O)} \qquad (I)$$

$$Q(j, knn) = \frac{D(j, l_I)}{D(j, l_O)} \qquad (II)$$

Q represents the distance between the point j $\in I$ and the k-th nearest neighbor $l_I \in I$ in the input space I divided by the distance between the point j $\in I$ and the point $l_O \in I$ corresponding

to the k-th nearest neighbor in O. Now, the product of q and Q of *(I)* and *(II)* for all orders knn can be calculated in Eq. 6.12:

$$P(j,n) = \left(\prod_{knn=1}^{n} q(j,knn) * Q(j,knn) \right)^{\frac{1}{2n}} \qquad (6.12)$$

The resulting QM is then defined as

$$F = \frac{1}{N(N-1)} \sum_{j}^{N} \sum_{knn}^{N-1} \log\big(P(j,knn)\big) \qquad (6.12')$$

F takes different values depending on whether the dimension of the output space is smaller than (F<0), similar to (F≈0) or greater than (F>0) the dimension of the input space [Revuelta et al., 2004]. Thus, in our case, F is always smaller than zero. [Revuelta et al., 2004] improved the topographic product by using the shortest-path distances in a Euclidean graph (geodesic distances) in Eq. *(I')* and *(II')* instead of the direct distances of Eq. *(I)* and *(II)*:

$$q(j,knn) = \frac{g(j,l_I)}{g(j,l_O)} \qquad (I')$$

$$Q(j,knn) = \frac{G(j,l_I)}{G(j,l_O)} \qquad (\mathrm{II}')$$

6.1.12 Topographic Function (TF)

The topographic function (TF) for SOMs was introduced in [Villmann et al., 1994]. This measure operates on Voronoi tessellations [Toussaint, 1980]. The TF quantifies the identity of the Delaunay graphs in I and O [Herrmann, 2011]. This work follows the general definitions found in [Villmann et al., 1997], where the TF is defined as given in Eq. 6.13 (denoted by *F*), with $h \neq 0$ being the cardinality of O or I:

$$F(h) = \frac{1}{N} \sum_{j=1, j \in I}^{N} \phi(j,h) \qquad h \neq 0 \qquad (6.13)$$

$$\phi(j,h) = \#\{\forall l \in I: g(l,j,\mathcal{D}) > h \wedge G(l,j,\mathcal{D}) = 1\}, \; h > 0 \qquad (6.13a)$$

$$\phi(j,h) = \#\{\forall l \in I: g(l,j,\mathcal{D}) = 1 \wedge G(l,j,\mathcal{D}) > |h|\}, h < 0 \qquad (6.13b)$$

The shortest path in the Delaunay graph of the input space between the data points $(l, j) \in I$ is denoted by $G(l,j,\mathcal{D})$, and that between the projected points $(l, j) \in O$ is denoted by $g(l,j,\mathcal{D})$. The Delaunay-graph distances *G* and *g* are equal to the number of Voronoi cells between the two points. If *h* is greater than zero, then $(l, j) \in I$ are neighbors in the input space, and if *h* is smaller than zero, then $(l, j) \in O$ are neighbors in the output space.

In Eq. 6.13a, ϕ represents the number of neighbors surrounding a data point $j \in I$ at a Delaunay distance greater than *h,* with the restriction that only the projected points $l \in O$ that are located in adjacent Voronoi cells in O are considered.

The converse situation is considered in Eq. 6.13b: ϕ represents the number of neighbors surrounding a projected point $j \in O$ at a Delaunay distance greater than h, with the restriction that only the data points $l \in I$ that are located in adjacent Voronoi cells in *I* are considered.

In summary, the shape of $F(h)$ enables a detailed discussion of the magnitude of distortions occurring in O [Bauer et al., 1999]: "Small values of h indicate that there are only local dimensional conflicts, whereas large values indicate the global character of a dimensional conflict" [Villmann et al., 1997]. [Bauer et al., 1999] proposed the following simplified equation:

$$F(h=0) = F(h=1) + F(h=-1) \qquad (6.13')$$

Here, h is equal to zero if and only if two points are neighbors in both the input space and the output space; thus, the overlap of Voronoi neighbors in I and O is required.

6.1.13 Trustworthiness and Discontinuity (T&D)

[Venna/Kaski, 2001] introduced the T&D measures, namely, trustworthiness and discontinuity. For each point j, let the points $l \in \mathrm{H_j}(\mathrm{knn}, \mathrm{O}\backslash\mathrm{I})$ be in the neighborhood consisting of the k nearest neighbors of the point j in the output space O, but not in the input space. Then, the T&D are defined as

$$F_1(knn) = 1 - \frac{1}{N(knn)} * \sum_{j,} \left(\sum_{l \in \mathrm{H_j}(\mathrm{knn},\mathrm{O}\backslash\mathrm{I})} (R(j,l) - knn) \right) \qquad (6.14a)$$

$$F_2(knn) = 1 - \frac{1}{N(knn)} * \sum_{j,} \sum_{l \in \mathrm{H_j}(\mathrm{knn},\mathrm{I}\backslash\mathrm{O})} (r(j,l) - knn) \qquad (6.14b)$$

where $N(knn)$ is a normalization factor that scales the values to the interval between zero and one [Kaski et al., 2003]. F_1 is the trustworthiness (T), and F_2 is the discontinuity (D). By counting the number of intruders, the T&D measures quantify the difference in the overlap of rank-based neighborhoods in I and O: F_1 represents the number of points that are incorrectly included in the input-space neighborhood, and F_2 represents the number of points that are incorrectly ejected from the input-space neighborhood.

[Venna/Kaski] claim that the trustworthiness (F_1) quantifies from "how far from the original neighborhood [in the input space] the new points [$l \in I$] entering the [output-space] neighborhood [*H(knn, O/I)*] come" [Venna/Kaski, 2001, p. 487]. For the calculation of the T&D measures, KNN graphs must be generated for various knn values. Then, the trend of the curve can be interpreted. It is unclear how many knn values must be considered. Hence, knn values up to 25% of the total number of points are plotted. [Lee/Verleysen] showed that the T&D measures can be expressed as a special case of the co-ranking matrix [Lee/Verleysen, 2009].

6.1.14 U-ranking

In [Ultsch/Herrmann, 2005], a QM based on a lattice was proposed. To generalize the idea to any projection method, one would use a graph. Let Γ be a graph, and let g(l, j, Γ) be the shortest path between the projected points $(j, l) \in O$; then, the U-distance can be generalized as

$$u(j,l) = \mathrm{g(l,j,\Gamma)} \qquad (6.15)$$

Let $\big(u(j,1), \ldots, u(j,n)\big)$ be the ascending sequence of all U-distances, as defined in Eq. 6.15, with respect to an arbitrary projected point j. The rank $r(j,l) = y \in \{1, \ldots, n\}$ represents the y^{th} position in the consecutive sequence of all U-distances $u(j,l)$ with respect to a projected point $l \in O$. Now, the minimal U-ranking measure can be defined as follows:

$$F(j) = \sum_{\iota \in \{i | x_i \in H(x_j,\ I)\}} r(j,l) \qquad (6.15')$$

Considering [Lötsch/Ultsch, 2014], a good choice for Γ is the Delaunay graph $\mathcal{D}$.

6.1.15 Overall Correlations: Topological Index (TI) and Topological Correlation (TC)

Various applications of the two correlation measures introduced below can be found in the literature.

The first type of correlation was introduced in [Siegel/Castellan, 1988] as Spearman's ρ and, in the context of metric topology preservation, was renamed as the topological index (TI) in [Bezdek/Pal, 1993]; see [Bezdek/R Pal, 1995] for further details. In Eq. 6.16, we follow the definition of the TI given in [Bezdek/R Pal, 1995], with $\kappa = n(n-1)/2$, where n is the number of distances:

$$F = 1 - \frac{6}{\kappa^3 - \kappa} \sum_{l,j=1}^{\kappa} \big(R(j,l) - r(\mathrm{j},\mathrm{l})\big)^2 \qquad (6.16)$$

The values of the TI are between zero and one, but [Goodhill et al., 1995] argued that the values of Spearman's ρ depend on the dimensions of the input and output spaces. Moreover, research has indicated that the elementary Spearman's ρ does not yield proper results for topology preservation [Karbauskaitė/Dzemyda, 2009].

[Handl et al., 2006] used the Pearson correlation, which is also called the topological correlation (TC) [Doherty et al., 2006]. The latter is notable because Delaunay-graph distances are used instead of Euclidean distances, as illustrated in the following equation:

$$F = \frac{1}{N} \sum \big(\mathrm{g}(\mathrm{l},\mathrm{j},\mathcal{D}) - \hat{\mathrm{g}}(\mathcal{D}) * \kappa^{-1}\big) * \big(\mathrm{G}(\mathrm{l},\mathrm{j},\mathcal{D}) - \widehat{\mathrm{G}}(\mathcal{D}) * \kappa^{-1}\big) \qquad (6.17)$$

where $\widehat{\mathrm{g}(\mathcal{D})}$ and $\widehat{\mathrm{G}}(\mathcal{D})$ are the means of the entries in the lower half of the distance matrices and $\kappa = n(n-1)/2$, with n being the number of distances. The TC is preferable to the TI as a means of characterizing topology preservation because in the case of the TI, the matching of extreme distances is sufficient to yield reasonably high overall correlation values [Handl et al., 2006].

6.1.16 Zrehen's Measure

Zrehen's measure operates on the empty ball condition of Gabriel graphs [Gabriel/ Sokal, 1969]. The neighborhood of each pair of projected points (l, j) in the output space is depicted using locally organized cells:

> *"A pair of neighbor cells A and B is locally organized if the straight line joining their weight vectors W(A) and W(B) contains points which are closer to W(A) or W(B) than they are to any other" [Zrehen, 1993, p. 664].*

In this work, the strong connection between the TF value $F(-1)$ and Zrehen's measure [Bauer et al., 1999] is remarked, but in contrast to [Zrehen, 1993], who assumed a neural net in two dimensions with precisely defined neighborhoods, here the output-space neighborhood is generalized to a Gabriel graph representation. Furthermore, for each pair of nearest neighbors, the

TF considers the neighborhood order *h* for that pair, whereas [Zrehen, 1993] counts the number of intruding points in neighborhoods of all orders *h* (for details, see the section on the TF above). In summary, if the condition $(l,j) \in H(1, Gabriel, O)$ is met, then all points $m \in I$ that lie between the corresponding points $(l,j) \in H(Gabriel, I)$ are deemed intruders and are counted. The sum of the number of intruders for all pairs of neighbors is normalized using a factor that depends only on the size and topology [Zrehen, 1993]:

$$f(j,l) = \#\{\forall k \in I \backslash \{l,j\}: (l,k) \in H_j(Gabriel, I) \wedge g(l,j,Gabriel) = 1 \wedge G(j,k) < G(j,l)\} \qquad (6.18)$$

$$F = \frac{1}{N} * \sum_{j,l \neq j} f(j,l) \qquad (6.18')$$

where N is the number of data points. The range of F starts at zero and extends to positive infinity, with a value of zero indicating the best possible projection.

6.2 Types of Quality Measures for Assessing Structure Preservation

In general, three types of QMs and some special cases can be identified, as shown in Figure 6.3. The first type of measure is called *compact*[33] because a measure of this type compares the arrangement of all given points in the metric space as expressed in terms of distance. In the literature, the term *topographic* is often used for such measures, e.g., [Goodhill et al., 1995]. These measures depend on some kind of comparison between inter- and intracluster distances. Measures in the second group are based on a neighborhood definition and, analogously to the terminology used in chapter 3, are called *connected*. These QMs rely on a type of predefined neighborhood *H* based on graph theory with a varying neighborhood extent *k*; thus, these neighborhoods are denoted by $H_j(k, \Gamma, M)$ (see chapter 2 for the corresponding definition). The expression *topology preservation* is often used in reference to this type of measure, e.g., [Bezdek/R Pal, 1995]. The special cases are grouped together under the term SOM-based measures. These measures, namely, the quantization error [Uriarte/Martín, 2005] and the topographic error [Kiviluoto, 1996], are not considered any further here because they require calculations of the distances between the data points in the input space and the weights of the neurons (prototypes) in the output space in an SOM. Instead of prototypes, general projection methods consider projected points, which can also refer to the positions of neurons on a lattice. Distances between spaces of unequal dimensions are not mathematically defined. A number of high-quality reviews are available on the subject of measuring SOM quality [Bauer et al., 1999; Beaton et al., 2010; Pölzlbauer, 2004].

The neighborhood-based QMs are divided into two groups, called *unidirectional measures* and *direction-based measures*. The reason for this is explained in chapter 2, section 2.2.1: two points *(j, k)* that lie in the same direct neighborhood of point l in $H_l(1, \mathcal{D}, M)$ may not lie in the same neighborhood $H_l(knn = 2, K, M)$ in the KNN graph if the distance D(l, k) is greater than the distance D(l, m) for a point m behind point j (see Figure 2.4 in chapter 2.2.1).

[33] Analogously to the usage of this term in chapter 3, where a compact structure is defined by inter- versus intracluster distances.

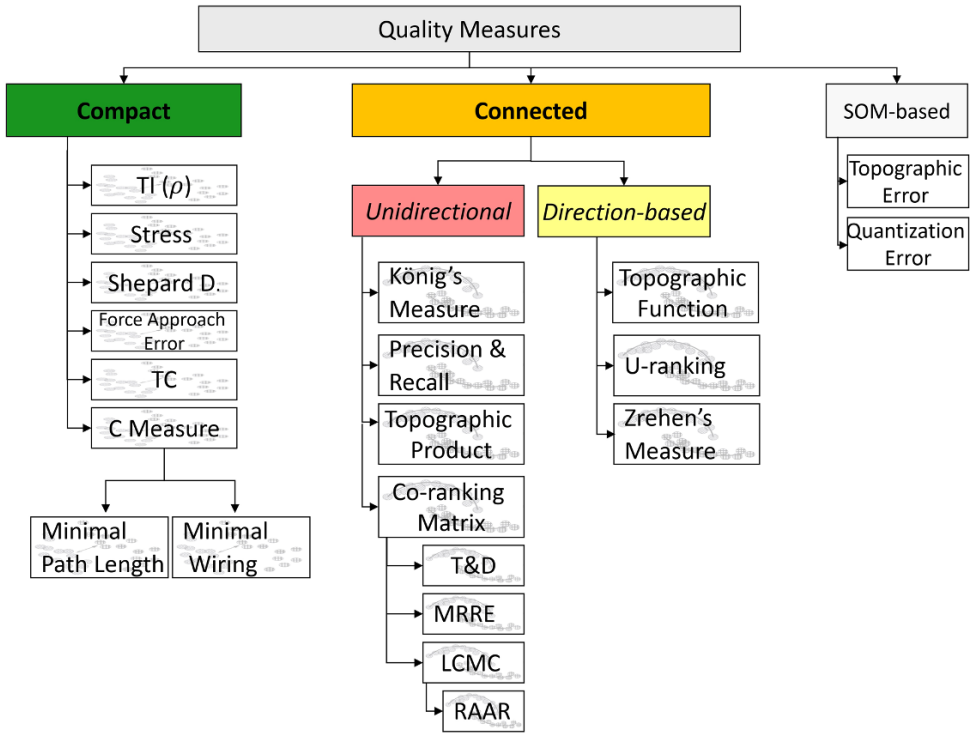

Figure 6.3: Groups of quality measures (QMs). The "Compact" group is only able to evaluate projections of compact structures (shaded with the first pattern), whereas the group of "Connected" QMs should be able to evaluate projections of connected structures (shaded with the second pattern) if the neighborhood definition is properly chosen. SOM-based measures are QMs that require weights of neurons (prototypes) and therefore are not generalizable to every projection method.
Supervised methods are not considered here (see chapter 3 for details).
Abbreviations: trustworthiness and discontinuity (T&D), mean relative rank error (MRRE), local continuity meta-criterion (LCMC) and rescaled average agreement rate (RAAR).

6.2.1 Theoretical Assessment of Quality Measures

A good QM should reflect the quality of structure preservation and have the following properties:

I. The result should be easily interpretable and should enable a comparison of different projection methods.
II. The result should be deterministic, with no or only simple parameters.
III. The result should be statistically stable and calculable for high-dimensional data in $\mathbb{R}^d$.
IV. The result should measure the preservation of high-dimensional discontinuities and should distinguish between backward projection errors (BPEs) or forward projection errors (FPEs) and gaps based on high-dimensional discontinuities.

QMs for evaluating the preservation of compact structures are easily interpretable; this is because they measure the quality of the preservation of distances. In most cases, the outcome is a single value in a specified range. However, no projection is able to completely preserve all distances or even the ranks of the distances [Drygas, 1978; Kirsch, 1978; Schmid, 1980]; here,

it is argued that only the preservation of discontinuities in the distances is important. Therefore, any attempt to measure the quality of a projection by considering all distances is greatly disadvantageous. For example, the major disadvantage of the stress and the C measure is that the largest distances, which are likely associated with outliers in the data, exert the strongest influences on the F value. Moreover, the C measure does not consider gaps. Correlation measures capture only linear correlations; however, in most cases, a non-linear projection method is required for structure preservation [Verleysen et al., 2003]. Additionally, outliers resulting in extreme distances are over-weighted in all correlation approaches.

QMs of the second type, connected measures, compare only local neighborhoods *H*. For unidirectional connected QMs, it is necessary to choose the correct number of k nearest neighbors, which is a complicated problem in itself. Even worse, for the comparison of different projection methods, it may be necessary to choose different knn values for the output space if there is a need to measure structure preservation. For this reason, unidirectional QMs that result in a single value, such as König's measure [König, 2000], do not satisfy quality conditions I and II. In other approaches, e.g., MRRE and T&D, two F values are obtained for every knn, and it is necessary to plot both functions, $F_{1/2}(knn)$. In this case, no distinction is possible between gaps and FPEs. Any further comparison of functional profiles for different projection methods is abstract and, consequently, not easily interpretable. Notably, the co-ranking matrix framework defined in [Lee/Verleysen, 2009, 2010] allows for the comparison, from a theoretical perspective, of several measures (the MRRE, T&D, and LCMC measures) based on $H(knn, K, M)$. However, no transformation of the co-ranking matrix into a single meaningful value exists [Mokbel et al., 2013], and the practical application of co-ranking matrices is controversial [Lueks et al., 2011]. With regard to the LCMC, [Chen/Buja, 2009] showed that it is statistically unstable and not smooth. Consequently, conditions I and II are not met, but the KNN graph is always calculable (IV).

The direction-based approach has the advantage that a distinction between FPEs and gaps is possible. However, an obvious disadvantage is the very high cost of calculation: $O\left(d^{\frac{n}{2}}\right)$ for a Delaunay graph and $O(n^2)$ for a Gabriel graph [Aupetit, 2003]. [Villmann et al., 1997] attempted to solve this problem by proposing an approximation of the intrinsic dimension of [Grassberger/Procaccia, 1983]. In theory, the TF seems to be the best choice, but in the context considered here, a projection is defined as a mapping into a lower-dimensional space. In this case, the quality measure F(h) is equal to zero for h<0. It follows that F(h=0)=F(h=1)+F(h=-1)=F(h=1). Consequently, half of the definition proves to be useless for the purpose considered here. The second problem is that the TF does not consider the input distances, apart from calculating the Delaunay graph in the input space. Thus, there is no difference between FPE and BPE, as long as no other points lie in between. Further disadvantages include numerical instability, because the Delaunay graph is sensitive to rounding errors in higher dimensions, and the fact that the Delaunay graph does not always correctly preserve neighborhoods if the intrinsic dimensionality of the data does not match the dimensionality of the output space O [Bauer et al., 1999].

Based on the classification of the QMs into semantic groups, here, one is able to identify several approaches that have not yet been considered. For example, one could develop a QM based on unit disk graphs.

6.2.2 Practical Assessment of Quality Measures

Various QMs were used to evaluate the structure preservation of projections of the Hepta and Chainlink data sets. In supplement A**,** it is shown that every approach used to measure the quality of projection methods is based on the preservation of discontinuities only when the discontinuities serve as a representation of compact or connected structures (directed or unidirectional). Consequently, the assessment of projections using QMs requires prior assumptions about the underlying structure of the data. If these assumptions are wrong, the QM will fail to correctly measure the projection quality. Figure 6.4 and 6.5 show the compact QM results obtained using the Shepard density plot method, introduced earlier in the chapter, for the Hepta and Chainlink data sets. It is possible to evaluate the preservation of compact structures in the Hepta data set (Figure 6.4), whereas the evaluation of the preservation of connected structures fails (Figure 6.5).

None of the QMs is fully credible. This is because none of them is able to measure structure preservation in all possible cases of the existence of discontinuities in the input space. To date, QMs have mostly been applied to data sets, such as a Swiss roll [Mokbel et al., 2013] or a sphere [Venna et al., 2010], for which the problem lies only in the visual representation of a continuous high-dimensional object. Therefore, the aim has been to measure the BPE and FPE. However, these examples show that structure preservation is more important, and if the goal is to visualize structures that can be used in clustering algorithms, higher FPEs and BPEs are sometimes necessary.

In supplement A, the simple Hepta example shows that every connected QM has difficulty capturing the quality of structure preservation. This is because such measures depend on compact structures defined by intra- versus intercluster distances (in a Euclidean graph). The Chainlink example illustrates that compact QMs are unsuccessful because each ring is closer to some points in the other cluster than it is to points in its own cluster, and therefore, the relevant structures are of the connected type. The density plots obtained using the Shepard diagram and Kendall's τ approaches are only able to capture discontinuities that can be unambiguously identified based on the intra- versus intercluster distances. This is not the case for the Chainlink data set, and consequently, these compact QMs fail for this data set. Moreover, because some connected QMs are not direction-based, even they encounter difficulties in evaluating structure preservation.

It seems that in the case of discontinuities in data and data sets that contain natural clusters, the user must make certain assumptions regarding which structures are most relevant and should be preserved. Based on this decision, the user can choose the most appropriate QM. Furthermore, the problem of trial-dependent projections, which is mostly ignored in the literature, is demonstrated in the example of the CCA projection of the Chainlink data set.

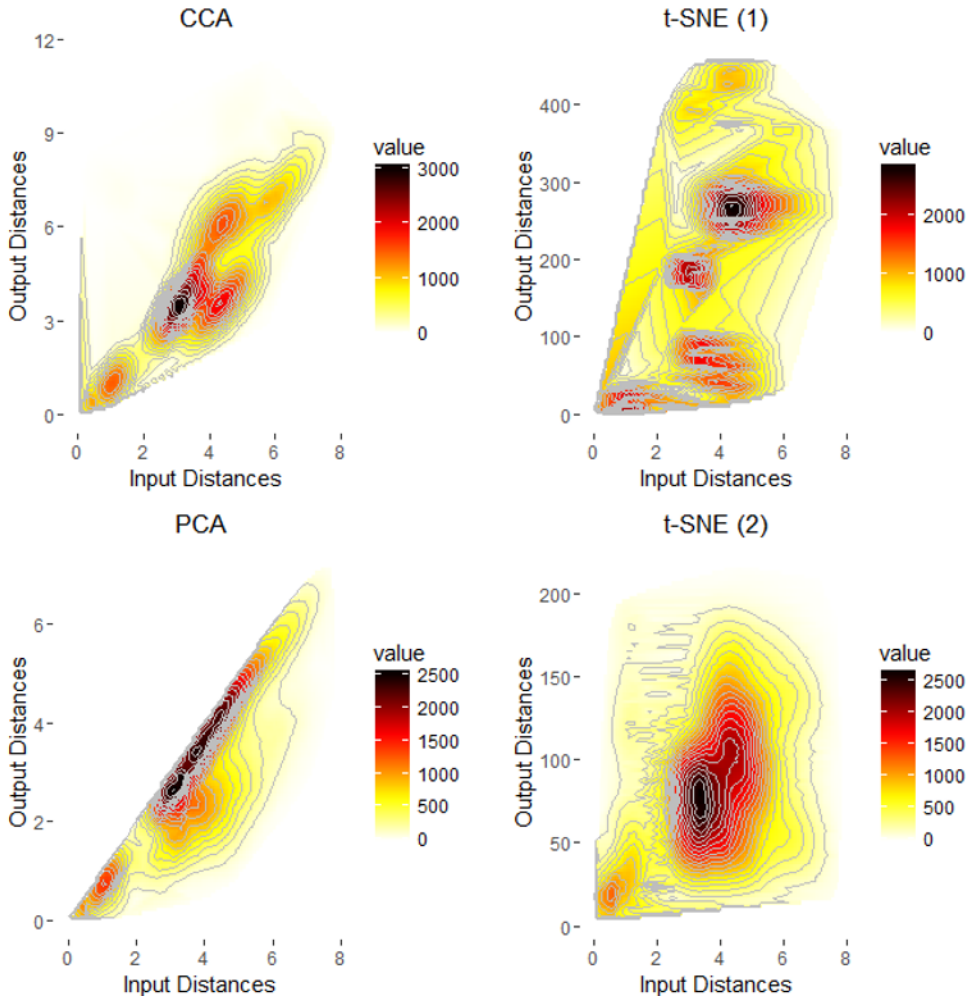

Figure 6.4: Density plots of the Shepard diagrams [Shepard, 1980] of the four projections of the Hepta data set shown in chapter 5, Figure 5.2. It is clearly apparent that PCA best preserves the structure of the data.

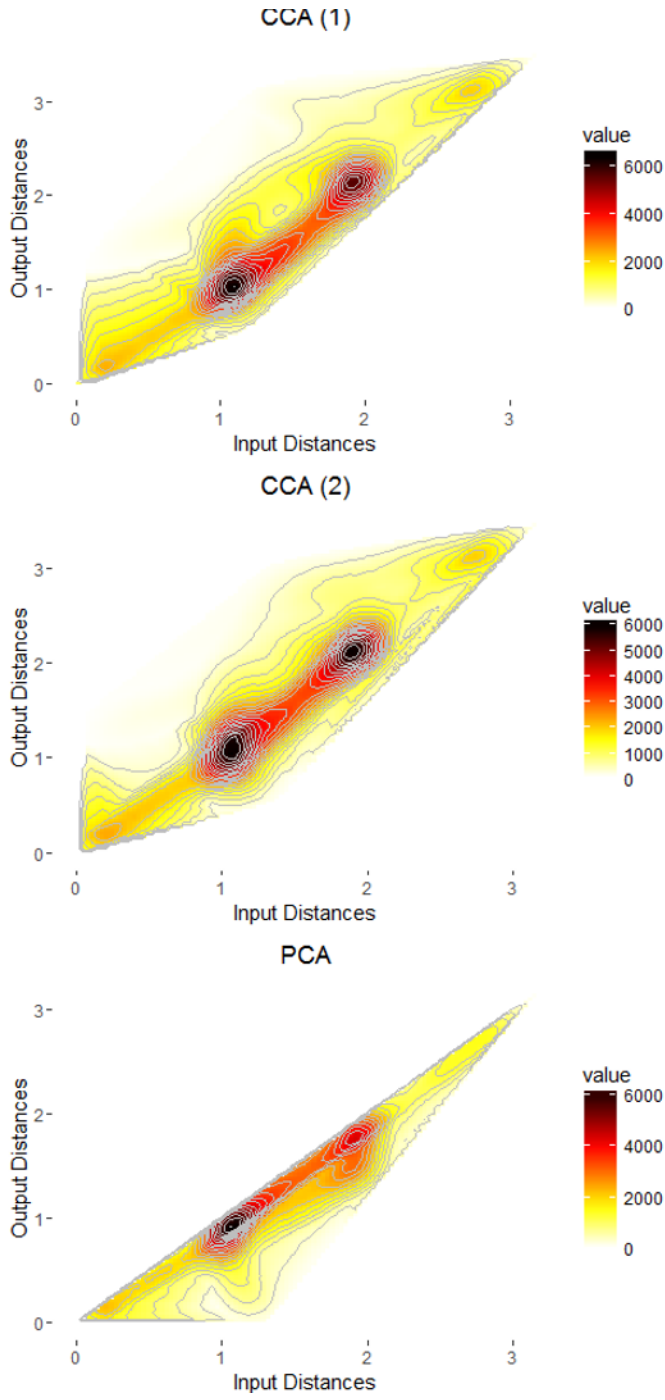

Figure 6.5: Density plots of the Shepard diagrams (density plots) for three projections of the Chainlink data set. PCA appears to produce the best projection of the data set, but in reality, it results in the worst structure preservation (see the supplement A). No clear difference between the CCA projections can be distinguished.

6.3 Introducing the Delaunay Classification Error (DCE)

On the one hand, QMs have difficulty measuring structure preservation when discontinuities exist in data sets (supplement A). On the other hand, in the case of natural clusters, discontinuities are important for cluster analysis, and projections of high-dimensional data sets should be able to visualize cluster structures accordingly. Consequently, identifying the most suitable method of evaluating projections of high-dimensional data for every case of high-dimensional discontinuities, with no available prior classification, remains an unsolved problem. However, if a prior classification of the data is known and if it represents patterns characterized by discontinuity, then these structures can be used for projection evaluation.

In chapter 5, it was shown that for every projection produced by any projection method, the generation of a U-matrix is possible. Consequently, the approach proposed herein assumes that an abstract U-matrix is available for every projection, as proven in [Lötsch/Ultsch, 2014] in the case of SOMs. Therefore, a Delaunay graph can be computed in the output space, and the edges are weighted using the high-dimensional distances in the input space.

Let $c \in C$ be the classification of the points $j \in I$ in the input space, where C_k is a cluster of C and N=|I|. Let $l \in O$ be the projected points in the output space that are mapped to I, and let $H_j(1, Del, O)$ be the direct neighborhood of j in the Delaunay graph in the output space. Then, the neighboring points of j are sorted using the Euclidean input-space distances between j and these neighboring points $l \in H_j(1, Del, O)$:

$$\widetilde{H}_j(1, Del, O, knn) = \left\{ l \in H_j(1, Del, O) \middle| \forall\, l_1, \dots l_k, D(l_1, j) < D(l_2, j) < \cdots < D(l_{knn}, j) \right\} \tag{6.19a}$$

where the number of nearest neighbors considered is

$$knn \in \mathbb{N}, \qquad knn \leq |\mathrm{H_j}(1, Del, O)| \tag{6.19b}$$

Then, the incorrectly classified points in the neighborhood $\widetilde{H_j}(1, \mathrm{knn}, Del, O)$ can be counted as follows:

$$|\bar{C}_k(l)| = \left| \{ p \in I, j(p) \in O | \forall p, j(p) \in |\widetilde{H}_l(1, Del, O, knn) \right| \wedge p \notin \bar{C}_k(l) \}| \tag{6.19c}$$

Finally, the DCE measure is defined as

$$DCE = \frac{1}{N} \sum_{\mathrm{knn}=2}^{\mathrm{k}} \sum_{l=1}^{N} \frac{|\bar{C}_k(l)|}{|\widetilde{H}_j(1, Del, O, knn,)| - 1} \tag{6.19d}$$

A low DCE value indicates a structure-preserving projection. Following the discussion in [Ultsch, 2016a], the DCE can be simplified to

$$DCE = \sum_{l,j=1}^{N} HD_j(N) * cc_{lj} \tag{6.19e}$$

where $HD_j(N) = \{1, 1 + \frac{1}{2}, \dots, 1 + \frac{1}{2} + \dots \frac{1}{n}\}$ is the vector of the decay function and CC_{ij} is an NxN matrix with the following definition. Let $NN_{ij} = D_{lj} * Delauany_{lj}$ be the distance matrix multiplied by the Delaunay adjacency matrix, where every element of this adjacency matrix is defined as

$$delaunay_{lj} = \begin{cases} 1, & if\ l\ and\ j\ are\ connected \\ \infty, & if\ l\ and\ j\ are\ not\ connected \end{cases} \qquad (6.19f)$$

Let $\widetilde{NN_{lj}}$ be the matrix NN_{ij} with the columns sorted in ascending order; then, every element of the matrix CC_{lj} is defined as

$$cc_{lj} = \begin{cases} 0, & if\ l\ and\ j\ are\ in\ the\ same\ class \\ 1, & otherwise \end{cases} \qquad (6.19g)$$

With the help of [Ultsch, 2016a], the harmonic decay function is approximately $HD_j(N) \approx log(N) + 0.5772156649 + 1/(2 * N)$. It assigns the heaviest weights to the errors that are nearest to a given point. The range of the DCE, which is approximately $[0, N * \sum_{i=1}^{N} log(i) + 0.5772156649 + 1/(2 * i)]$, can be restricted to $[-2,2]$ by calculating a baseline. An example of a baseline is a NeRV projection ([Venna et al., 2010]) with $\lambda = 0.5$, which means that the precision and recall are equally weighted. The relative difference can be calculated as

$$RelDiff = \frac{x - y}{0.5 * (x + y)} \qquad (6.19h)$$

Then, the normalized DCE is defined as

$$F = RelDiff(DCE, baseline) \qquad (6.19)$$

When the relative difference is used in this way, the range of values is fixed to $[-2,2]$. A positive value indicates a lower error compared with the baseline projection, whereas a negative value indicates a higher error compared with the baseline. In addition, the use of the relative difference enables the comparison of different projection methods in a direct and statistical manner.

6.3.1 Summary

Overall, 19 QMs were reviewed in this chapter, and the most common measures used to assess the quality of projections were compared. The QMs were grouped into semantic classes with the aid of graph theory. The QMs presented in the literature require prior assumptions regarding the underlying high-dimensional structures in a data set of interest (examples, see supplement A). Here, it is argued that for structure preservation, one must assume the presence of discontinuities in the high-dimensional data, which should correspond to gaps in their two-dimensional projection. In the case of such structures, the QMs reviewed here seemingly do not capture the important and unavoidable errors that occur in the projections because they assume certain definitions regarding which types of neighborhoods should be preserved (see supplement A).

Otherwise, an objective function could be defined using the best QM, and it would always be possible to obtain a structure-preserving two-dimensional visualization or clustering by optimizing this objective function.

Hence, a new QM is required to measure the quality of structure preservation. It must utilize information provided by a prior classification. The DCE is formulated based on the idea that an abstract U-matrix is available for every projection method, as demonstrated in [Lötsch/Ultsch, 2014] for the case of SOMs. A generalized U-matrix visualization called topographic map method for any arbitrary projection method was presented in the previous chapter. The DCE

allows projections to be ranked and normalized compared with a baseline and also enables statistical testing.

This work will present an alternative approach using swarm intelligence, self-organization, and the Nash equilibrium concept [Nash, 1950] from game theory, with the goal of eliminating the need for an objective function. The expectation is that novel and coherent properties that can be used for visualization and clustering will emerge from such a system. Chapter 7 will explain the relevant concepts, and chapter 8 will introduce the Pswarm projection method, which serves as part of the Databionic swarm clustering algorithm.

7 Behavior-based Systems in Data Science

Many technological advances have been achieved with the help of bionics, which is defined as the application of biological methods and systems found in nature. A related, rarely discussed subfield of information technology is called databionics. *Databionics* refers to the attempt to adopt information processing techniques from nature. This chapter will discuss the imitation of natural processes (also called biomimicry [Benyus, 2002]) using swarm intelligence, which is a form of artificial intelligence (AI) [Bonabeau et al., 1999] and was introduced as a term in the context of robotics [Beni/Wang, 1989]. In the context considered here, AI may be described as a field of study that seeks to explain and emulate intelligent behavior in the form of a computational process[34] [Russell et al., 2003, p. 5].

Consequently*, swarm intelligence* is defined as the emergent collective behavior[35] of simple entities called agents[36][Bonabeau et al., 1999, p. 12]. An agent is a software entity, situated[37] in a given environment, that is capable of flexible, autonomous action in order to meet its design objectives [Jennings et al., 1998]. In the context of swarms, the terms behavior and intelligence are used synonymously, bearing in mind that in general, the definition of intelligence is controversial [Legg/Hutter, 2007] and complex [Zhong, 2010]. The properties of swarm behavior will be explained later in this section.

> *"There are […] three key concepts […] [related to agents]: situatedness, autonomy, and flexibility. Situatedness, in this context, means that the agent receives sensory input from its environment and that it can perform actions which change the environment in some way" [Jennings et al., 1998, p.8].*

Autonomy refers to an agent's capability for independent, decentralized action, and flexibility refers to its ability to proactively respond to its environment in a "timely fashion" [Jennings et al., 1998].

Inspired by Beni's definition of intelligent robots [Beni/Wang, 1993, p. 705], here, an intelligent agent is described as one whose behavior is neither random nor predictable [Beni, 2004, p. 4]. On the one hand, "intelligent behavior is the production of something ordered, i.e., unlikely to occur: an improbable outcome" [Beni, 2004, p. 3]. On the other hand, unpredictability is not equivalent to intelligence; a roulette, for example, is not intelligent [Beni, 2004, p. 3]. "It seems that somehow both unpredictability and the creation of some order are necessary to be able to speak of "intelligence" [Beni, 2004, p. 3]. In the context of data science, the first intelligent agents to be developed and applied were called DataBots [Ultsch, 2000a]. DataBots possess probabilistically defined movement strategies, take in food, consume food and store quantities of food. However, the question of whether DataBots themselves exhibit swarm intelligence is controversial [de Buitléir et al., 2012, p. 2], and as such, they will be separately introduced in the next section. It will be shown that in the case of swarm-organized projection (SOP)

[34] The author focuses on AI in the context of behavior; however, thought process and reasoning types of AI also exist, of which neural networks and Bayesian learning are representative examples.

[35] The term collective behavior generically denotes any behavior of agents in a system of more than one agent [Cao et al., 1997].

[36] See also a similar definition in [Martens et al., 2011, p. 2].

[37] "The word "situated," […] is intended to emphasize that the process of deliberation takes place in an agent that is directly connected to an environment" [Russell et al., 2003, p. 422].

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_7

[Herrmann, 2011], DataBots do not exhibit swarm intelligence, whereas in the case of Pswarm (introduced in the next chapter), they do.
Another example of the use of intelligent agents is Schelling's segregation model [Schelling, 1969, 1971]. The model consists of a lattice of square patches (tiling). Agents are located on this landscape, initially at random, with no more than one on any patch. The agents are of two different types, e.g., blue and red, and there are free patches available. Each agent has a tolerance parameter. A blue agent is "happy" when the ratio of blues to reds in its Moore neighborhood (the eight immediately adjacent patches) is above its tolerance threshold. Unhappy agents are allowed to move randomly to a new open position (white). Schelling's segregation model leads to segregation of the agents, even when individual agents have only a mild preference for living near agents of the same type. An example of the segregation process is illustrated in Figure 7.1.

> *"Originally the model was intended to explain how racialized city ghettos might emerge from individual choices, given even slight racial biases. Some important constraints on effective segregation have been described by [Vinković/Kirman, 2006]. Segregation is greatly increased if agents are allowed to jump to any node that yields less stress, instead of neighbouring nodes only" [Herrmann, 2011, pp. 54-55].*

Swarm behavior can be imitated based on observations of herds [Wong et al., 2014], bird flocks and fish schools [Reynolds, 1987], bats [Yang/He, 2013], or insects such as bees [Karaboga, 2005; Karaboga/Akay, 2009], ants [Deneubourg et al., 1991], fireflies [Yang, 2009], cockroaches [Havens et al., 2008], midges [Passino, 2013], glow-worms or slime moulds [Parpinelli/Lopes, 2011]. [Grosan et al.] define five main principles of swarm behavior: *Homogeneity,* meaning that every agent has the same behavior model; *Locality,* meaning that the motion of each agent is influenced only by its nearest neighbors; *Velocity Matching,* meaning that each agent attempts to match the velocity of nearby flockmates; *Collision Avoidance*, meaning that each agent avoids collisions with nearby agents; and *Flock Centering*, meaning that agents attempt to stay close to neighboring agents [Grosan et al., 2006, p. 2; Reynolds, 1987, pp. 6, 7]. Here, these definitions are given greater specificity in two respects.
First, the term *agent* is modified to the term *agents of the same type* because many swarms consists of more than one type of agent, e.g., small and large workers in the Pheidole genus of ants [Bonabeau et al., 1999, pp. 3, 4].Second, a swarm need not necessarily move. For example, fire ants self-assemble into waterproof rafts to survive floods [Mlot et al., 2011]. The individual ants are linked together to construct such self-assemblages [Mlot et al., 2011]. Therefore, velocity matching can result in a velocity of zero.

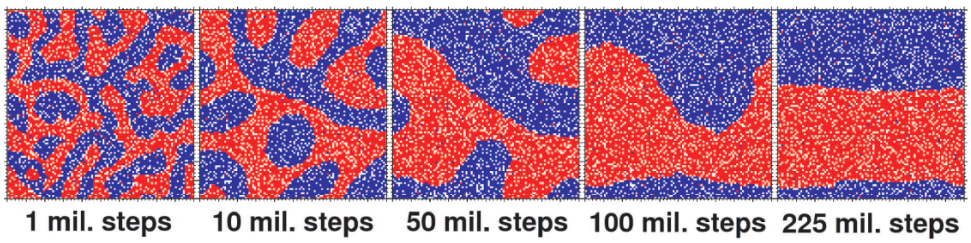

Figure 7.1: The Schelling model of a liquid on a periodic lattice [Vinković/Kirman, 2006, Fig. 5 a]. After 225 mil. steps the agents are fully segregated. The segregation requires many iterations if agents are allowed only to jump to the positions nearest to them.

If a swarm contains a sufficient number of agents, self-organization may emerge. *Self-organization* is defined as the spontaneous formation of patterns by a system itself [Kelso, 1997, p. 8 ff.], without any central control. The snowflake in Figure 7.2 serves as an example of self-organization. During self-organization, novel and coherent structures, patterns, and properties may arise [Goldstein, 1999]. This ability of a system to produce phenomena on a new, higher level is called *emergence* [Ultsch, 1999], and it is separately discussed in the next section.
"Self-organizing swarm behavior relies on four basic ingredients" [Bonabeau et al., 1999, pp. 22-25]: *positive feedback*, *negative feedback*, *amplification of fluctuations* and *multiple interactions*. The first two factors promote the creation of convenient structures and help to stabilize them. Fluctuations are defined to include errors, random movements and task switching. For swarm behavior to emerge, multiple interactions are required. Agents can communicate with each other either directly or indirectly. An example of direct communication is the dancing behavior of bees, in which a bee shares information about a food source, such as how plentiful it is and its direction and distance away [Karaboga/Akay, 2009]. Indirect communication is observed, for example, in the behavior of ants [Schneirla, 1971]. If the agents communicate only through modifications to their environment (through pheromones, for example), then this type of communication is defined as *stigmergy* [Beckers et al., 1994; Grassé, 1959].
The exact number of agents required for self-organization is unknown, but it should be not so large that it must be handled in terms of statistical averages and not so small that it can be treated as a few-body problem [Beni, 2004]. For example, 4096 neurons are required for self-organization in SOMs [Ultsch, 1999], and for the coordinated marching behavior of locusts, a minimum density of least 73.8 $locusts/m^2$ was reported in [Buhl et al., 2006, p. 1404].

Figure 7.2: Example of self-organization: a large, 10.1x10.1 mm snow crystal [Libbrecht, 2016]. This snow flake is a spontaneous formation of a pattern by molecules of H_2O.

Considering the two requirements stated above, Beni defined a swarm as a formation of cellular robots with a number exceeding 100 [Beni, 2004]. Here, consistent with [Beni, 2004], the argument is made that for self-organization[38], the number of agents should be higher than 100.
The two main types of swarm-based analysis discussed in data science, namely, particle swarm optimization (PSO) and ant colony optimization (ACO) [Martens et al.], are distinguished by the type of communication used: PSO agents communicate directly, whereas ACO agents communicate through stigmergy. PSO methods are based on the movement strategies of particles [Kennedy/Eberhart, 1995] and typically used as population-based search algorithms [Rana et al., 2011], whereas ACO methods are applied for sorting tasks [Martens et al., 2011]. In addition to being used to solve discrete optimization problems, PSO has been used as a basis for rule-based classification models, e.g., AntMiner, or as an optimizer within other learning algorithms [Martens et al., 2011], whereas ACO has been used primarily for supervised classification within the data mining community [Martens et al., 2011]. Pseudocode for both types of algorithms and illustrative descriptions can be found in [Abraham et al., 2006].

7.1 Artificial Behavior Based on DataBots

The term DataBots refers to agents in the sense discussed here. DataBots were introduced in [Ultsch, 2000a] as the first artificial-behavior-based approach to data science. Each DataBot $b_j \in B$ has a position $i_j \in O$ and takes in food, consumes food and stores quantities of food. Quantities of food are represented by numbers in the range from 0% to 100%. All positions lie on a toroidal lattice, and each DataBot is capable of detecting a scent λ at its current position. This approach is used to perform clustering tasks.
In [Ultsch, 2000c], each DataBot possesses an opinion, defined by one high-dimensional data point, and the DataBots are used as a projection method for a classification task. The movement of the DataBots is defined in terms of probabilities, which are computed using various movement programs called strategies, for each of the four directions (south, east, west and north) and for no movement (origin). With the use of these strategies, self-organization of the system is possible. Unlike in ACO methods, each DataBot possesses an opinion defined by a high-dimensional data point [Ultsch, 2000c]. Hence, reduction of the agents is impossible.
[Kämpf/Ultsch] suggested the use of movement strategies with a decreasing neighborhood radius. The underlying idea of the decreasing radius approach is to promote self-organization, first of a global structure and then of local structures [Kämpf/Ultsch, 2006]. In [Herrmann/Ultsch, 2008b], a set of additional strategies was defined for a subset of DataBots based on labeled data, requiring a prior classification. The authors used this approach to address a classification task by combining it with emergent self-organizing map (ESOM) and the gray-scale two-dimensional U-matrix method. The U-matrix was partitioned into clusters using an entropy-based heuristic algorithm called U*C [Ultsch, 2006].Here, it is assumed that the DataBots are defined similarly to their definition in [Herrmann/Ultsch, 2008b]: Let each DataBot $b_j \in B$ be an agent identified by a numerical vector $z_j \in \mathbb{R}^d$; it resides on a large, finite, two-dimensional discrete lattice that is embedded on the surface of a torus [Ultsch, 2003a]. The

[38] Beni himself only indirectly restricted systems that exhibit self-organization to those consisting of more than 100 agents [Beni, 2004].

current position of DataBot b_j is denoted by $i_j \epsilon$ O. Every DataBot $b_j = \{i_j, z_j\}$ emits a scent λ, which is detected by all other DataBots in its neighborhood.

By analyzing ant-based clustering[39] (ABC) [Lumer/Faieta, 1994] and the batch self-organizing map (batch-SOM) method [Kohonen/Somervuo, 2002] the local stress of an ABC projection[40] can be extracted [Herrmann, 2011, pp. 137-138; Herrmann/Ultsch, 2008a, p. 3; 2008c, p. 217; 2009, p. 4]: It is an upper limit on the best matching unit criterion[41] of batch-SOM and forms the topographic term of the Attractiveness function used in ant-based clustering. [Ultsch/Herrmann, 2010] used this mathematical stress term to define a scent as follows:

Let D(l, j) be the distance between two points $x_l, x_j \in$ I, let d(l, j) be the corresponding distance in the output space O, and let h_R: $R \rightarrow$ [0,1] be an arbitrary but continuous and monotonically decreasing function; then, the scent $\lambda(b_j, R)$: $\mathbb{R}_0^+ \times O \rightarrow \mathbb{R}_0^+$ is defined as

$$\lambda(b_j, R) = \frac{\sum_{l \epsilon I} h_R(d(j,l)) * D(j,l)}{\sum_{l \epsilon I} h_R(d(j,l))} \qquad (7.1)$$

The scent λ is the weighted sum of the distances to neighboring objects; consequently, h_R "realizes a neighborhood function by means of focus" [Herrmann, 2011, p. 65]. To better distinct this neighborhood function from the Databionic swarm, in the following chapters it will be referred to with the same capital letter $F_R = h_R$ as in [Herrmann, 2011].

7.1.1 Swarm-Organized Projection (SOP)

The discussion in this section is based on the thesis of [Herrmann, 2011], which is a continuation of the work of [Herrmann, 2009; Ultsch/Herrmann, 2010]. The SOP algorithm was proposed as a self-adaptive projection method with the aim of creating a cohesive visualization of clusters [Herrmann, 2011]. The algorithm combines a DataBot approach, a scent definition derived from the above analysis of ABC, and Schelling's segregation model [Schelling, 1969]: the better (weaker) the scent λ becomes, the happier the DataBot is. The SOP algorithm, as presented in Listing 7.1, operates on a finite data set with pairwise dissimilarities, which are usually defined as Euclidean distances [Herrmann, 2011]. The numeric vector z_j associated with each DataBot b_j represents a high-dimensional data point, and the cardinality of the data set I is equal to the number of DataBots. The positions of the DataBots are defined on a rectangular lattice tiling (quad grid) O, which is typically toroidal but could also be planar, in Cartesian coordinates $i(x, y) \epsilon O$, where the numbers of lines L and columns C must be set by the user. Every DataBot chooses between its current position and one new position. If the scent λ, which is defined by the function F_R, would be weaker in its new neighborhood, then the DataBot jumps to the new position. Another DataBot may already be located in the new position, but this does not affect the decision to jump.

In each iteration, all DataBots are allowed to move simultaneously [Herrmann, 2011]. An epoch ends when the following condition is met [Herrmann, 2011]: As long as the number of DataBots that want to jump exceeds an arbitrary threshold value, called a fixed point in [Herrmann, 2011],

[39] See next section for a more detailed description.
[40] In [Herrmann/Ultsch, 2008a] called topographic mapping.
[41] It "is a weighted sum of local input space distances" [Herrmann/Ultsch, 2009, p. 4].

the current epoch proceeds to the next iteration. Otherwise, the next epoch starts, with a decrease in the neighborhood radius R. To ensure the convergence of the algorithm, a maximum number of iterations must be set. [Kohlhof, 2010] proposes a 5% threshold and a maximum number of 500 iterations, but in [Herrmann, 2011], no exact numbers are indicated.

The maximum possible distance in the map space is defined by $R_{max} = \sqrt{L^2 + C^2}$, and the algorithm ends if the smallest possible radius $R = 1$ is reached [Herrmann, 2011, p. 65]. The following contradiction should be taken into account: sometimes, a different minimal radius (e.g., R=8 in [Herrmann, 2011, p. 118] for the gene data set, *R>1* in [Herrmann, 2011, p. 167] for the GPD194 data set) is chosen without any scientific basis other than the author's experience. In practice, the neighborhood function F_R is chosen to be a Gaussian function where the mean is equal to zero and the standard deviation is equal to the radius *R*. Each possible new position is drawn from a Gaussian-shaped probability distribution (Fig 4.1) [Herrmann, 2011, p. 64]. Pseudocode for the SOP algorithm is provided in [Herrmann, 2011, p. 65], with the scent $\lambda(b_j)$ defined as in equation (1).

Previous work has revealed, based on the practical experience of the inventor [Herrmann, 2009], that SOP is almost as good as or even better than the best of its carefully parameterized competitor methods, such as curvilinear component analysis (CCA), t-distributed stochastic neighbor embedding (t-SNE) and ESOM, in terms of the 1-nearest-neighbor classification accuracy and the specially formulated dispersion measure of [Herrmann, 2011, p. 101] on several natural and artificial data sets.

function O=sop(I)

for all $z_i \in$ *I: assign an initial random Cartesian position* $i(x,y) \in$ *O on the lattice to generate DataBots* $b_i \in B$

for R={Rmax,..., 1} do

m=Gaussian(R) of a Gaussian-shaped distribution: $N(m(x),s) + N(m(y),s)$

iteration=0

repeat

for j={1,...n} do

$l = \text{argmin}_j(\lambda(b_j))$ *with* $j = \{i,m\} \in O$

end for

iteration = iteration +1

until *{*$l \in O$ *fix with* $|\{l \in O \mid l = m|\}| <$ *threshold) OR (*$\text{iteration} > \text{i\_max}$*)}*

return O

end function SOP

Listing 7.1: The swarm-organized projection (SOP) algorithm as described in [Herrmann, 2011, p. 65]. The are some parameters to be set by a user (e.g. *Rmax*, *threshold,_max,* i_max*, ...*).

7.2 Swarm Intelligence for Unsupervised Machine Learning

As mentioned earlier in this chapter, there are two main types of artificial swarm optimization methods: PSO and ACO. In unsupervised learning, two additional approaches are known. The first one is based on bees [Karaboga/Akay, 2009], and the second is based on foraging theory [Stephens/Krebs, 1986].

For clustering tasks, PSO has mainly been applied in hybrid algorithms [Esmin et al., 2015]; e.g., [Van der Merwe/Engelbrecht, 2003] applied PSO combined with k-means clustering. Here, it is argued that the hybridization of PSO and k-means may improve the choice of centroids or may, in some special cases, even allow the problem of the number of clusters to be solved. However, this approach is subject to several of the shortcomings of k-means, which is known to search for spherical clusters [Hennig et al., 2015, p. 721]/[Hennig, 2015a, p. 18]; i.e., it is unable to find clusters in elementary data sets, such as those in the Fundamental Clustering Problems Suite[42] (FCPS) [Ultsch, 2005a].

According to [Rana et al., 2011], the advantages of the clustering process when the PSO approach is used are that it is very fast, simple and easy to understand and implement. "PSO also has very few parameters to adjust [Eberhart et al., 2001] and requires little memory for computation. Unlike other evolutionary and mathematical algorithms it is more computationally effective" [Rana et al., 2011] (citing [Arumugam et al., 2005]). Again according to [Rana et al., 2011], the disadvantages are the "poor quality results when it deals with large and complex data sets". "PSO gives good results and accuracy for single objective optimization, but for a multi objective problem it becomes stuck in local optima" [Rana et al., 2011] (citing [Li/Xiao, 2008]). Another problem with PSO is its tendency to reach fast and premature convergence at mid-optimum points [Rana et al., 2011]. It is difficult to find the correct stopping criterion for PSO [Bogon, 2013, p. 155], which is usually one of the following: a fixed maximum number of iterations, a maximum number of iterations without improvement or a minimum objective function error [Abraham et al., 2006; Esmin et al., 2015]. Hybrid PSO algorithms usually optimize an objective function [Bogon, 2013, pp. 39 ff, 46] and therefore always make implicit assumptions regarding the underlying structures of the data (see chapters 2, 4 and 5 for details). Notably, there is no single "best" criterion for obtaining a clustering because no precise and workable definition of "a cluster" exists [Jain/Dubes, 1988, p. 91]. For the task of dimensionality reduction, the swarm-inspired projection (SIP) method [Su et al., 2009] are discussed later in this section.

ACO methods for clustering tasks are referred to as ABC methods (for an overview, see [Kaur/Rohil, 2015]). ABC methods model the behavior of ant colonies, and data points are picked up and dropped off accordingly [Bonabeau et al., 1999]. ABC was introduced by [Deneubourg et al., 1991] as a way to explain the phenomenon of the gathering and sorting of corpses observed among ants. In an experiment (Figure 7.3), the ants formed cemeteries of dead ants that had been randomly scattered beforehand. [Deneubourg et al., 1991] proposed probability functions for the picking up and dropping off of the corpses. Because ants are very specialized in their roles, several different types of ants of the same species exist in a colony, and different individuals in the colony perform different tasks. The probabilities are calculated as functions of the number of corpses of the same type in a nearby area (positive feedback).

[42] See also the results presented in chapter 9.

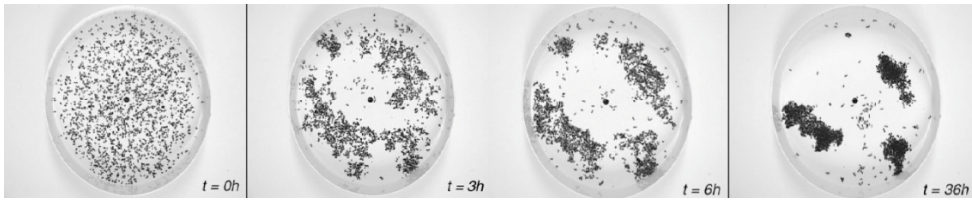

Figure 7.3: Randomly scattered ant corpses are clustered by living ants in a matter of hours [Bonabeau et al., 1999, p. 151; Martens et al., 2011, Fig.5]. The different stage depicted correspond to 0, 3, 6 and 36 hours after the beginning of the experiment.

For a clustering task, the ants and data points (representing ant corpses) are randomly placed on a lattice, and the ants move randomly across the lattice, at times picking up and carrying the data points [Lumer/Faieta, 1994]. The probabilities of picking up and dropping off the data points are modified according to a dissimilarity-based evaluation of the local density (see [Kaur/Rohil, 2015] and [Jafar/Sivakumar, 2010], citing [Lumer/Faieta, 1994]).

[Handl et al., 2006] enhanced the algorithm; they called their version Adaptive Time-dependent Transporter Ants (ATTA) because they incorporated adaptive heterogeneous ants and time-dependent transport activities into the algorithm. Further improvements to the picking up and dropping off activities were presented in [Omar et al., 2013; Ouadfel/Batouche, 2007], and improvements to the initialization and post-processing were proposed in [Aparna/Nair, 2014]. Another version of the approach was developed by introducing an annealing scheme [Tsai et al., 2004]. A feature of ABC algorithms is that the clustering objective is implicitly defined: neither the overall clustering objective nor the type of clusters sought is explicitly defined at any point during the clustering process[43] [Handl/Meyer, 2007].

The main problem in ABC lies in the fact that the picking up and dropping off behaviors are independent of the number of agents required to execute the task [Herrmann, 2011, p. 81; Herrmann/Ultsch, 2008a, 2008c, 2009; Tan et al., 2006]. Furthermore, ABC methods can be regarded as derived from the batch-SOM algorithm [Herrmann/Ultsch, 2008a]. From this perspective, an ABC algorithm possesses an objective function, which can be decomposed into an output density term multiplied by one minus a topographic quality term [Herrmann, 2011, pp. 137-138; Herrmann/Ultsch, 2008a, p. 3; 2008c, p. 217; 2009, p. 4]. Both terms are minimized simultaneously [Herrmann/Ultsch, 2008a, 2008c, 2009]. The output density term is easy to optimize but distorts the correct clustering of the data. Here, it is argued that at least 100 agents are required for self-organization in a swarm. However, this many agents are not required in ABC methods, and consequently, the self-organization property of ABC-based swarm algorithms is controversial.Methods of the third type are founded on an analysis of the behavior of bees [Karaboga, 2005]. These are hybrid approaches to clustering that use swarm intelligence in combination with other methods, e.g., k-means[44] [Karaboga/Ozturk, 2011; Marinakis et al., 2007; Pham et al., 2007; Zou et al., 2010] or SOM [Fathian/Amiri, 2008].

To the best of the author's knowledge, only seven instances of the application of AI in projection methods exist. One method is based on foraging theory, which focuses on two basic prob-

[43] This feature will be used in Databionic swarm.

[44] k-means is known to search for spherical clusters [Hennig et al., 2015, p. 721]/[Hennig, 2015a, p. 18]; see above.

lems: which prey a forager should consume and when a forager should leave a patch [Stephens/Krebs, 1986, p. 6]. A forager is viewed as an agent who compares a potential energy gain with a potential opportunity for finding an item of a superior type [Martens et al., 2011] (citing [Stephens/Krebs, 1986]). This approach is also called the prey model [Martens et al., 2011]: the average energy gain can be mathematically expressed in terms of the expected time, energy intake, encounter rate and attack probability for each type of prey. In the projection method proposed by [Giraldo et al., 2011], in addition to the characteristics of the approach described above, the "foraging landscape was viewed as a discrete space, and objects representing points from the dataset as prey." There were three agents defined as foragers. Here, the approaches based on the prey model are classified as basic swarm algorithms.

A second method, called the self-organizing swarm (SOSwarm) method, is a clustering method based on a hybrid of PSO and SOM [O'Neill/Brabazon, 2008]. In SOSwarm, 100 particles were used on a 10x10 SOM feature map. However, because only a few units are used, SOSwarm represents a combination of k-means-SOM (see chapter 3) with PSO. Thus, it can be viewed as an application of swarm intelligence, but it is questionable whether this swarm is self-organizing because 4096 neurons are required for self-organization in SOMs [Ultsch, 1999] and the conditions for self-organizing swarm behavior may not apply [Bonabeau et al., 1999, pp. 22-25].

A third method is known as the swarm-inspired projection (SIP [Su et al., 2009], as briefly mentioned above. SIP is a PSO approach that is loosely related to foraging theory because it is inspired by the foraging behavior of doves. The authors report that the number of doves should be significantly smaller than the number of data points and need only be higher than the expected number of clusters. Because of the small number of agents used, it is questionable whether this swarm is self-organizing, but as a PSO approach, it is an example of swarm intelligence.

The fourth approach, SOP [Herrmann, 2011], was already introduced. In terms of swarm behavior, the SOP algorithm does not consider collision avoidance (see the second section of this chapter), as seen from the fact that one or more DataBots may occupy the same position. After an annealing process, the SOP agents are uniformly distributed [Herrmann, 2011, pp. 68-69]; thus, the principle of flock centering is also disregarded. In the next chapter, it will be shown that the SOP algorithm also does not necessarily exhibit the property of *fluctuations* (referred to in the next section as *randomness*) because the position choices of the DataBots are predictable because of their self-interaction and the oblique neighborhood definition. In summary, SOP is a self-organizing swarm of DataBots based on Schelling's idea to unsupervised machine learning that cannot be regarded as an example of swarm intelligence.

Because ABC methods can be reduced to one ant, these approaches are classified as basic swarms. To exhibit swarm intelligence, a swarm must contain more than one independent agent. Therefore, LF [Lumer/Faieta, 1994] and its derivatives[45] ATTA-TM [Handl et al., 2006] and ASM [Xu et al., 2007] are not applications of swarm intelligence. Notably, the argument presented here is only valid for ABC methods of unsupervised learning; the categorization may prove invalid for other ACO methods that are supervised.

[45] The fifth, sixth and seventh applications of unsupervised learning.

The discussion presented in this section is summarized in Figure 7.4, in which only projection methods are explicitly listed. All of the various methods used for clustering cannot be illustrated in one figure. Thus, only general hybrid types are depicted. For all of the publications mentioned above, there is currently no open-source code[46] available except for applications of rule-based classification [Martens et al., 2011].

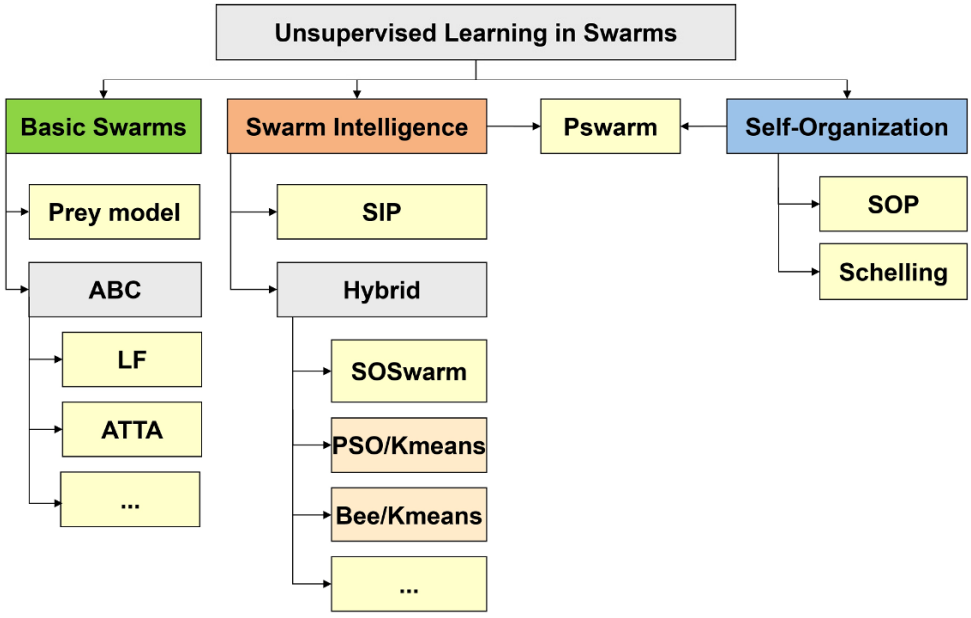

Figure 7.4: Types of swarm algorithms used in unsupervised learning. Pswarm will be introduced in the next chapter; it combines self-organization with swarm intelligence. Various PSO and bee hybrids are used for clustering tasks. Most of these are based on k-means. Aside from Schelling's segregation model, only projection methods are explicitly listed. Abbreviations: ant-based clustering (ABC), particle swarm optimization (PSO).

[46] The authors of [O'Neill/Brabazon, 2008; Su et al., 2009; Giraldo et al., 2011] were contacted via email, but only Giraldo et al. responded and provided their source code. Due to various limitations, it could not be used for this thesis.

7.3 Missing Links: Emergence and Game Theory

Through self-organization, novel and irreducible[47] structures, patterns, and properties can emerge in a complex system [Goldstein, 1999]. In analogy to SOMs [Ultsch, 1999], this idiosyncratic behavior of a swarm is defined here as *emergence* (see also [Stephan, 1999]).
Sometimes, a distinction is made between strong and weak emergence [Janich/Duncker, 2011, p. 19]. Here, only strong emergence is relevant. In the literature, the existence of emergence is controversial[48]; it is possible that the concept is only required because the causal explanations for certain phenomena have not yet been found [Janich/Duncker, 2011, p. 23]. Figure 7.5 presents an example of emergence in swarms. The non-deterministic movement of fish is temporarily and structurally unpredictable and consists of many interactions among many agents. Nevertheless, this fish school forms a ball-like formation.
It appears that the concept of emergence has remained unused and rarely discussed in the literature on swarm intelligence, although it is a key concept in AI [Brooks, 1991]. Emergence is mentioned in the literature as a biological aspect of swarms [Garnier et al., 2007], in distributed AI for complex optimization problems [Bogon, 2013, p. 19],in the context of software systems [Bogon, 2013, p. 19] (citing [Timm, 2006]) and as emergent computation [Forest, 1990]. Contrary to Forest, who assumes that only cooperative behavior can lead to emergence [Forest, 1990, p. 8], this works shows that egoistic behavior of a swarm can lead to emergence as well (see chapter 8). With regard to swarms, emergence should be a key concept. The four factors leading to emergence in swarms are

I. Randomness
II. Temporal and structural unpredictability
III. Multiple **non-linear** interactions among **many** agents
IV. Irreducibility

[Bonabeau et al., 1999, p. 23] agrees with [Ultsch, 1999, 2007] regarding the first factor: "*Randomness* is often crucial, since it enables the discovery of new solutions, and *fluctuations* can act as seeds from which structures nucleate and grow." Here, an algorithm is considered to have the property of *randomness* if it uses a source of random numbers in its calculations (non-determinism) [Ultsch, 2007]. The power of randomness is evident in Schelling's segregation model (Fig 3.).
The second factor, *unpredictability* [Ultsch, 2007, O'Connor/Wong, 2015], is incompatible with the PSO approach, in which an objective function is optimized [Martens et al., 2011] and, therefore, predictable assumptions are implicitly made regarding the structures of data sets in the case of unsupervised machine learning (see chapter 4 for further details on projection methods).
The third factor, multiple interactions among many agents, was identified by [Forest, 1990, pp. 1-2] for nonlinear systems. Although [Bonabeau et al., 1999] defines a requirement of multiple interactions for self-organization, the authors argue on page 24 that a single agent may also be sufficient. This is not the case for emergence, for which many elementary processes are mandatory [Beni, 2004; Ultsch, 1999]. Hence, ACO methods cannot exhibit the property of emergence Nonlinearity means that adding or removing interactions among agents or any agents

[47] There is no way to derive the property from any part, subset or partial structure of the system [Ultsch, 2007].
[48] For applications, the existence of emergence is irrelevant. Even if emergent phenomena can be causally explained, they can still be used in the future (see [Stephan, 1999] for discussion).

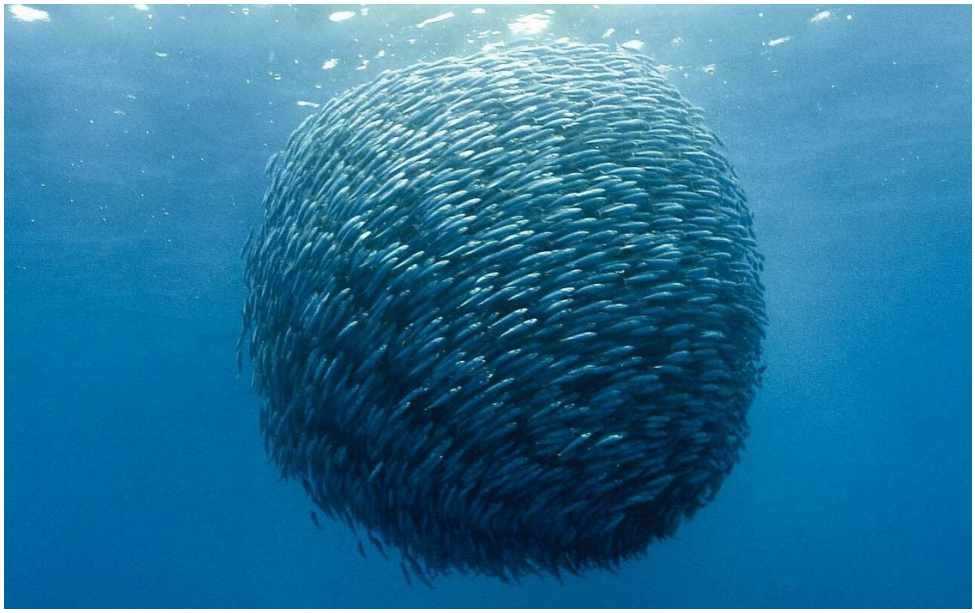

Figure 7.5: A fish swarm in the form of a ball [Uber_Pix, 2015]: an example of emergence in swarms. It illustrates the ability of a system to produce phenomena on a new, higher level.

themselves results in behavior that is linearly unpredictable. For example, the removal of one DataBot results in the elimination of one data point.

The fourth factor, *Irreducibility* [Kim, 2006, p. 555, Ultsch, 2007, O'Connor/Wong, 2015], means that the (novel) property cannot be derived from any agent (or part) of the system, but is only a property of the whole system. It is the ability of a system to produce phenomena on a new, higher level [Ultsch, 1999]. Vividly, it mark a distinction between the self-organization in Figure 7.2, where essentially a pattern of a snow flake could be derived by the physical properties and chemical bonds of H_2O and Figure 7.5, where the formation of a ball cannot be predicted from any fish itself.

The second missing link is a connection to game theory, in which the four axioms of self-organization — *positive* and *negative feedback*, *amplification of fluctuations* and *multiple interactions* — are apparent. Game theory was introduced by [Neumann/Morgenstern] in 1947. The purpose of game theory is to model situations[49] in which multiple players interact with each other or affect each other's outcomes [Nisan et al., 2007, p. 3] (*multiple interactions*). Here, the focus lies on a general, not zero-sum, n-person game [Neumann/Morgenstern, 1953, p. 85]. A game is defined as a scenario with n players i=1, …, n in which each player makes a choice [Neumann/Morgenstern, 1953, p. 84] (*amplification of fluctuations*[50]).

Let a game *G* be defined by *n* players associated with *n* non-empty sets $\Pi_1, \ldots, \Pi_n$, where every set Π_i represent all choices made by player i; then, the pay-off function is defined as

$$p = (p_1, \ldots, p_n): \Pi_1 \times \ldots \times \Pi_n \rightarrow \mathbb{R}^n \qquad (7.2)$$

[49] To be more specific, rational decision-making behavior in social conflict situations.
[50] Task switching.

The choices of each player determine the outcome for each player, and the outcome will, in general, be different for different players [Nisan et al., 2007, p. 9]. In a game, the payoff for each player depends on not only his own choices but also the choices of all other players [Nisan et al., 2007, p. 9] (*positive* and *negative feedback*). Often, the choices are defined based on a set of mixed strategies for each player. From the biological point of view, these mixed strategies may include the five main principles of collective behavior: *Homogeneity*, *Locality*, *Velocity Matching*, *Collision Avoidance*, and *Flock Centering* [Grosan et al., 2006].

In a game with n players, let the k choices of player i be defined by a set $\Pi_i = \{\pi_1^i, \dots \pi_\alpha^i, \dots, \pi_k^i\}$, where π_α^i indicates the i^{th} player's α^{th} choice; then, a mixed strategy $s_j(i) \in S_i$ for player i is defined by

$$s_j(i) = \sum_{\alpha=1}^{k(i)} c_\alpha(i)\pi_\alpha(i) \qquad (7.3),$$

where $\sum_{\alpha=1}^{k(i)} c_\alpha(i) = 1$ and all $c_\alpha(i) \geq 0$.

For noncooperative games, [Nash, 1951] proved the existence of at least one equilibrium point. Let $t_j(i) \in S_i$ be the mixed strategy that maximizes the payoff for player i; then, the Nash equilibrium is defined as

$$p_i\big(s(1), \dots, s(i-1), t_j(i), s(i+1), \dots, s(n)\big) = \max_{t_j(i) \in S_i} p_i\big(s(1), \dots, s(n)\big) \qquad (7.4)$$

if and only if this equation holds for every i [Nash, 1951]. The mixed strategy $t_j(i) \in S_i$ is the equilibrium point if no deviation in strategy by any single person results in a greater profit for that person. A Nash equilibrium is called *weak* if multiple mixed strategies $t_j(i) \in S_i$ for the same person exist in equation (4) that result in the same maximal payoff p_i, whereas in a *strong* Nash equilibrium, even a coalition of players cannot further increase their payoffs by simultaneously changing their strategies $t_j(i) \in S_i, i = 1 \dots m \leq n$, in (4). An illustrative example is the prisoner's dilemma [Poundstone, 1992]. Because of the interactions among the mixed strategies of all players that govern the payoff for a single player, the Nash equilibrium is not necessarily unique, and multiple different equilibria could exist.

8 Databionic Swarm (DBS)

This chapter introduces a new concept for the use of swarm intelligence. It makes use of insights from the previous chapter and proposes a projection method based on a swarm of intelligent agents called DataBots [Ultsch, 2000c]. This new swarm is called a polar swarm (Pswarm) because its agents move in polar coordinates based on symmetry considerations (see [Feynman et al., 2007, pp. 147-153, 745]). All parameters are automatically chosen according to, and directly based on, the appropriate high-dimensional definition of distance. The main idea of Pswarm is to combine the concepts of swarm intelligence and self-organization with non-cooperative game theory [Nash, 1950]. The main advance is the reliance on the concept of emergence [Ultsch, 2007] instead of the optimization of an objective function. This allows Pswarm to preserve structures in data sets that are characterized by discontinuity.

The extensive analysis of ant-based clustering (ABC) methods that has been performed in previous work allows the formulation of a precise mathematical definition of pheromonal stigmergy (a *scent*) [Herrmann/Ultsch, 2009]. The scent is defined in each neighborhood using an annealing scheme. The approach based on neighborhood reduction during the annealing process was invented by Kohonen [Kohonen, 1982b] and was used, for example, in [Demartines/Hérault, 1995; Hinton/Roweis, 2002; Ultsch, 1999]. In the context of swarm-based techniques, it was used for the first time in [Tsai et al., 2004]. Until now, finding the correct annealing scheme for a high-dimensional data set has remained a challenging task [Nybo et al., 2007]. The Pswarm algorithm utilizes randomness and the Nash equilibrium [Nash] of non-cooperative game theory to find an appropriate annealing scheme based on the data as given in the input space. For this purpose, the scent will be redefined as the payoff function[51].

Having projected the high-dimensional points into two dimensions using Pswarm in section 8.1, the author applies the insights from chapters 4 and 5, particularly with regard to the generalized U-matrix, to propose a three-dimensional topographic map with hypsometric tints [Thrun et al., 2016a] based on the high-dimensional distances and the density of the two-dimensional projected points. Drawing further insights from [Lötsch/Ultsch, 2014], a semi-interactive, but parameter-insensitive, clustering approach is possible. The framework as a whole is called Databionic swarm (DBS) and has only two parameters: the number of clusters and the type of clustering (connected or compact). The key feature of DBS is that neither an overall objective function for the process nor the type of clusters sought is explicitly defined at any point during the Pswarm process. Both parameters can be deduced from a topographic map of the Pswarm projection and a dendrogram. For DBS clustering and Pswarm projection the CRAN R package Databionic swarm was used [Thrun, 2017].

8.1 Projection with Pswarm

This section introduces the Polar swarm (Pswarm algorithm, which is the key foundation for the clustering performed in the DBS framework. Although the entire algorithm is used in an interactive clustering approach, Pswarm by itself may be used as a projection method. Because

[51] However, DataBots will still be described as “smelling” their surroundings.

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_8

this enables direct comparison with the swarm-organized projection (SOP) algorithm, Pswarm is introduced and discussed separately from DBS.
The analysis presented in the second section of this chapter strongly indicates that Pswarm outperforms SOP in terms of structure preservation by virtue of the property of emergence arising from its self-organizing collective behavior (see also chapter 10, section 3). In contrast to SOP and all other common projection methods [Venna/Kaski, 2007; Venna et al., 2010], Pswarm does not require any input parameters other than the data set of interest, in which case Euclidean distances are used in the input space. Alternatively, a user may also provide Pswarm with a matrix defined in terms of a particular dissimilarity measure, which is typically a distance but may also be a non-metric measure.

8.1.1 Motivation: Game Theory

The purpose of game theory is to model situations in which multiple players interact with each other and/or affect each other's outcomes [Nisan et al., 2007, p. 3]. The author of this thesis focuses on a general, not zero-sum**,** non-cooperative game of n players [Neumann/Morgenstern, 1953, p. 85] in which the choices each player makes determine the outcome for each player [Nisan et al., 2007, p. 9]. For this kind of game, Nash proved the existence of at least one equilibrium point [Nash, 1951]. The payoff for each player depends on not only his own choices but also the choices made by all other players [Nisan et al., 2007, p. 9]. Often, these choices are defined based on a set of mixed strategies for each player.
The key idea of Pswarm is to redefine a game as one annealing step (epoch), the players as DataBots, and the scent as a payoff function and to find an equilibrium for each game. In the context of Pswarm, the game consists of rules governing the movement of the DataBots, which is defined by the grid, the neighborhoods and the payoff function. Each DataBot searches for its strongest payoff by either moving across the grid or staying in its current position. A new game (epoch), which is defined based on the considered neighborhood radius R, begins once an approximate equilibrium is achieved, i.e., once no movement of any DataBot leads to a stronger or better payoff for any other DataBot any longer (weak Nash equilibrium). This approach leads to a data-driven annealing scheme with steps which are not defined by parameters, contrary to SOP (e.g. *threshold_max,* i_max in Listing 7.1), CCA and ESOM (e.g. number of epochs) as well as NeRV[52].

8.1.2 Symmetry Considerations

If we consider DataBots that occupy space in two dimensions, such as spheres or atoms, two points must be considered: first, no two DataBots are allowed to be in the same spot at the same time (*collision avoidance*), and second, a hexagonal lattice (tiling) is the densest possible packing of identical spheres in two dimensions [Hunklinger, 2009, p. 65]. Every such sphere represents a possible position for a DataBot. To ensure that the two-dimensional output space is used most efficiently, a hexagonal lattice tiling (*grid*) is used in Pswarm. To avoid problems associated with the surface of the grid, such as the positioning of DataBots near the border, the grid must have periodic boundary conditions and consequently must possess full *translational symmetry* [Haug/Koch, 2004, p. 34]. If the third dimension (e.g., as in a crystal) is disregarded, this two-dimensional grid can be represented by a three-dimensional torus [Pasquier, 1987], which

[52] e.g. iterations, cg_steps, cg_steps_final in [Nybo/ Venna, 2015, Thrun et al., 2017].

is hereafter referred to as a *toroidal grid*. This means that the borders of the grid are cyclically connected. The periodicity of the grid is defined by its size in terms of the numbers of lines *L* and columns *C*. If the grid were *planar* (not toroidal), undesired boundary effects could affect the outcome of any method.

Boundary effects are effects related to the borders of the output space in which the patterns of interactions across the borders of the bounded region are ignored or distorted, giving rise to shape effects, such that the shape imposed on the planar output space affects the perceived interactions between phenomena (see [McDonnell, 1995]). For example, if the output space is planar, it is unknown whether a projected point on the left border is similar (or dissimilar, in this case) to a projected point on the right border. It could be that the projection method is constrained to split similar points (with regard to the input space) in the output space. Another example is the distorted interactions between DataBots on the four borders when the output space is planar. Compared with a planar output space, a toroidal output space imposes fewer constraints on a projection (or clustering) method[53] and therefore enables a more optimized folding of the high-dimensional input space. A toroidal output space (in the case of Pswarm, a grid) possesses the advantage of translational symmetry in two dimensions, and in this case, the direction of a DataBot's movement is less important than its extent (length) because of the periodicity (of the grid).

In addition to the above considerations, the positions on the grid are coded using polar coordinates because the distances between DataBots on the grid will be most important in later computations of the neighborhoods and the annealing scheme. Consequently, based on the relevant *symmetry considerations,* a transformation of the Cartesian *(x, y)* coordinate system into polar coordinates $(r, \phi) \in O$ is proposed as follows:

$$r = x^2 + y^2 \qquad (8.1)$$

$$\phi = \tan^{-1}\left(\frac{y}{x}\right) * \frac{180}{\pi} \qquad (8.2)$$

Hereafter, *r* represents the length of a DataBot's movement (*jump*), and ϕ represents the direction of that movement.

Previously, the size of any grid (e.g., in SOP or emergent self-organizing map (ESOM)), as defined by the numbers of lines *L* and columns *C*, had to be chosen by the user. Choosing an incorrect size could result in a poor projection of the data. This was noted in previous works describing DataBot approaches prior to the development of the SOP algorithm [Kohlhof, 2010]. By contrast, in Pswarm, the grid size is chosen automatically, subject to three conditions. Let $\tilde{D}$ be an upper triangle of the matrix of the input distances, let *N* be the number of DataBots, let α be the number of possible jump positions, let $\beta \in (0.5,1]$ be a scaling factor, and let p_{99} and p_{01} denote the 99-th and first percentiles, respectively, of the distances; then, the conditions for determining the grid size are

$$\frac{\sqrt{C^2 + L^2}}{1} \geq \frac{p_{99}(\tilde{D})}{p_{01}(\tilde{D})} =: A \qquad (I)$$

$$L * C \geq \alpha * N \qquad (II)$$

[53] To the author's knowledge, only the emergent self-organizing map (ESOM) and the swarm-organized projection (SOP) method offer the option to switch between planar and toroidal spaces (see [Ultsch, 1999], [Herrmann, 2011, p. 98]).

$$\frac{L}{C} = \frac{\beta}{1} \qquad (III)$$

These conditions result in the following bi-quadratic equation:

$$C^4 - A^2 * C^2 + \alpha^2 * N^2 = 0 \qquad (8.3)$$

$$z_{1/2} = A^2 \pm \frac{1}{2}\sqrt{A^4 - \frac{\alpha^2}{4}N^2}$$

$$=> C = \begin{cases} \frac{1}{\sqrt{2}}\sqrt{A^2 + \sqrt{A^4 - \frac{\alpha^2}{4}N^2}}, & A^4 \geq \frac{\alpha^2}{4}N^2 \\ approximation, & A < \frac{\alpha^2}{4}N^2 \end{cases} \qquad (8.4)$$

The first condition ensures that the shortest and longest distances of interest are assignable to grid units. It defines the possible resolution of high-dimensional structures in the grid. The second condition ensures that there are sufficient available positions to which a DataBot can jump. The third condition causes the grid to be more rectangular than square because in the case of SOMs, "rectangular maps outperform square maps" [Ultsch/Herrmann, 2005]. The first two conditions are used to formulate the bi-quadratic equation under the assumption of equality (see Eq. 8.4). If the equation has no solution for the case of $A^4 < \frac{\alpha^2}{4}N^2$, then conditions I and III are used to generate approximate solutions. The scaling factor β is arbitrary and used only to ensure a solution in the case of approximation but it is not a parameter which has to be chosen. In this solution space, a solution that fulfills condition II is chosen.

8.1.3 Algorithm

Several previously developed ideas are applied in Pswarm: scent[54] [Herrmann/ Ultsch, 2008a], DataBots [Ultsch, 2000c] and the decreasing neighborhood radius proposed for DataBots by [Kämpf/Ultsch, 2006]. The decrease in the radius is based on the data and is not predefined by parameters, which was a goal of [Herrmann, 2011], where it was called self-adaptation. The underlying idea of the decreasing radius approach is to promote self-organization, first of a global structure and then of local structures [Kämpf/Ultsch, 2006].

The intelligent agents of Pswarm operate on a toroidal grid where the positions are coded using polar coordinates, $i_\phi(r) \epsilon\ O$. This permits the DataBots' movement, the neighborhood function and the annealing scheme to be precisely defined. The numeric vector z_j associated with each DataBot b_j represents its distances from all other DataBots in the input space I. The output-space distances are coded using only the polar coordinate *r*. The size of the squared-distance matrix D is defined by the number of DataBots.

After the assignment of initial random positions on the grid O (and therefore random output distances) to the DataBots in Listing 8.1**,** a data-driven decreasing of the radius R begins. In every iteration, a portion of the DataBots are allowed to jump if the payoff in one of their new positions is better (stronger) than that in their old positions. In other words, each DataBot is given a chance *c(R)* to try new positions on the grid.

The chance $c(R)$: $\mathbb{N} -> [0.05,0.5]$ is a continuous, monotonically decreasing linear function addressing the number of the DataBots which are allowed to search for a new position to jump

[54] Called topographic stress in [Herrmann/Ultsch, 2008].

to. Initially, many[55] DataBots are allowed to jump simultaneously to reproduce the coarse structure of the high-dimensional data set. However, as the algorithm progresses to address finer structures, only a small number[56] of DataBots may move simultaneously. The chance function depends on the number of DataBots and on the current radius R and consequently is based on the data itself.

In Pswarm, the length of a possible DataBot jump is not reduced during annealing[57]. The possible jumps of DataBots to new positions are drawn from a uniform distribution; therefore, the probability of selection is the same for all possible jumps, from a jump to zero to a jump to R_{max} in any direction. The direction of a jump to a new position is chosen separately from among all positions corresponding to an equal jump length. This approach prevents local minima from causing the DataBots to become stuck in an incorrect cluster because the length of their jump is smaller than half of the cluster's diameter. No DataBot is allowed to jump to an occupied position. Each DataBot may choose one of the four best different positions ($\alpha = 4$) in different directions to which to jump if it is sampled for jumping. This approach ensures a high probability that every sampled DataBot will find a free position.

function Positions O=Pswarm(matrix D(l, j))

for all $z_i \in I$*: assign an initial random polar position* $i_\phi(r) \in O$ *on the grid*

to generate N DataBots $b_i \in B$

for R={Rmax=Lines/2,…,Rmin} do

calculate chance c(R)

Repeat *for each iteration*

$c = sample\ (c(R), B)$

$m_k(c) = uniform(1, Rmax)$, *with k=1,…,*$\alpha$, $m_k(c) \in O$

$l(c) = \underset{j \in \{i, m_k(c)\}}{\operatorname{argmax}} \left(\lambda\left(b_j, R\right)\right)$

$l(!\,c) = i$

$S = \sum_{l=1}^{N} \lambda_l(b_l, R)$

Until $\frac{\partial S(e,\lambda(R))}{\partial e} = 0$

return O in Cartesian coordinates

end function Pswarm

Listing 8.1: The Pswarm algorithm consisting of *N* DataBots. New possible positions are depicted with $m_{k(i)}(c)$ where k indicates up to the number of α polar positions $i_\phi(r)$ chosen with an equal chance in the range from 1 up to *Rmax* (*uniform*) relative to the old position *i* and the old position with i of a DataBot which has a chance c to jump. After the decision to jump or not to jump the position is depicted with l(c). All other DataBots do not search for a new position depicted with $!\,c$ and remain on their old position i. The data-driven annealing scheme (repeat/until) is parameter free due to the application of the Nash equilibrium of game theory (see 8.1.6).

[55] However, no more than half of the DataBots are allowed to search for a new position.
[56] At the end exactly five percent of all DataBots.
[57] Unlike in the SOP algorithm.

8.1.4 Data-driven Annealing Scheme

Let each annealing step be defined as an epoch *e*; then, a new epoch begins (and a game[58] ends) if the radius *R* is reduced by the condition defined below.

Let $r(j,l)$ be the one-dimensional distance from $\mathrm{l} \in O$ to $\mathrm{j} \in O$ in polar coordinates (r,ϕ) as specified by the radius R_e; then, the neighborhood function "Cone" is defined as

$$h_R: R_e \to [0,1]:$$

$$h_R = \begin{cases} 1 - \frac{r(j,l)^2}{\pi R_e^2}, & iff\ \frac{r(j,l)^2}{\pi R_e^2} < 1 \\ 0, & otherwise \end{cases} \qquad (8.5)$$

where R_e is the radius of the neighborhood during epoch *e*.

Let *D(l, j)* be the distance between $\mathrm{x_l}, \mathrm{x_j} \in \mathrm{I}$, and let $r(j,l)$ be the one-dimensional radial distance in two-dimensional polar coordinates (r,φ) in the output space O; then, in Pswarm, the scent around a DataBot b_j is redefined to

$$\lambda_e(b_j, R_e, S_0) = \begin{cases} S_0 - \frac{\sum_{\mathrm{l}\in I} h_R(r(j,l)) * D(j,l)}{\sum_{\mathrm{l}\in I} h_R(r(j,l))}, & iff \sum_{\mathrm{l}\in W} h_R(r(j,l)) > 0 \\ S_0, & otherwise \end{cases} \qquad (8.6)$$

where

$$S_0 = \sum_j \left|\lambda(b_j, R_{max}, 0)\right| \qquad (8.7)$$

Following the discussion in section 8.1.2, the scent $\lambda(b_j, R)$ is identified as the payoff function $\lambda_e(b_j, R)$: $\mathbb{R}_0^+ \times O \to \mathbb{R}_0^+$ for a DataBot.

The high-dimensional input distances *D(l, j)* must be calculated only once, which is done prior to starting the algorithm, thereby reducing the computational cost. The computational cost of the algorithm does not depend on the dimension of the data set but does depend on the number of DataBots and the number of possible jump positions α. Additionally, Pswarm allows the conversion of distances or dissimilarities into two-dimensional points.

Let *e* be the current epoch, let R_e be the current neighborhood radius, and let $b_j \in B$ denote the DataBots; then, the sum of all payoffs is the current global happiness, which may be called the stress[59] $S(e, R_e)$, and is defined as

$$S(e, R_e) = \sum_j \lambda_e(b_j, R_e) \qquad (8.8)$$

The neighborhood is reduced if the derivative of the current global happiness is equal to zero:

$$\frac{\partial S(e, R_e)}{\partial e} = 0 \qquad (8.9)$$

which is called the *equilibrium of happiness* condition. The neighborhood radius R is reduced from R_{max} toward R_{min} with a step size of 1 if the derivative of the sum of all payoffs λ_e is equal to zero. This is the case if a (weak) equilibrium for all DataBots is found.

Because not all DataBots are allowed to jump simultaneously during a single iteration, as imposed by the function $sample\ (c(R), B)$, the DataBots are able to pay off their neighborhoods

[58] In the context of game theory.

[59] To simplify the comparison with SOP.

more often, thereby promoting the process of self-organization. By searching for an equilibrium, the net number of DataBots that would like to jump or are unhappy is irrelevant to the self-adaptive annealing process. Instead, the decision to shrink the neighborhood size or to proceed to the next epoch *e* is made based on a Nash equilibrium [Nash, 1950]. The criterion is clearly defined to correspond to the condition in which the global amount of happiness in the current epoch remains unchanged, which is defined as the *equilibrium of happiness*, $\frac{\partial S}{\partial e} = 0$.

8.1.5 Annealing Interval

Rmax is equal to Lines/2 if Lines<Columns to prevent self-interaction of the DataBots. If the radius R were to be greater than Lines/2, then the neighborhood of a given DataBot would overlap with itself because of the toroidal nature of the grid. Moreover, the probability density function for choosing a new position cannot be uniformly (or Gaussian) distributed in this case because border positions can be reached from two directions ϕ on a toroidal grid.

Rmin is determined by the size of the grid and the number of DataBots. It is set to a value that allows every DataBot to smell a minimum of 5% of the other DataBots if they are distributed uniformly[60]. This selection is inspired by an emergent phenomenon called an ant mill [Schneirla, 1971, pp. 281-283]: Army ants are an aggressive, nomadic species, incessantly moving around. Based on its payoff, every ant follows another ant in front of it. If the head of the ant colony runs into the tail of the colony, the ants form a so-called circle of death, because they keep moving until they die. This phenomenon would not occur if the ants were able to smell a region farther ahead of them.

8.1.6 Convergence

In game theory, for a game with egoistic agents, a solution concept exists called the Nash equilibrium [Nash, 1950].

Let (P, Λ) be a game with n DataBots b_i, i $= 1, \dots, N$, where P is a set of movement strategies and $\Lambda = \{\lambda_{e,i}(b_i, R_e = const) | i = 1, \dots, N\}$ is the payoff function evaluated for every grid position $w_i \in P_i$. Each DataBot chooses a movement strategy consisting of a probability associated with a position on the grid. Upon deciding on a position, a DataBot receives the payoff defined by the scent. P is a set of mixed strategies that are chosen stochastically with fixed probability in the context of game theory. Nash proved that in this case, the following equilibrium exists:

$$\forall i. w_i, b_i \in P: \lambda_i(b_i') \geq \lambda_i(b_i) \qquad (8.10)$$

The strategy b_i is the equilibrium, for which no deviation in strategy (position on the grid) by any single DataBot results in a greater profit for that DataBot. In the case of Pswarm, the Nash equilibrium is called weak because there may be more than one strategy with the same payoff for some DataBots. Because of the existence of this equilibrium, the Pswarm algorithm will always converge.

[60] Rmin (and Rmax) are chosen automatically by the Pswarm algorithm based on the gird size and consequently based on the data.

8.2 Comparing Pswarm with a Previously Developed Approach

Although the entire algorithm is used in an interactive clustering approach that does not require any sensitive input parameters, in this section, Pswarm is treated as an independent projection method and is compared with swarm-organized projection (SOP, see also chapter 10, section 3).

It will be demonstrated that changing the coordinate system from Cartesian to polar coordinates enables precise and practical definitions of neighborhoods, stigmergy and distances in the output space. With this approach, by using the Nash equilibrium [Nash, 1950] and modifying the DataBots' movements, it is possible to deduce a parameter-free and data-driven annealing scheme. This section will show that the self-adaptive annealing scheme of SOP requires important parameters and is, in fact, not always self-adaptive, as opposed to the Pswarm algorithm.

8.2.1 Neighborhood Definition

The main problem with regard to SOP lies in the neighborhood definition and annealing scheme of [Ultsch/Herrmann, 2010] and [Herrmann, 2011], as shown in Figure 8.1.

Because the lattice tiling is rectangular (quad grid), as is justified for Cartesian coordinates by [Ultsch/Herrmann, 2005], the neighborhoods are square and not round; this was explicitly defined in [Herrmann, 2011, p. 46] and remains unchanged in the SOP algorithm [Herrmann, 2011, pp. 64-70], and it is relevant to the scent λ (as defined in chapter 7.1 in Eq. 7.1).

In SOP, the following applies $d_1(l,j) = d_2(l,j)$, where these distances denote the lengths of jumps between *l, j=x, y* in Cartesian coordinates. This means that the probability of selecting a diagonal position for a DataBot jump is equal to that of selecting a horizontal/vertical position in the SOP lattice because the two-dimensional Gaussian neighborhood consists of two Gaussian functions, from which the vertical and horizontal coordinates are drawn separately to determine the chosen lattice positions: $N(m(x), s = \sigma = R) + N(m(y), s = \sigma = R)$.

For the choice of new positions for the DataBots, Hermann proposed that the selection probability should a Gaussian [Herrmann, 2011, p. 64], where the center is the current position of the DataBot, m(x, y), and the standard deviation s [Ultsch/Herrmann, 2010, p. 3] is equal to the radius R. In [Ultsch/Herrmann, 2010], a two-dimensional Gaussian distribution $N^2(m, s)$ was mentioned, but a practical solution to the problem of how to implement a two-dimensional Gaussian distribution on a discrete lattice was not addressed [Herrmann, 2011]. Moreover, the neighborhood considered in [Herrmann, 2011, p. 64] was defined only on a finite lattice.

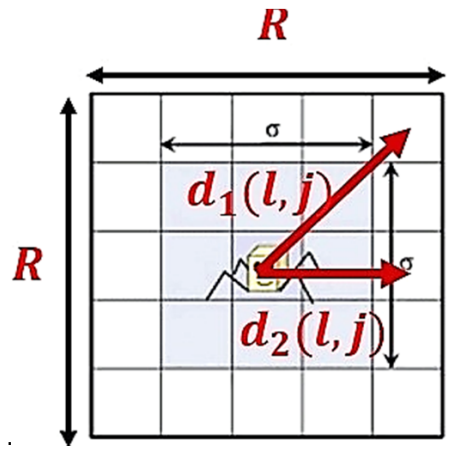

Figure 8.1: Neighborhood definition in the (rectangular) lattice tiling of a square shape of the SOP algorithm, adapted from [Herrmann, 2011, p. 47]. All positions defined at distances of less than or equal to r=2 are shown. Independent of the coordinate system, the SOP lattice is rectangular, with a size of (*L, C*).

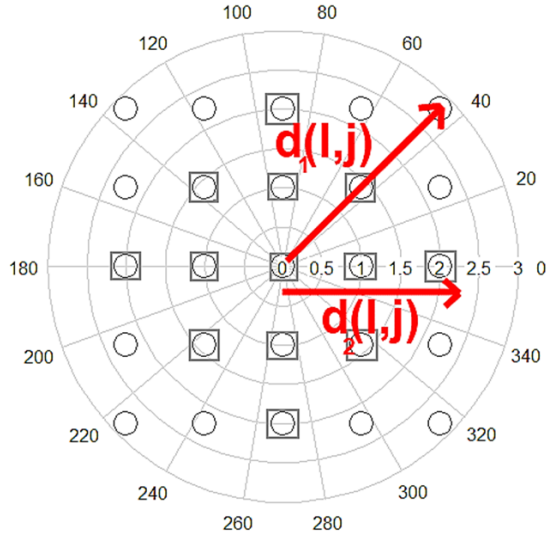

Figure 8.2: A similar rectangular lattice tiling of a square shape in polar coordinates for comparison[61]. In Pswarm, it applies $d_1(l,j) \neq d_2(l,j)$ for j, l=r, ϕ in polar coordinates. All positions at distances smaller than or equal to r=2 are marked by gray squares. In this case, the neighborhood (Eq. 8.5) depends on a precise one-dimensional grid distance, and for Gaussian neighborhoods, jump positions can be drawn from $N(m(r), s = R)$. Independent of the coordinate system, the Pswarm (hexagonal) grid has a rectangular shape of borders, with a size of (*L, C*).

On a toroidal grid or lattice (tiling), such a neighborhood will always overlap itself because Gaussian functions are never equal to zero. No solution for the case of a toroidal lattice was offered in [Herrmann, 2011]. Instead, in practice, the choice of a new DataBot position in the SOP algorithm is made by drawing separately from one normal distribution for the x coordinate

[61] In reality, Pswarm uses a hexagonal tiling instead of a rectangular tiling referenced as a grid.

and another normal distribution for the y coordinate, where the means are the corresponding coordinates of the current position and the standard deviations are equal to the radius[62] R. However, the following inequality applies:

$$N(m(x), s) + N(m(y), s) \neq N^2(m(x, y), s) \qquad (8.11)$$

Consequently, diagonal jumps are equal in length to horizontal and vertical jumps. However, [Bauer et al., 1999] argues that in a rectangular lattice, diagonal neighbors cannot be regarded as nearest neighbors. Moreover, the Gaussians overlap at the origin.

Based on *symmetry considerations,* a transformation from the Cartesian (x, y) coordinate system to the polar *(r, ϕ)* coordinate system is exploited in Pswarm.

This allows Pswarm to use a more precise neighborhood definition with sharp borders in Eq. 8.5, as illustrated in Figure 8.2, and makes the calculation of Euclidean distances in the two-dimensional output space unnecessary[63]. The neighborhood is defined only by the radius r of the polar coordinates. If the radius exceeds the borders of the toroidal grid, then the distance and jump length can be adapted using a modulus operation if drawn from uniform distributions. Allowing the maximum possible jump lengths prevents the algorithm from becoming trapped in local minima: if the jump length is too short, there is a possibility that the DataBots may be unhappy in their positions but unable to find new positions because no open positions exist.

In contrast to Pswarm, in SOP, the neighborhood definition for the scent λ remains vague. In [Herrmann, 2011, p. 63], it is stated that the development of SOP led to the revision of the ABC method based on Figure 8.1, where quadratic neighborhoods are explicitly defined [Herrmann, 2011, p. 46]. Still, this definition remained unchanged [Herrmann, 2011, pp. 64-70]. However, if the maximal radius is set to R>Lines/2 for Lines<Columns, then the Gaussian function F_R required to calculate the scent λ [Herrmann, 2011, p. 64] overlaps itself if no sharp borders are defined or if the grid or lattice is not finite (see chapter 7.1 Eq. 7.1). This overlap changes the weights of the output-space distances and the probabilities of choosing new positions to which to jump.

Additionally, the neighborhood of the lattice in which the DataBot is moving is defined by equal (square) diagonal and vertical jumps, but the two-dimensional distances on the lattice are defined as Euclidean distances (radial). These definitions are inconsistent with each other. Thus, the annealing scheme of the SOP algorithm is more square (jump length, position probability) than radial (output-space Euclidean distance). In summary, the use of Gaussian functions prevents the possibility of precisely defining the DataBot jump length and neighborhood, and worse, the jump length and neighborhood are not consistent with the output distances; see Figure 8.1.

More importantly, the radius R does not define a border for the SOP neighborhood; instead, it defines only the standard deviation of the density of a normal distribution. This results in very large neighborhoods without sharp borders. The adaptation of this neighborhood definition for a toroidal lattice was not addressed, and if the definitions of [Herrmann, 2011] were to be used on a toroidal lattice without modification, this would lead to significant mistakes.

Consequently, the definition of the scent λ is not consistent because the Euclidean output-space distance definition is inconsistent with the neighborhood definition.

[62] Taken from [Kohlhof, 2010] and Lutz Herrmann's 2011 Java implementation.

[63] A spherical coordinate system is the appropriate extension for a three-dimensional system.

Only a polar coordinate approach, such as that used in Pswarm, allows the selection of a neighborhood function h_R that precisely defines the neighborhood borders (Eq. 8.5). Moreover, the computational effort needed to calculate the output-space distances from one DataBot to all others is reduced in such an approach because it is sufficient to look up radii coded in hash tables.

8.2.2 Annealing Scheme

The second problem with the SOP algorithm lies in the annealing scheme itself, which is not self-adaptive, as is claimed in [Herrmann, 2011]. This is because it is governed by two magic numbers: a threshold in terms of the number of DataBots that are allowed to jump and the maximum number of iterations after which an epoch ends given that this arbitrary threshold is exceeded in every iteration. The term "magic" indicates that these numbers are not derived from data but instead must be carefully chosen by an experienced user.

Only if the number of DataBots that want to jump exceeds a certain threshold value, called a fixed point in [Herrmann, 2011], will another iteration of the current epoch start. Otherwise, a new epoch with a smaller radius begins. This threshold value is required in SOP because the following case was not sufficiently considered: Often, as a result of a jump of one DataBot, not only will the scent of that DataBot change, but so will those of all the other DataBots in its new neighborhood and, more importantly, its old neighborhood. Because all DataBots are allowed to jump simultaneously, the DataBots are unable to update their scents sufficiently quickly in response to the changes occurring around them before they jump themselves; the scent at a possible new position is compared with an outdated (incorrect) scent at the current position, because the scent at the current position will have changed as a result of the jumps of other DataBots. This may result in random jumping.

In addition, if the scents at their current positions become worse, other DataBots will become unhappy. Therefore, on the one hand, they should also be allowed to jump, but on the other hand, allowing these DataBots to jump could trigger a cyclic process in which the DataBots simply follow each other. There is also a possibility that when DataBots are unhappy with their current positions, they may be unable to find new ones. Either no open positions may exist, or the scents at all other positions in the small circle around the DataBot itself may be even worse. This occurs because in a Gaussian distribution, there is a very high probability of making only small jumps and an exponentially lower probability of making larger jumps.

To summarize, these problems are intrinsic to the SOP algorithm and are unrelated to the sparse probabilistic movements of the agents, as claimed by [Herrmann, 2011, p. 66].

Another problem with the annealing process in SOP is the assumption that the stress $S(\lambda, e)$ will be decreased only through iterations (Fig. 4.3 in [Herrmann, 2011, p. 69]) in which the DataBots move.

If the neighborhood function F_R is chosen to be a Gaussian distribution, then a smaller radius implies a reduction of the neighborhood function, i.e., $R_1 < R_2 \Rightarrow F_{R_1} < F_{R_2}$., because the standard deviation is defined by the radius. As shown by the curve in Fig. 4.3 in [Herrmann, 2011, p. 69], the sum of the scent[64] in a neighborhood (in Hermann's thesis, this is called the

[64] Defined in chapter 7.1.

sum of (topographic) stress) therefore also decreases because for lower values of the neighborhood function F_R, the scent[64] values and, consequently, the stress S must be lower:
$F_{R_1} > F_{R_2} => \lambda(R_1) < \lambda(R_2) => S(R_1) < S(R_2)$**.** Only if the iterations are within the same epoch (with a constant radius R) must a reduction in stress be driven by DataBot movement. Therefore, applying *argmin* between scent[64] values associated with different neighborhood radii results in random jumping of the DataBots.
Furthermore, the annealing scheme appears to reduce the stress S until convergence is reached (see Fig. 4.3 in [Herrmann, 2011, p. 69]). However, defining the scent[64] and $Rmin = 1$ for the SOP algorithm as proposed by Herrmann results in $\lambda = \infty$ if there are no other DataBots in the neighborhood of a jumping DataBot. Even worse, this could lead to random jumping if, for example, two simultaneously jumping DataBots can smell only themselves when changing positions or if a reduction in the scent is only an effect of a reduction in the number of DataBots in the neighborhood.
By contrast, in Eq. 8.6 the payoff $\lambda_e(b_j, R_e)$ considered in Pswarm was modified based on symmetry considerations, because the two-dimensional output-space distances are irrelevant if the coordinate system is polar. In this case, it is sufficient simply to use radii, and thus, it is not necessary to simulate radial neighborhoods by means of expensive computations using a Gaussian neighborhood function. Pswarm allows the definition of a sharp, radial, and deterministic neighborhood function (called Cone, Eq. 8.5) instead of the blurry, squarer than radial, and stochastic neighborhood of SOP.
In Pswarm, the "fixed point condition" of [Herrmann, 2011] is replaced with the equilibrium of happiness, $\frac{\partial S}{\partial e} = 0$ in Eq. 8.8. The use of the derivative makes it possible, during an epoch with a specific radius R, to find an iteration in which changes to the positions of some unhappy DataBots will not change the global happiness of all DataBots. In other words, an unhappy DataBot may jump to a new, more profitable position to become happier, but the DataBots surrounding its old position will simultaneously be left with less profitable positions and, in turn, become unhappier. This results in a kind of equilibrium in which, on the global scale of the toroidal plane[65], the DataBots are incapable of finding more profitable positions.
When the DataBots are not allowed to jump simultaneously, they are able to detect the payoffs related to other DataBots in their current positions before deciding to jump. By allowing all DataBots to jump in every iteration, as in SOP, the process of finding emergent structures could be delayed or even destroyed.
On a toroidal grid, setting the maximal neighborhood radius to the maximal distance on the grid results in self-interaction of the DataBots: the probabilities of choosing a new position will overlap for radii that extend beyond the closer edge of the grid (*R>Lines/2* if *Lines<Columns*). Moreover, the neighborhood of one DataBot will overlap with itself, which will result in an incorrect calculation of the payoff and disrupt the process of emergence. Furthermore, the (maximal) neighborhood radius R in SOP is determined based on the architecture of the lattice-shaped output space [Herrmann, 2011, p. 138], which was set to a constant value of 64x64 in the cited thesis regardless of the specific structures of the various data sets to be analyzed.

[65] This statement is only true if the possible jump length does not decrease with the neighborhood size.

Using Schelling's model in SOP is difficult because the dependence on chance, the data and the parameter settings causes an enormous number of iterations to be required [Hatna/Benenson, 2012] for the separation of the DataBots. Consequently, the number of iterations must be limited, and a threshold must be set on the number of jumping DataBots. Additionally, the attempt to find the minimum scent between two possible positions results in the problems discussed above. By contrast, Pswarm exploits the Nash equilibrium concept [Nash, 1950] based on the redefinition of scent as a payoff function λ_e and important changes to the neighborhood definition. This results in an annealing scheme that is based on the data.
In conclusion, SOP requires the user to choose a lattice size, two magic numbers for the annealing process and, in some cases, a minimal radius, whereas Pswarm does not. Additionally, the annealing scheme of Pswarm is fully radial with sound neighborhoods, whereas the neighborhood definition and annealing process of SOP are inconsistent with each other, which could prevent effective self-organization and, thus, emergence (examples in chapter 10.3).

8.2.3 Swarm Intelligence and Self-Organization

As described in the previous chapter, swarm behavior is characterized by five main principles [Grosan et al., 2006]: *Homogeneity*, *Locality, Collision Avoidance, Velocity Matching* and *Flock Centering*. In Pswarm, every agent is based on a DataBot, and the motion of each DataBot is influenced only by a well-defined neighborhood in which no two DataBots can be located in the same place at the same time. Hence, the first three main principles are obviously used. Velocity is defined as the rate of change in position with respect to time.
Considering fluctuations due to randomness, the average change in position is defined as $\Delta\bar{R} = \frac{1}{2}\left|0.5 - \frac{Lines}{2}\right| = \frac{Lines-1}{4}$ because the DataBots can jump with uniform probability to positions at distances ranging from 0.5 to $\frac{Lines}{2}$ units of length and the relevant time interval is one iteration (within an epoch). Therefore, on average, the agents in Pswarm exhibit velocity matching. Flock centering, in our case, refers to centering around more than one flock, if a flock is understood to have the figurative meaning of a group of similar agents. In summary, all five principles of swarm behavior are represented in Pswarm. For the simplified definition of intelligence reduced to behavior, as presented in the last chapter, Pswarm therefore uses *swarm intelligence*.
Self-organization relies on four principles [Bonabeau et al., 1999]: *positive* and *negative feedback*, *amplification of fluctuations* and *multiple interactions.* Fluctuations appear because of the random jump lengths and the random choices of new DataBot positions. Multiple interactions among DataBots are required for stigmergy in a given neighborhood in which various DataBots are present. Positive feedback and negative feedback are reflected in the choices of a DataBot to not jump when it is "happy" and to jump when it is "unhappy". Moreover, the number of DataBots cannot be reduced because each DataBot represents one data point in the data set. Consequently, self-organization is a property of Pswarm if the data set of interest contains more than 100 high-dimensional data points. Because of the randomness of the choice of possible jump positions, the system is temporally and structurally unpredictable, and Pswarm exhibits multiple interactions among many agents. The property of *irreducibility* is shown through the found compact and connected structures (chapter 10-12). Therefore, this system of DataBots possesses the property of emergence, as defined in chapter 7.3.

8.3 Clustering on a Generalized U*-Matrix

Chapter 4 introduces a generalized U*-matrix visualization called topographic map that can be used for any projection method. The U*-matrix represents high-dimensional density- and distance-based structures and is visualized as a topographic map with hypsometric tints [Thrun et al., 2016a]. Chapter 4 explains the connection between an approximation made by the simplified ESOM (sESOM) algorithm and an abstract U-matrix (AU-matrix) [Lötsch/Ultsch, 2014]. The clustering approach here uses the idea applied for the ESOM method that the abstract U*-matrix can be used for hierarchical clustering [Ultsch et al., 2016a].

Here, Pswarm, the AU-matrix concept and the proposed visualization are combined in the DBS clustering approach. In contrast to SOP and ESOM, this semi-interactive approach does not require any parameters other than the number of clusters and the cluster structure, which is either connected or compact (for details, see chapter 3). The number of clusters and the cluster structure can be estimated by counting the valleys in a topographic map and from a dendrogram. If the number of clusters and the clustering method are chosen correctly, then the clusters will be well separated by mountains in the visualization. Outliers are represented as volcanoes and can be interactively marked in the visualization after the automated clustering process.

The distances required for hierarchical clustering are defined by the AU-matrix, which was introduced in [Lötsch/Ultsch, 2014] for the U-matrix of a SOM. Here, the AU-matrix itself is defined by the Pswarm projection. In principle, the approach described in this section can be used for clustering based on any projection method because it is possible to generate a generalized U-matrix for any projection method (see chapter 5).

Let $\mathrm{G}(\mathrm{l},\mathrm{j},\mathcal{D})$ be the minimum of all possible path distances $\mathrm{p}_{\mathrm{j},1}$ between a pair of points $\{j,1\} \in O$ in the output space, as defined in chapter 2; then, the graph $\mathcal{D}$ is defined as the Delaunay graph weighted by the high-dimensional Euclidean distances between the points $\{j,1\} \in \mathrm{I}$ in the input space. In every direct neighborhood $H_j(k=1,\mathcal{D},O)$, all direct connections from the points *l* to the point *j* in the output space are weighted using the input-space distances D(l, j). In comparison to the ESOM clustering method proposed in [Ultsch et al., 2016a], here the shortest paths $\mathrm{G}(\mathrm{l},\mathrm{j},\mathcal{D})$ are calculated additionally using the algorithm of [Dijkstra, 1959]. Contrary to [Ultsch et al., 2016a], the DBS clustering is not based on density information coded in the P-matrix, because Pswarm itself is already able to project density-based structures (e.g. projection of EngyTime in chapter 10.3, Figure 10.7).

For example, in Figure 8.3, there are two well-separated clusters (green and blue), which the compact DBS clustering can detect in the dendrogram (Figure 8.4, left). In fact, the dendrogram could indicate also three or four clusters, but this is not verified by the visualization. If three or four clusters were chosen, the DBS clustering algorithm would not label points in the same cluster with the same color because they would not be well separated by mountains. The cluster heatmap shown in Figure 8.4 (right) verifies the clustering result of two clusters.

The outliers in a data set may be manually identified by the user. In this case, choosing the connected structure option for the clustering process would result in the automatic detection of all outliers. However, this option does not always lead to the detection of the main clusters in terms of the $\mathrm{G}(\mathrm{l},\mathrm{j},\mathcal{D})$ distances. A second example of outlier detection is presented in chapter 10 using the Tetragonula data set [Franck et al., 2004].

Figure 8.3: DBS visualization as a topographic map of the Target data set of [Ultsch, 2005a]. Two main clusters are shown; the cluster labeled in green has a higher density than the cluster labeled in blue. The outliers (orange, yellow, magenta and cyan) lie in volcanoes.

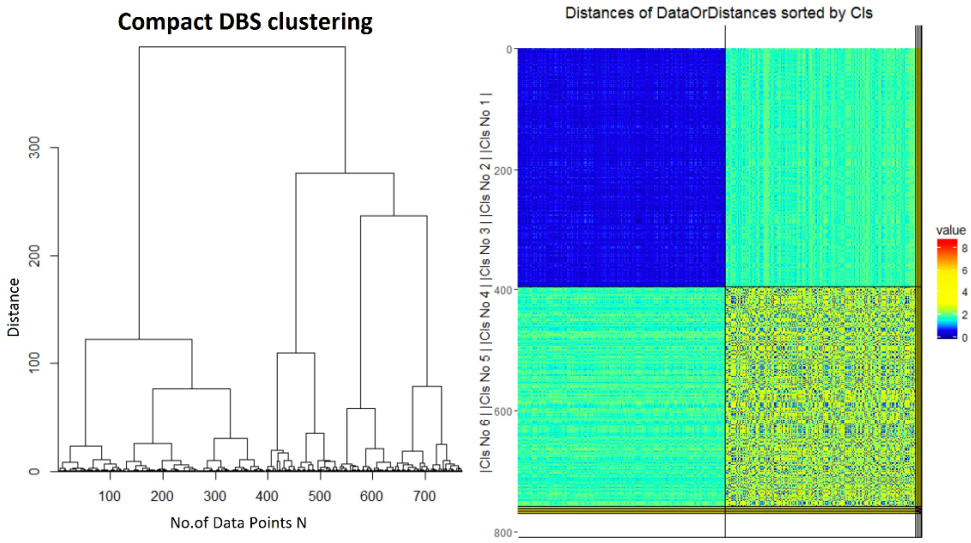

Figure 8.4: The dendrogram (left) of Target data set generated using the Ward algorithm shows either two or four clusters; however, in Figure 8.3, only two clusters are visible. The heatmap of the Target data set (right) shows two separated clusters with some outliers, because the intracluster distances are distinctively smaller than the intercluster distances.

9 Experimental Methodology

This chapter describes all the data sets used in the results chapter and the parameter settings for the various methods. In the final section, brief overviews of the Gene Ontology (GO) database and overrepresentation analysis (ORA) are provided. For general distribution analyses, the CRAN R package AdaptGauss [Thrun/Ultsch, 2015; Ultsch et al., 2015] was used. For the topographic map and island visualization the CRAN R package GeneralizedUmatrix was used [Thrun/Ultsch, 2017b]. For the ABC analysis the CRAN R package ABCanalysis was used [Thrun et al. 2015]. For DBS clustering and Pswarm projection the CRAN R package Databionic swarm was used [Thrun, 2017].

9.1 Data Sets

For the comparison of Pswarm as a projection method with the swarm-organized projection (SOP) algorithm, the original data sets of [Herrmann, 2011] were used. The artificial data sets of the Fundamental Clustering Problems Suite (FCPS) [Ultsch, 2005a] are summarized in Tab. 1 with regard to the cluster structures discussed in chapter 2.

> *"The FCPS comprises a collection of intentionally simple data sets with known classifications offering a variety of problems on which the performance of clustering algorithms can be tested. The data sets in the FCPS are specially designed such that the performance of clustering algorithms on particular challenges, for example, outliers or density- vs. distance-defined clusters, can be tested" [Ultsch/Lötsch, 2016, p. 4].*

All FCPS data sets have uniquely unambiguously defined class labels. For the error rate is defined as 1-Accuracy (Eq. 3.1 on p. 29) was is used as a sum over all true positive labeled data points by the clustering algorithm. The best of all permutation of labels of the clustering algorithm regarding the accuracy was chosen in every trial, because the labels are arbitrarily defined by the algorithms.
Additional data sets that are used in later chapters are also described below in alphabetical order. If these data sets are not discussed directly in chapter 10 and 11 than please see to Supplement C and D where the clusterings and the visualizations of DBS are shown. The hydrology data set and the pain genes data set are separately introduced in chapter 12.

9.1.1 Atom

> *"The Atom data set [Ultsch, 2005c] consists of two clusters in $\mathbb{R}^3$. The first cluster is completely enclosed by the second one and, therefore, cannot be separated by linear decision boundaries. Additionally, both clusters have different densities and variances. The Atom data set consists of a dense core of 400 points surrounded by a well separated, but sparse hull of 400 points. Both clusters are not linearly separable and many algorithms cannot construct a cohesive projection. The core is located in the center of the hull, which, for some methods based on averaging, makes it hard to solve it. The density of the core is much higher than the density in the hull. For data in the hull, some of the inner-cluster distances are bigger than the distance to the other clusters. The data set was not preprocessed" [Herrmann, 2011, pp. 99-100].*

9.1.2 Chainlink

The Chainlink data set [Ultsch, 1995; Ultsch et al., 1994] consists of two clusters in $\mathbb{R}^3$. Together, the two clusters form intricate links of a chain, and therefore, they cannot be separated by linear decision boundaries [Herrmann, 2011, pp. 99-100]. The rings are cohesive in $\mathbb{R}^3$; however, many projections are not. This data set serves as an excellent demonstration of several

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_9

challenges facing projection methods: The data lie on two well-separated manifolds such that the global proximities contradict the local ones in the sense that the center of each ring is closer to some elements of the other cluster than to elements of its own cluster [Herrmann, 2011, pp. 99-100]. The two rings are intertwined in $\mathbb{R}^3$ and have the same average distances and densities. The data set was not preprocessed [Herrmann, 2011, pp. 99-100]. Every cluster contains 500 points.

9.1.3 EngyTime

The EngyTime data set [Baggenstoss, 2002] contains 4,096 points belonging to two clusters in $\mathbb{R}^2$; the data set is typical for sonar applications with the variables "Engy" and "Time" as a two-dimensional mixture of Gaussians. The clusters overlap, and cluster borders can be defined only by using density information. There is no empty space between the clusters. The data set was not preprocessed [Herrmann, 2011, pp. 99-100].

9.1.4 Golf Ball

The Golf Ball data set "consists of an artificial data set with 4,002 points, resembling a 3-D view of a golf ball" [Ultsch/Lötsch, 2016, p. 3]. "The points are located on the surface of a sphere at equal distances from each of the six nearest neighbors" [Ultsch/Lötsch, 2016, p. 4]. This data set does not contain any natural clusters. The data set was not preprocessed.

9.1.5 Hepta

The Hepta data set [Ultsch, 2003a] is used to illustrate the general problems with quality measures (QMs) and projections from the perspective of structure preservation. The three-dimensional Hepta data set consists of seven clusters that are clearly separated by distance, one of which has a much higher density. The data set consists of 212 points, comprising seven clusters of thirty points each plus two additional points in the center cluster. The centroids of the clusters span the coordinate axes of $\mathbb{R}^3$. The density of the central cluster is almost twice as high as the density of the other six clusters. The structure of the data set is clearly defined by distances and is compact. The data set was not preprocessed.

9.1.6 Iris

> *"Anderson's [Anderson, 1935] Iris data set was made famous by Fisher [Fisher, 1936], who used it to exemplify his linear discriminant analysis. It has since served to demonstrate the performance of many clustering algorithms" [G. Ritter, 2014, p. 220].*

The Iris data set consists of data points in $\mathbb{R}^4$ with a prior classification and describes the geographic variation of *Iris* flowers. The data set consists of 50 samples from each of three species of *Iris* flowers, namely, Iris setosa, Iris virginica and Iris versicolor. Four features were measured for each sample: the lengths and widths of the sepals and petals (see [Herrmann, 2011, pp. 99-100]). The observations have "only two digits of precision preventing general position of the data" [G. Ritter, 2014, p. 220] and "observations 102 and 142 are even equal" [G. Ritter, 2014, p. 220]. The *I.* setosa cluster is well separated, whereas the *I.* virginica and *I.* versicolor clusters slightly overlap (see [Herrmann, 2011, pp. 99-100]). This presents "a challenge for any sensitive classifier" [G. Ritter, 2014, p. 220]. The data set was not preprocessed (see [Herrmann, 2011, pp. 99-100]).

9.1.7 Leukemia

The anonymized leukemia data set consists of 12,692 gene expressions[66] from 554 subjects and is available from a previous publication [Haferlach et al., 2010]. Each gene expression is a logarithmic luminance intensity (presence call), which was measured using Affymetrix technology. The presence calls are related to the number of specific RNAs in a cell, which signals how active a specific gene is. Of the subjects, 109 were **healthy**, 15 were diagnosed with acute promyelocytic leukemia (**APL**), 266 had chronic lymphocytic leukemia (**CLL**), and 164 had acute myeloid leukemia (**AML**). "The study design adhered to the tenets of the Declaration of Helsinki and was approved by the ethics committees of the participating institutions before its initiation" [Haferlach et al., 2010, p. 2530]. The leukemia data set was preprocessed, resulting in a high-dimensional data set with 7.747 variables and 554 data points separated into natural clusters, as determined by the illness status and defined by discontinuities (see chapter 2). Additionally, patient consent was obtained for the data set, in accordance with the Declaration of Helsinki, and the Marburg local ethics board approved the study (No. 138/16) [Brendel, 2016].

9.1.8 Lsun3D

The Lsun3D data set consists of three well-separated clusters and four outliers in $\mathbb{R}^3$; it is based on the two-dimensional Lsun data set of Moutarde and Ultsch [Moutarde/Ultsch, 2005]. Two of the clusters contain 100 points each, and the third contains 200 points. "The inter-cluster minimum distances, however, are in the same range as or even smaller than the inner-cluster mean distances" [Moutarde/Ultsch, 2005, p. 28]. The data set consists of 404 data points and was not preprocessed.

9.1.9 S-shape

"The plain s-curve data set is an artificial set sampled from an S-shaped two-dimensional surface embedded in three-dimensional space" [Venna et al., 2010, p. 462]. The authors claim that "an almost perfect two-dimensional representation should be possible for a non-linear dimensionality reduction method, so this data set works as a sanity check" [Venna et al., 2010, p. 462]. Here, it is more important that the data set does not possess any natural clusters. The data set consist of 2000 data points in $\mathbb{R}^3$ and was not preprocessed.

9.1.10 Swiss Banknotes

> *"The idea is to produce bills at a cost substantially lower than the imprinted number. This calls for a compromise and forgeries are not perfect" [G. Ritter, 2014, pp. 223-224]. "If a bank note is suspect but refined, then it is sent to a money-printing company, where it is carefully examined with regard to printing process, type of paper, water mark, colors, composition of inks, and more. Flury and Riedwyl [Flury/Riedwyl, 1988] had the idea to replace the features obtained from the sophisticated equipment needed for the analysis with simple linear dimensions" [G. Ritter, 2014, p. 224].*

The Swiss Banknotes data set consists of six variables measured on 100 genuine and 100 counterfeit old Swiss 1000-franc bank notes. The variables are the length of the bank note, the height of the bank note (measured on the left side), the height of the bank note (measured on the right side), the distance from the inner frame to the lower border, the distance from the inner frame to the upper border and the length on the diagonal. The robust normalization of Milligan and

[66] Process with which information from a gene is used in the synthesis of functional RNA or protein.

Cooper [Milligan/Cooper, 1988] is applied to prevent a few features from dominating the obtained distances [Herrmann, 2011, pp. 99-100].

9.1.11 Target

The Target data set [Ultsch, 2005c] consists of two main clusters and four groups of four outliers each. The first main cluster is a sphere of 363 points, and the second cluster is a ring around the sphere and consists of 395 points. The data set as a whole consists of 770 points in $\mathbb{R}^2$. The main challenge of this data set is the four groups of outliers in the four corners. The data set was not preprocessed.

9.1.12 Tetra

The Tetra data set, which is part of the FCPS, consists of 400 data points in four clusters in $\mathbb{R}^3$ that have large intracluster distances [Ultsch, 2005c]. The clusters are nearly touching each other, resulting in low intercluster distances.

9.1.13 Tetragonula

The Tetragonula data set was published in [Franck et al., 2004] and is available to the public in the R package prabclus:

> *"It contains the genetic data of 236 Tetragonula (Apidae) bees from Australia and Southeast Asia. The data give pairs of alleles (codominant markers) for 13 microsatellite loci. The 13 string variables consist of six digits each" [Hennig, 2014]. The format is derived from the data format used by the GENEPOP 4.0 software implemented by Rousset in 2010. "Alleles have a three digit code, so a value of "258260" on variable V10 means that on locus 10, the two alleles have codes 258 and 260. "000" refers to missing values" [Hennig, 2014].*

9.1.14 Cuboid

The uniform Cuboid data set "was constructed by filling a cuboid with uniformly distributed random numbers in the x, y and z directions" [Ultsch/Lötsch, 2016, p. 5]. It was introduced in this publication. "A group structure [is] clearly absent by construction" [Ultsch/Lötsch, 2016, p. 5]; thus, the data set does not possess any natural clusters. The data set consists of 1000 data points in $\mathbb{R}^3$ and was not preprocessed. Additionally, another data set was generated by filling the same cuboid with Gaussian-distributed random numbers in the x, y and z directions.

9.1.15 Two Diamonds

"The data consists of two clusters of two-dimensional points. Inside each "diamond" the values for each data point were drawn independently from uniform distributions" [Ultsch, 2003c, p. 8]. The clusters contain 300 points each. "[In] [e]ach cluster[, the] points are uniformly distributed within a square, and at one point the two squares almost touch. This data set is critical for clustering algorithms using only distances" [Moutarde/Ultsch, 2005, p. 28]. The data set was not preprocessed.

9.1.16 Wine

The Wine data set [Aeberhard et al., 1992] is a 13-dimensional, real-valued data set. It consists of chemical measurements of wines grown in the same region in Italy but derived from three different cultivars. The robust normalization of Milligan and Cooper [Milligan/Cooper, 1988] is applied to prevent a few features from dominating the obtained distances [Herrmann, 2011, pp. 99-100].

9.1.17 Wing Nut

> *"The Wing Nut dataset […] consists [of] two symmetric data subsets of 500 points each. Each of these subsets is an overlay of equal[ly] spaced points with a lattice distance of 0.2 and random points with a growing density in one corner. The data sets are mirrored and shifted such that the gap between the subsets is larger than 0.3. Although there is a bigger distance in between the subsets than within the data of a subset, clustering algorithms like K-means parameterized with the right number of clusters (k=2) produce classification errors" [Moutarde/Ultsch, 2005, pp. 27-28].*

The data set was not preprocessed.

9.1.18 World Gross Domestic Product (World GDP)

The World GDP data set of [Leister, 2016] was constructed by selecting the purchasing power parity (PPP)-converted gross domestic product (GDP) per capita for the years from 1970 to 2010 from the data published in [Heston et al., 2012] of 190 countries. The data were logarithmized, and countries with missing values were not considered. In the resulting data set, 160 countries remain.

Table 9.1: Structures of the clusters in the artificial benchmark sets of the FCPS [Ultsch, 2005a] as defined in Chapter 2.

Data Set	Cluster Structure
Atom	Connected, direction-based, varying density, non-linear separable
Chainlink	Connected, direction-based, non-linear separable
EngyTime	Connected, unidirectional, varying density
Hepta	Compact, spherical, high intercluster distance
Lsun3D	Compact, ellipsoidal, outliers
Target	Connected, direction-based, outliers
Tetra	Compact, spherical, low intercluster distance
Two Diamonds	Compact, spherical, borders defined by discontinuity
Wing Nut	Connected, direction-based, linear separable
Golf Ball	No natural clustering tendency

9.2 Parameter Settings

The parameter settings for the clustering algorithms, the projection methods and the QMs used in this thesis are as follows.

9.2.1 Quality Measures (QMs)

Freely available implementations of the trustworthiness and discontinuity (T&D) measures and the precision and recall (P&R) measures (see chapter 6.1) in C++ code were used [Nybo/Venna, 2015]. For all other measures, self-developed implementations were used. Every QM is available in our R package, projections, which also includes R wrappers for the C++ code for the T&D and P&R measures. Our density-based version of the Shepard diagram is also available in the R package projections. This package can be downloaded from CRAN.

9.2.2 Projection Methods

For the projection methods considered here (see chapter 4), we used freely available code which is summarized in the ProjectionBasedClustering CRAN package [Thrun et al., 2017]: for principal component analysis (PCA) [Pearson, 1901], we used the PCA software available in the R package stats [R Development Core Team, 2008]; due to technical limitations ICA was omitted in the analysis; for curvilinear component analysis (CCA) [Demartines/Hérault, 1995], the CCA source code [Alhoniemi, et al., 2005] was ported from MATLAB to R and for t-distributed stochastic neighbor embedding (t-SNE) [Van der Maaten/Hinton, 2008], we used Donaldson's t-SNE implementation. Also included in the evaluation of various projection methods were the Neighbor Retrieval Visualizer (NeRV) algorithm ([Venna et al., 2010]) as implemented in the freely available C++ code [Nybo/ Venna, 2015] called in R (Thrun et al., 2017b]), the Sammon mapping technique for multidimensional scaling (MDS) [Sammon, 1969] available from [R Development Core Team, 2008], and the emergent self-organizing map (ESOM) algorithm as implemented in the R package Umatrix [Thrun et al., 2016a] which reproduced the results of [Ultsch/Mörchen, 2005].

For every projection method, only the default parameters were used, as given here (see also [Thrun et al., 2017]): The ESOM algorithm was set with 20 epochs; a planar lattice; 50 lines; 80 columns; a Euclidean neighborhood function; and a linear annealing scheme with a starting radius of 25, an end radius of 1, a starting learning rate of 0.5 and an end learning rate of 0.1.

For the NeRV method, lambda was set to 0.5 (for DCE baseline with PCA initialization) and 0.1 (default); the optimization scheme was set with 20 neighbors, 10 iterations, 2 conjugate gradient steps per iteration, and 20 conjugate gradient steps in the final iteration; and the points were randomly initialized (default). PCA and Sammon mapping did not require any input parameters. For CCA, 20 epochs, an initial step size of 0.5, and a radius of influence of 3*max($std(data)$) were specified. The t-SNE method was set with a perplexity of 30,100 epochs and a maximum number of iterations of 1.000. Aside from ESOM, every projection method is available through standardized wrappers in our R package projections on CRAN. The NeRV source code was modified only as required for compatibility with the CRAN package Rcpp. The Delaunay classification error (DCE) measure is also available in our R package projections on CRAN.

9.2.2.1 Swarm-Organized Projection (SOP)

The SOP parameterization was chosen following Herrmann [Herrmann, 2011, p. 98], using a 64 x 64 toroidal lattice with Gaussian neighborhoods, as described above. Further parameter specifications included a maximum of 500 iterations per epoch (for a single radius) and a jumping DataBot threshold of 5%. In a given iteration, the DataBots were allowed to jump only if the number of DataBots that wished to jump was above this threshold. If only 5% or fewer of the DataBots could find a better position or if the maximum number of iterations was exceeded, the radius was reduced. The starting radius was set to the maximum possible distance in the output space as defined by [Herrmann, 2011, p. 65]. The source code was implemented in R by Kohlhof [Kohlhof, 2010] under the supervision of Lutz Hermann and the SOP algorithm was executed using version 3.2.3 of R on a 64-bit Windows 7 system. Only Euclidean distances were used for SOP, consistent with the settings defined by [Herrmann, 2011, p. 98] and the restrictions of the source code. For this reason, the GDP194 data set was excluded because this

data set requires the use of special dissimilarities [Herrmann, 2011, p. 100]. Moreover, it should be mentioned that Rmin was set to a value much larger than 1 for this data set, although the precise number was not recorded [Herrmann, 2011, p. 167].
Other functional code for SOP or its extension for very large data sets, swarm-organized quantization, was not available to the author[67]. A self-developed implementation based on the algorithm exactly as described in chapter 7 yielded worse results on the data sets compared with that of Kohlhof [Kohlhof, 2010] because of the problems discussed in chapter 8.

9.2.2.2 Pswarm

For Pswarm, there are no parameters to set. In the case of the Wine data set, the distances were changed to squared Euclidean distances because the resulting distance distribution yielded a better distinction between the intra- and intercluster distances (see supplement B). The data sets were compared using the generalized U-matrix technique for three-dimensional visualization, as described in chapter 5. The CRAN R package Databionic swarm was used [Thrun, 2017].
Notably, the three-dimensional topographic map with hypsometric tints that is referred to as the generalized U-matrix in this thesis is completely different from the gray-scale two-dimensional visualization of Hermann [Herrmann, 2011, p. 72], which was also called the generalized U-matrix.All source code was executed in R 3.3.1 [R project, , 2008] on a 64-bit Windows 7 system.

9.2.3 Common clustering algorithms

For the k-means algorithm, the CRAN R package cclust was used [Dimitriadou/Hornik 2017]. For the single linkage (SL) and Ward algorithms, the CRAN R package stats was used [R Development Core Team, 2008]. For the Ward algorithm, the option “ward.D2” was used, which is an implementation of the algorithm as described in [Ward Jr, 1963]. For the spectral clustering algorithm, the CRAN R package kernlab was used [Karatzoglou et al., 2016] with the default parameter settings: “The default character string “automatic” uses a heuristic to determine a suitable value for the width parameter of the RBF kernel”, which is a “radial basis kernel function of the “Gaussian” type”. The “Nyström method of calculating eigenvectors” was not used (=FALSE). The “proportion of data to use when estimating sigma” was set to the default value of 0.75, and the maximum number of iterations was restricted to 200 because of the algorithm’s long computation time (on the order of days) for 100 trials using the FCPS data sets. For the mixture of Gaussians (MoG) algorithm, the CRAN R package mclust was used [Fraley et al., 2017]. In this instance, the default settings for the function “Mclust()” were used, which are not specified in the documentation. For the partitioning around medoids (PAM) algorithm, the CRAN R package cluster was used [Maechler et al., 2017].

9.3 Gene Ontology (GO)

An ontology is a representation of knowledge in which the relationships *part of* and *is a* are visualized in a directed acyclic graph (DAG). For the analysis of pain genes, the GO database was accessed via R 3.3.1 [R Development Core Team, 2008]. In the GO database, knowledge

[67] Lutz Herrmann’s 2011 Java implementation is largely identical to that of [Kohlhof, 2010], but the source code could not be compiled.

about molecular functions, biological processes and the cellular components of genes is defined using a controlled vocabulary consisting of labels called GO terms, which are used to represent biological concepts [Ashburner et al., 2000]. These terms describe and unify the attributes of genes and gene products[68] in a species-independent manner. "The GO terms are ordered in a directed acyclic graph (DAG), in which the set of genes annotated[69] to a certain term (node) is a subset of those annotated to its parent nodes" [Goeman/Mansmann, 2008]. Here, the important relationships between the nodes are of the "part of" type, resulting in a "top-down poly-hierarchy of GO terms" starting "at the root with terms with the broadest definition" and specializing "toward the leaves representing GO terms of the narrowest definition (details)" [Ultsch et al., 2016b]. Given a set of genes, ORA reveals the significance of a GO term that represents these genes or a subset of these genes [Backes et al., 2007].

9.3.1 Overrepresentation Analysis (ORA)

"In ORA, the most commonly used statistical test is based on the hypergeometric distribution or its binomial approximation ([…] among others). Let A denote a GO term or the set of genes annotated to A (with cardinality I_A), and let S denote the set of genes (with cardinality I_S) based on a certain criterion (i.e. differential expression) from a full gene list G (with cardinality I) in an experiment. The number of genes belonging to both S and A ($S \cap A$), denoted by n_A, indicates the representation of A in S. Under the null hypothesis that S and A are independent (i.e. the GO term is irrelevant to the gene cluster), n_A follows a hypergeometric distribution. The [p-value p] measuring the significance of association is the tail probability of observing n_A, or more genes annotated by A in S,

$$p = \sum_{k=n_A}^{min(I_A, I_S)} \frac{\binom{I_A}{k}\binom{I - I_A}{I_S - k}}{\binom{I}{I_S}} \qquad (9.1)$$

where $\binom{m}{n} = \frac{m!}{n!(m-n)!}$ is the binomial coefficient. Many software packages and webtools (Onto-Express, CLASSIFI, GoMiner, EASEonline, GeneMerge, FuncAssociate, GOTree Machine, etc.) have been developed based on the hypergeometric [p-value]. A detailed review can be found in Khatri and Drăghici [Khatri/Drăghici, 2005].

The hypergeometric [p-value] provides a straightforward measure of overrepresentation for each individual GO term. However, the major drawback of this approach is that it ignores the hierarchical structure in the GO DAG, which contains a substantial amount of information regarding the interactions among the GO terms" [Zhang et al., 2010, pp. 905-906].

For the ORA algorithm, the R package ORA was used [Lippmann et al., 2016].

9.3.2 Filtering via ABC Analysis

The resulting p-values p were filtered via ABC analysis (see chapter 5.3.2 on p. 49 for further explanation) [Ultsch/Lötsch, 2015]; thereafter, only the most important group A was considered for interpretation. For the ABC analysis algorithm, the CRAN R package ABC analysis was used [Thrun et al., 2015].

Here, it is argued that changing the threshold with respect to the significance of the p-value does not lead to better results. Aside from the problems discussed by Button and Nuzzo [Button et al., 2013; Nuzzo, 2014], the paramount goal of a gene analysis is to find GO terms with a

[68] Usually either Ribonucleic acid (RNA) or a protein

[69] For further details, see [Camon et al., 2003] and [Camon et al., 2004].

high effect strength. For this purpose, it is sufficient for the effect to be significant with regard to a commonly used (arbitrary) p-value threshold.

Let *E* be the strength of an effect as defined with respect to its p-value significance *p* (expressed as a percent), as follows:

$$E = -10\log(p) \qquad (9.2)$$

At first glance, the definition given in Eq. 9.2 is contradictory to the equation above (1).

On the one hand, the calculation of p-values based on the Fisher test with $p(I_A, I_S, k, I)$ requires four parameters; on the other hand, one would calculate the strength of an effect based on the relative difference between the expected value *e* and the observed value *o*, known as the fold change FC:

$$FC(k, e) = 2\frac{o - e}{o + e} \qquad (9.3)$$

Here, the p-values are calculated analogously to Backes [Backes et al., 2007], where the formula is called the hypergeometric test. However, the hypergeometric test is simply the Fisher test based on the hypergeometric distribution [Ultsch, 2014a]. The hypergeometric distribution is defined as

$$f(I_A, I_S, k, I) = \frac{\binom{I_A}{k}\binom{I - I_A}{I_S - k}}{\binom{I}{I_S}} \qquad (9.4)$$

Given this distribution, the expected value $e(f)$ is defined as

$$e(f) = \sum_{k=0}^{I_S} k\frac{\binom{I_A}{k}\binom{I - I_A}{I_S - k}}{\binom{I}{I_S}} = I_S\frac{I_A}{I} \qquad (9.5)$$

It can be shown that Eq. 9.2 is directly proportional to the definition of the expected number of genes in Eq. 9.5 [Ultsch, 2014a]. Therefore, the observed number of genes *o* are compared against a hypergeometric distribution (Eq. 9.4) around the value for the expected genes number of *e* in Eq. 9.5, and in the special case of ORA, the p-values imply more than merely significance.

One may ask why the calculation must be complicated if the fold change, as defined in Eq. 9.3, could be used. The disadvantage of the fold change is illustrated in the following equation:

$$FC(o, e) = 2\frac{o - e}{o + e} = 2\frac{c * o - c * e}{c * o + c * e} \qquad (9.6)$$

According to this equation, one expected gene compared with four observed genes yields the same value as 100 expected genes compared with 400 observed genes. Clearly, the effect strength here is not the same.

It could be argued that this problem could be solved by reducing the p-value threshold to a low level, such that only a few GO terms are represented in the DAG. However, one would be obliged to do this manually for every ORA calculation. Moreover, to the author's knowledge, every tool or package that uses GO terms or performs ORA calculations has a different version of the GO database. Hence, the p-value calculation has a measurement error that is difficult to specify. Furthermore, even if a tool used the database obtained directly from the Gene Ontology Consortium, there is an even stronger source of measurement error: every list of genes I_S to be

analyzed was obtained based on microarray experiments with arbitrary thresholds or probe intensities (for a detailed discussion, see [Khatri et al., 2012, p. 3]).
Here, with regard to the definition of the effect strength given in (Eq. 9.2), it is assumed that the magnitudes of the p-values do not change regardless of measurement errors. This is the reason for taking the logarithm of the p-value in (Eq. 9.2). Moreover, Figure 9.1 shows the correlation between the fold change FC (Eq. 9.3) and the effect strength E (Eq. 9.2) for a given interval of the number of annotated genes per GO term. Consistent with Ultsch [Ultsch, 2014a], it is argued here that in ORA, the p-values are directly proportional to the effect sizes.
After setting the p-value threshold to *0.05*, which is a generally accepted level of significance, and calculating the corresponding GO terms, the results of an ABC analysis of the effect strengths as given by (2) can be obtained. The relevant GO terms are defined as those assigned to group A in the ABC analysis.

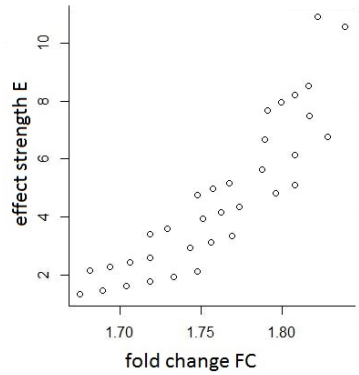

Figure 9.1: Scatter plot of the fold changes *FC* of Eq. 9.6 and the corresponding *E* value of Eq. 9.3 for numbers of annotated genes per GO term in the range [10,25] is proportional.

10 Results on Pre-classified Data Sets

This chapter has three sections. In the first section, the results of the Databionic swarm (DBS) clustering framework are compared with the given prior classifications for data sets from the Fundamental Clustering Problems Suite (FCPS) [Ultsch, 2005a]. The results for nine data sets analyzed using common clustering algorithms are compared in the first subsection. In the second subsection, the results for data sets with no natural clusters are compared (e.g., Golf Ball). Neighbor Retrieval Visualizer (NeRV) projection and Ward clustering indicate the presence of clusters, whereas DBS does not.
The second section compares Pswarm with other common projection methods using the Delaunay clustering error (DCE). The third section compares emergent self-organizing map (ESOM), swarm-organized projection (SOP) and Pswarm using topographic map visualizations based on the generalized U-matrix for the Wine, Iris, and Swiss Banknotes data sets as well as several FCPS data sets.

10.1 Comparison with Given Classifications

The FCPS [Ultsch, 2005a] is a repository consisting of ten data sets with known classifications. These data sets are intentionally simple enough to be visualized (in 2D or 3D) but nevertheless present a variety of problems that offer good tests of the performance of clustering algorithms [Ultsch/Lötsch, 2016]. The first Figure (10.1) shows the performance of several common clustering algorithms compared with DBS based on 100 trials. The performance is depicted using boxplots of the error rate, which is defined as one minus the accuracy and for which 50% is the level attributable to chance (see chapter 3, Eq. 3.1). Here, the common clustering algorithms considered are single linkage (SL) [Florek et al., 1951], spectral clustering [Ng et al., 2002], the Ward algorithm [Ward Jr, 1963], the Linde-Buzo-Gray algorithm (LBG-k-means) [Linde et al., 1980], partitioning around medoids (PAM) [L. Kaufman/Rousseeuw, 1990] and the mixture of Gaussians (MoG) method with expectation maximization (EM) [Fraley/Raftery, 2002] (also known as model-based clustering).
Aside from the number of clusters, which is given for each of the artificial FCPS data sets, only the default parameter settings of the clustering algorithms were used. ESOM/U-matrix clustering [Ultsch et al., 2016a] and DBscan [Ester et al., 1996] were omitted because no default clustering settings exist for these methods. k-means has the highest overall error rate, and spectral clustering shows the highest variance. The results for the other clustering algorithms vary depending on the data set. DBS has the lowest overall error rate. However, on the Tetra data set, it is outperformed by PAM and MoG; on the EngyTime data set, it is outperformed by MoG; and in the case of the Wing Nut data set, it is outperformed by spectral clustering. Additional statistical tests to Fig 10.1 can be found in supplement I. With the help of insights from chapter 3, Tab. 3101 lists the FCPS cluster structures alongside the algorithms with the best results in terms of the lowest error rate and variance for each data set.

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_10

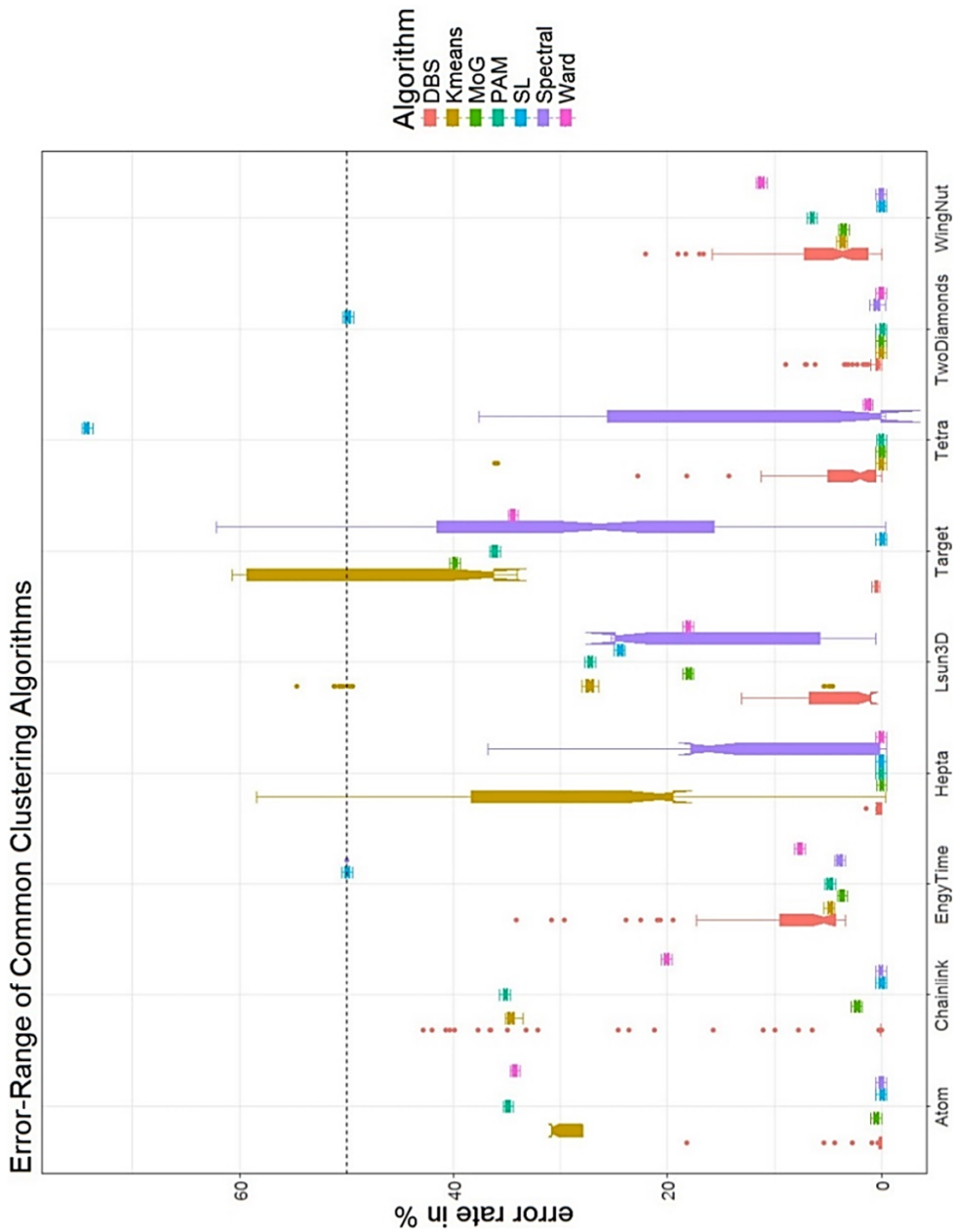

Figure 10.1: Error rate (see p. 107) of 100 trials of common clustering algorithms on nine FCPS data sets, shown as boxplots with the notch as median; chance level at 50%. The interactive clustering approach of DBS was not used here. Abbreviations: single linkage (SL), Linde-Buzo-Gray algorithm (LBG-k-means), partitioning around medoids (PAM), mixture-of-Gaussians clustering (MoG), Databionic swarm (DBS). Additional statistical tests can be found in supplement I.

10.1.1 Recognition of the Absence of Clusters

The Golf Ball data set (see chapter 9) does not exhibit natural clusters. Therefore, it is analyzed separately because, with the exception of SL and the Ward algorithm, the common clustering algorithms give no indication regarding the existence of clusters. This "cluster tendency problem has not received a great deal of attention but is certainly an important problem" [Jain/Dubes, 1988, p. 222]. Reproducing the results of [Ultsch/Lötsch, 2016], the Ward algorithm indicates six clusters, whereas SL indicates two clusters (Figure 10.2). As seen from the two dendrograms generated using DBS, the connected approach does not indicate any clusters, whereas the compact approach indicates four clusters (Figure 10.3). However, the presence of four clusters is not confirmed by the topographic map of DBS.

In Figure 10.4, the topographic maps of DBS with the NeRV are compared. The NeRV projection of the Golf Ball data set with $\lambda = 0.5$ (for the other parameters, see the R package projections), i.e., with precision and recall weighted equally, is shown in Figure 10.4 (top). The visualization of the NeRV projection strongly indicates a two-cluster structure, whereas the DBS projection does not (Figure 10.4, bottom). The compact DBS clustering divides the data points lying in valleys into different clusters and merges the data points into clusters through hills, resulting in cluster borders that are not defined by mountains.

The topographic map of DBS of the S-shape data set and the uniform and Gaussian Cuboid data sets (see chapter 9) are also shown in supplement D, Figure D.19. Neither data set contains any natural clusters; this is correctly visualized using the DBS approach.

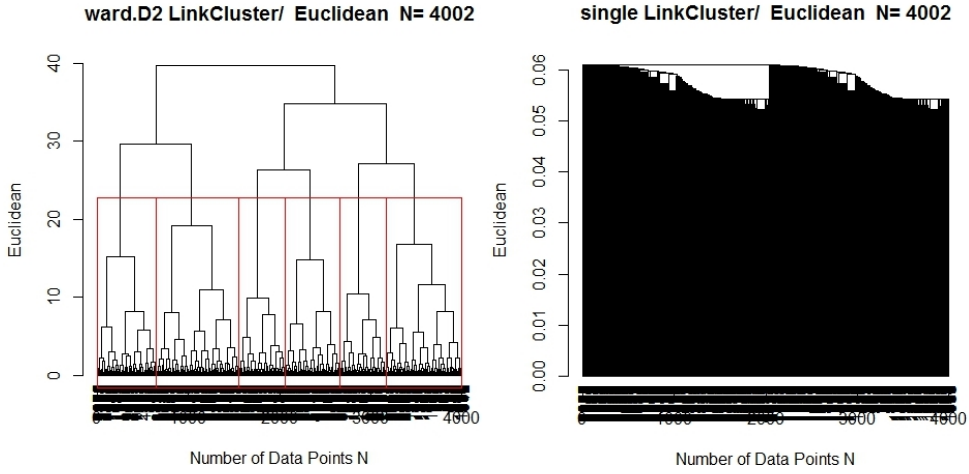

Figure 10.2: The dendrogram generated using the Ward algorithm indicates at least two clusters with a high intercluster distance. The SL dendrogram could indicate two clusters with a very low intercluster distance.

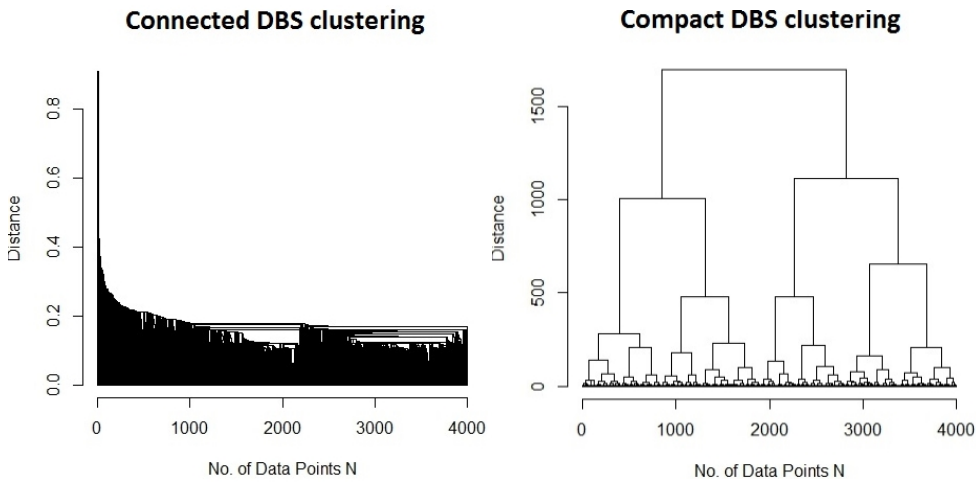

Figure 10.3: The two dendrograms generated using DBS. The connected DBS clustering does not indicate any structure whereas the compact DBS clustering indicates two or four clusters. The connected approach does not indicate any clusters, whereas the compact approach does indicate four clusters. However, Figure 10.4 shows that these clusters are inconsistent with the visualization.

10.2 Evaluation of Projections Using the Delaunay Classification Error (DCE)

Figure 10.5 shows the results for the DCE measure, relative to the baseline, for 100 trials of the common projection methods ESOM, NeRV, Sammon mapping (a multidimensional scaling (MDS) technique), curvilinear component analysis (CCA), principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). Positive values indicate higher errors compared with the baseline, whereas negative values indicate lower errors. The baseline is the NeRV projection with $\lambda = 0.5$ and PCA initialization; this baseline was chosen because the outcome of this initialization is deterministic (for the other parameters, see the R package projections). The parameter setting $\lambda = 0.5$ indicates that precision and recall are weighted equally. Every subfigure shows a robust mean estimate M and a robust standard deviation estimate S for the 100 relative DCEs. Notably, it is claimed that t-SNE projections are similar to NeRV projections with $\lambda = 1$ [Venna et al., 2010].

The linear method PCA and the MDS technique of Sammon mapping are unable to separate the connected structures of the Chainlink and Atom data sets based on their assumed neighborhood relations. This result confirms the assumptions made in chapter 4. By contrast, the CCA projections have difficulty separating compact structures based on intra- versus intercluster distances. However, not all focusing projection methods are able to separate connected structures, e.g., the t-SNE projections of Chainlink.

Without the U-matrix, the ESOM projection method distributes the points uniformly, which results in a higher DCE. The projections generated by t-SNE, Pswarm and NeRV with their default settings show high variances, although the variance in the accuracy of the DBS clustering results for these data sets is low (Figure 10.1).

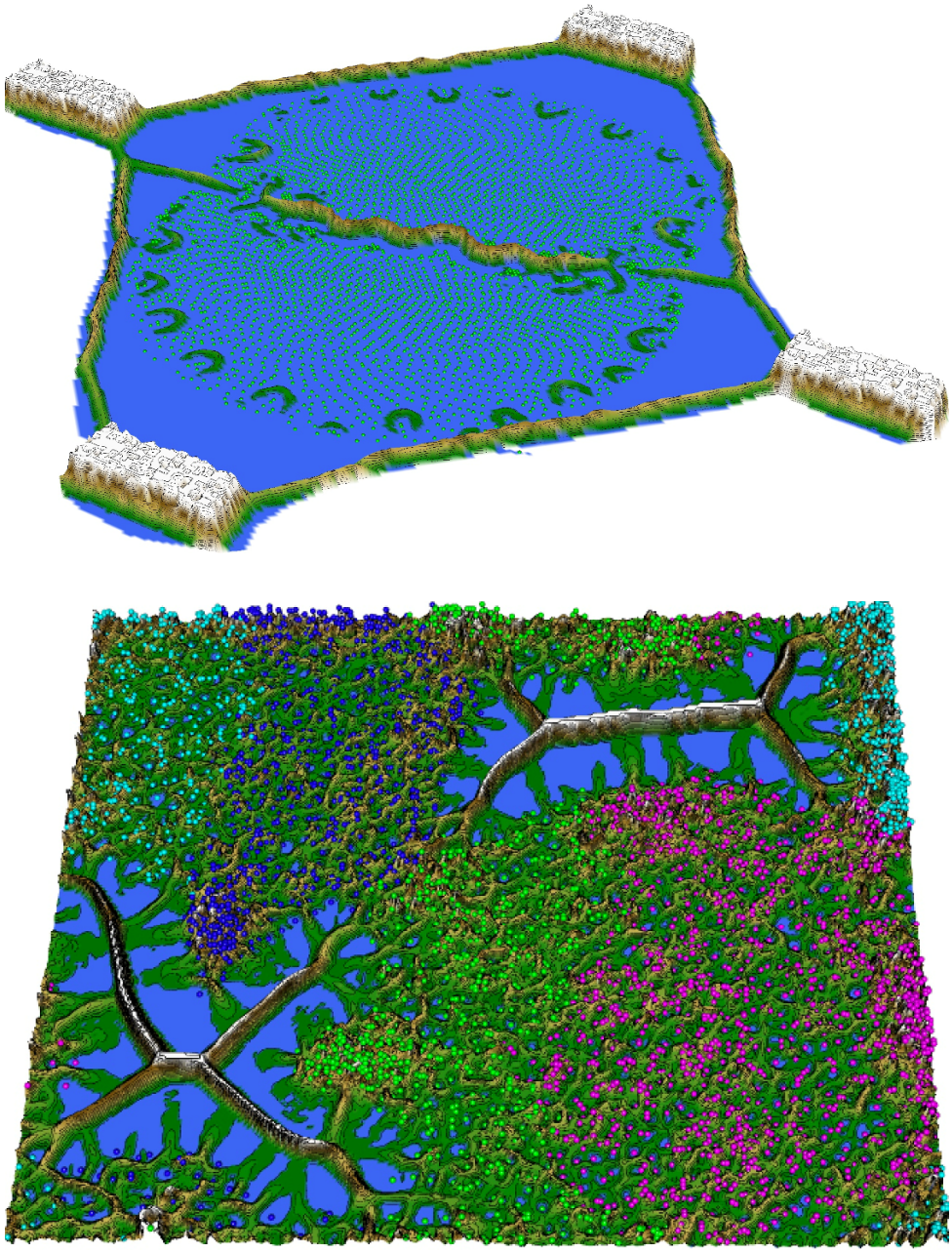

Figure 10.4: **Top**: Topographic map of the NeRV projection ($\lambda = 0.5$) of the Golf Ball data set indicates two well-separated clusters.
Bottom: The topographic map of the DBS projection and (compact) clustering of the Golf Ball data set. The projection does not indicate a cluster structure. The DBS clustering generates clusters that are not separated by mountains. No island can be extracted from the toroidal visualization.

Statistical testing was performed using the two-sample, one-sided Wilcoxon rank sum test with continuity correction [Hollander/Wolfe, 1973, pp. 68–75]. The DCE values for the Pswarm projections were compared with the projections obtained using the other methods with the "nearest"[70] ranges of DCE values "above" and "below" those of Pswarm (visually in the 90° rotated figures). In the former case, means that the DCE values of Pswarm are more negative (shifted to the left) compared with the DCE values of the projection method with the nearest range of values. Consequently, a significant result means that Pswarm's performance is considerably better. In the latter case, the DCE values of Pswarm are more positive (shifted to the right), and a significant result means that Pswarm's performance is worse than that of the projection method with the nearest range of DCE values "below" those of Pswarm. Statistical results regarding the performance of Pswarm in Figure 10.5 are as follows.

1.) **Atom**: The performance of Pswarm is significantly better than that of NeRV, with $W(100) = 1675$, $p < 0.001$, and worse than that of t-SNE, with $W(100) = 5795$, $p = 0.026$.
2.) **Hepta**: The performance of Pswarm is significantly better than that of CCA, with $W(100) = 1855$, $p < 0.001$, and worse than that of NeRV, with $W(100) = 8941$, $p < 0.001$.
3.) **Lsund3D**: The performance of Pswarm is significantly better than that of t-SNE, with $W(100) = 4145$, $p < 0.02$, and not significantly worse than that of CCA, with $W(100) = 5444$, $p = 0.14$. However, the performance of Pswarm is significantly worse than that of NeRV, with $W(100) = 7969$, $p < 0.001$.
4.) **Chainlink**: The performance of Pswarm is significantly better than that of NeRV, with $W(100) = 2472$, $p < 0.001$, and worse than that of CCA, with $W(100) = 6270$, $p = 0.001$.
5.) **Tetra**: The performance of Pswarm is significantly better than that of CCA, with $W(100) = 2879$, $p < 0.001$, and not significantly worse than that of ESOM, with $W(100) = 5000$, $p = 0.5$.

10.3 Topographic Maps with Hypsometric Colors

To compare Pswarm as a projection method with SOP and ESOM, the data sets of [Herrmann, 2011, pp. 99-100] were used. After the computation of several trials based only on the visually best[71] scatter plot, topographic maps with hypsometric colors (hypsometric tints) were generated. The Atom, Chainlink, EngyTime, Iris, Swiss Banknotes, and Wine data sets were projected using SOP, ESOM and Pswarm and visualized using the U-matrix or generalized U-matrix approach.

Figure 10.6 shows that only the colored labels corresponding to the prior classification separate the two clusters of EngyTime. The topographic map is inconsistent with the projected points in terms of lattice locations. Moreover, the separation is blurry, and several points are misplaced. Notably, the cardinality of the data set is 4096, and there are only 4096 positions on a 64x64 lattice. However, the visualization presented in Figure 10.6 shows many empty positions. Consequently, there are many positions at which more than one DataBot is located; therefore, the colored labels could be misleading, and the quality measures of [Herrmann, 2011] could be incorrect.

[70] With the highest overlap in $M \pm S$. It is assumed that non-overlapping ranges of DCE values are always statistically significant.
[71] In the sense that the structures defined by the prior classification were preserved.

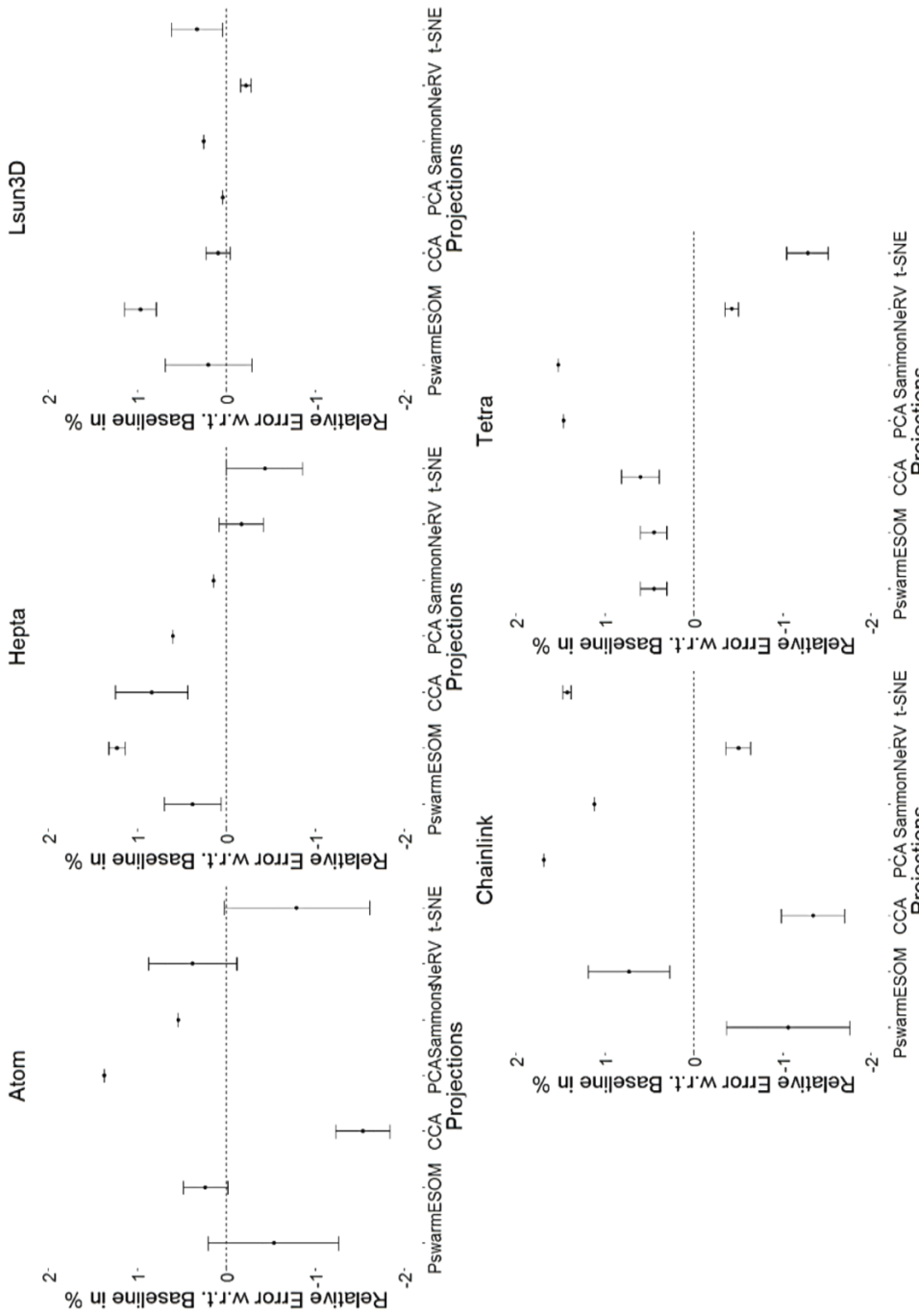

Figure 10.5: Relative DCE values for projections of the Atom, Hepta, Lsun3D, Chainlink and Tetra data sets. The following seven methods are compared: Pswarm ESOM, CCA, PCA, Sammons mapping, NeRV and t-SNE. The most structure-preserving projections have the lowest negative values. No projection method is able to outperform any other projection method on five all data sets.

Table 10.1: Cluster structures in the artificial benchmark sets of the FCPS [Ultsch, 2005a], as defined in chapter 2. The clustering algorithms with the lowest error rate and variance in Figure 10.1 are listed for each data set. These results confirm the assumptions discussed in chapter 3 regarding the cluster structures sought by common clustering algorithms. On the right the projection methods who were unable to find the structure are listed for the three-dimensional data sets. ESOM method is omitted, because it distributes the projected points uniformly. Additional statistical tests can be found in supplement I.

Data Set	**Cluster Structure**	**Clustering Algorithms that Found this Structure with a Small Variance in the Results**	**Projection Methods that did not Found this Structure**
Atom	Connected, direction-based, varying density, non-linear separable	DBS, MoG, SL, Spectral	NeRV, Sammon's mapping and PCA
Chainlink	Connected, direction-based, non-linear separable	DBS, SL, Spectral, (MoG)	t-SNE, Sammon's mapping and PCA
EngyTime	Connected, unidirectional, varying density	All except SL	
Hepta	Compact, spherical, high intercluster distance	DBS, MoG, PAM, SL, Ward	CCA
Lsun3D	Compact, ellipsoidal, outliers	DBS	t-SNE
Target	Connected, direction-based, outliers	DBS, SL, Spectral	
Tetra	Compact, spherical, low intercluster distance	All except SL and Spectral	PCA and Sammons mapping
Two Diamonds	Compact, spherical, borders defined by discontinuity	All except SL	
Wing Nut	Connected, direction-based, linear separable	DBS, SL, Spectral	
Golf Ball	No natural clustering tendency	DBS	

By contrast, in the topographic map of the Pswarm projection shown in Figure 10.7, the clusters are clearly separated by both the positions of the projected points and the high-dimensional distances and densities of the generalized U*-matrix. Here, only one DataBot is allowed per grid position. In comparison to Figure 10.7, the planar ESOM/U*-matrix projection presented in Figure 10.8 does not clearly show the border between the two clusters. As shown in Figure 10.9, when the default settings (toroidal) are used, it is difficult to distinguish between the two clusters. Because the extraction of an island was not possible, a tiled display is shown in Figure 10.9. Likewise, for the Wing Nut data set, the topograpic map of the Pswarm projection shows a clear cluster structure, whereas the toroidal ESOM/U-matrix projection does not (Figure 10.10 and supplement E, Figure E.23) when the P-matrix and U*-matrix visualization is not used.
On the Iris data set, the topographic map of the generalized U*-matrix of the SOP result shows three clusters that are clearly separated by hills, but these clusters do not match the colored labels of the prior classification (supplement C, Figure C.13). By contrast, the Pswarm projection visualized using the generalized U*-matrix approach does show these clusters, one of which is defined by its density (supplement C, Figure C.14). Five points are misplaced. The ESOM/U-matrix method is unable to separate two of the three clusters (supplement E, Figure E.22).

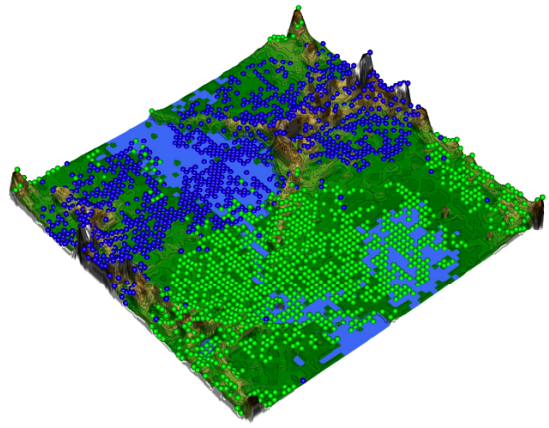

Figure 10.6: Topographic map of the EngyTime data set projected using SOP with the default parameters: The two clusters are mixed and difficult to separate without the colored labels corresponding to the classification. The radius of the P-matrix was automatically chosen to be 1.38. No island could be extracted.

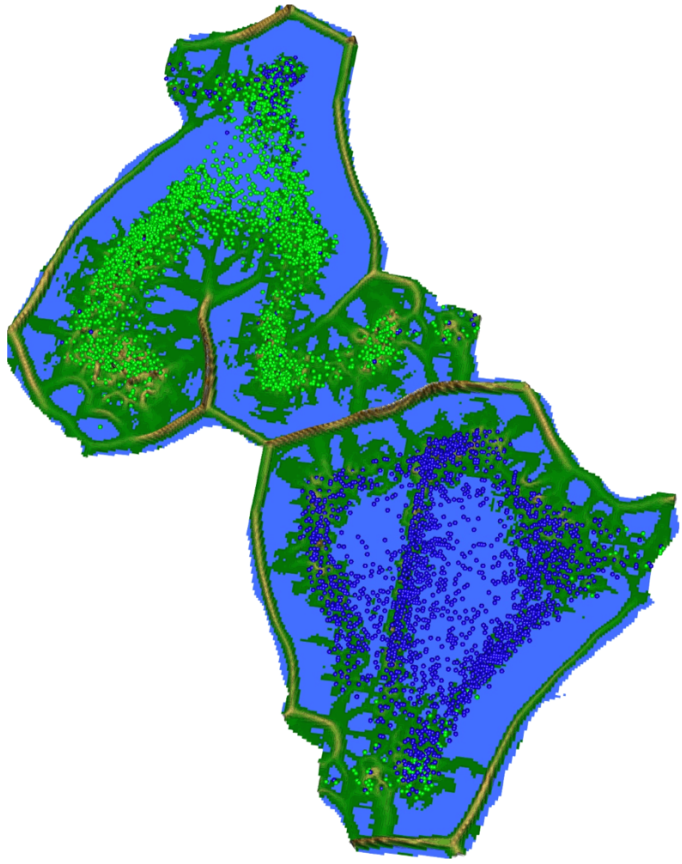

Figure 10.7: Topographic map of the EngyTime data set projected using DBS (196x220) with an automatically chosen lattice size: There are clearly two clusters with an accuracy of the DBS clustering of 95%

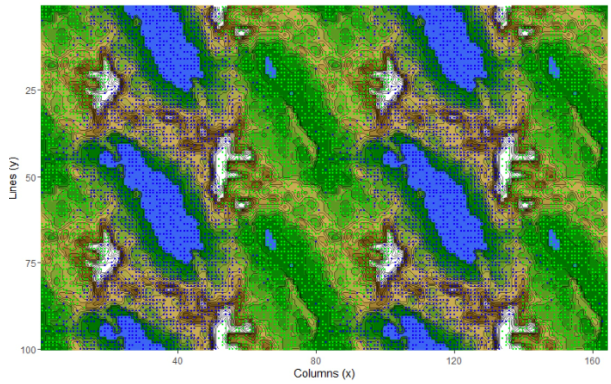

Figure 10.8: U*-matrix visualization of the toroidal ESOM projection of the EngyTime data set: The data set contains 4096 observations, and the lattice contains 4096 neurons. As shown, not every neuron is a best matching unit (BMU); therefore some BMUs include more than one observation, and the colored labels are misleading. The clusters are mixed, and no border between the green and blue BMUs can be found.

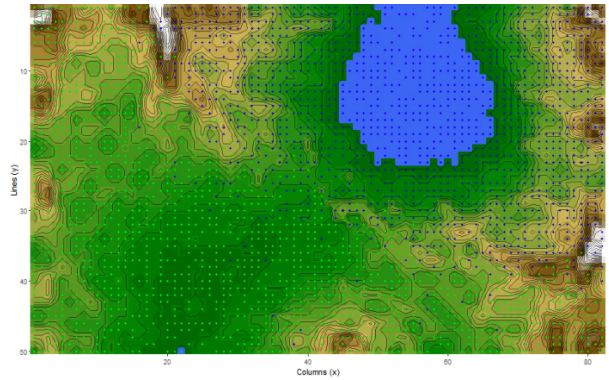

Figure 10.9: U*-matrix visualization of the planar ESOM projection of the EngyTime data set: The data set contains 4096 observations, and the lattice contains 4096 neurons. As shown, not every neuron is a best matching unit (BMU); therefore, some BMUs include more than one observation, and the colored labels are misleading. The clusters are mixed, and a border between the green and blue BMUs is difficult to locate.

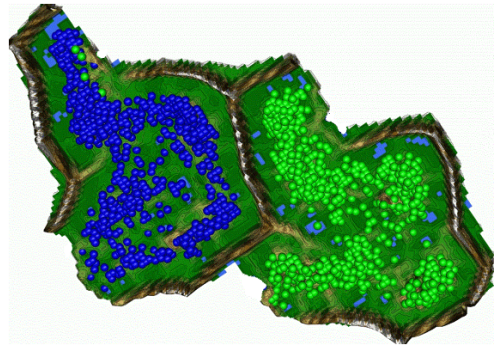

Figure 10.10: Topographic map of the DBS projection of the Wing Nut data set with Generalized Umatrix (64x68). Both clusters are clearly separated, but four points are misplaced.

The topograpic map of the Swiss Banknotes data set as projected using SOP shows three clusters based on high-dimensional distances in the generalized U-matrix, with one misplaced point (supplement C, Figure C.9). Without the topographic map, a scatter plot of the projected points would not lead the reader to the conclusion that the data set consists of separate clusters because the projected points defined by the DataBots are uniformly distributed. By comparison, Pswarm reveals two unambiguously separated clusters with two misplaced points (supplement C, Figure C.10). In the ESOM/U-matrix projection, one best matching unit is misplaced. The cluster of blue best matching unit could be interpreted as two clusters, one small and one large, based on the high hills in between (supplement E, Figure E.21).An interpretation of the uniformly distributed projected points of the Wine data set, as generated via SOP, does not allow the number of clusters to be determined (supplement C, Figure C.11). The generalized U-matrix shows no clear borders between projected points with differently colored labels. Several points are misplaced. By contrast, the topographic map of the Pswarm projection explicitly shows three clusters (supplement C, Figure C.12). — one triangular, one rectangular and one square — but six points are misplaced. In the ESOM/U-matrix projection, the clusters in the Wine data set are difficult to separate without their colored labels (supplement E, Figure E.20). Again, in the SOP result for the Atom data set, the clusters are distinguished only by the borders of the generalized U-matrix and the colored labels corresponding to the prior classification because the points are uniformly distributed (supplement C, Figure C.15). However, the visualization could also be misleading in suggesting that the data set consists of three clusters. The topographic map of the Pswarm projection explicitly shows two clusters (supplement C, Figure C.16).The projections of the Chainlink data set obtained using both SOP and Pswarm are similar (supplement C, Figure C.17) but the Pswarm visualization is smoother in terms of intracluster structure (supplement C, Figure C.18).

11 DBS on Natural Data Sets

Several real-world data sets are used in this chapter to show that Databionic swarm (DBS) is able to find clusters in a variety of cases. The leukemia data set is based on luminance measurements of 7747 different active or non-active genes in 554 human subjects. The World GDP data set is a multivariate time series that consists of monetary values for 190 countries from 1970 to 2010. The Tetragonula data set contains 13 string variables consisting of pairs of alleles for 13 microsatellite loci in bees. In each case, suitable preprocessing and a correctly chosen distance definition make it possible for DBS to cluster and visualize the data such that the known knowledge is reproduced.

11.1 Types of Leukemia

The leukemia data set consists of 7747 variables for 554 subjects (for details, see chapter 3). Of the subjects, 109 were healthy, 15 were diagnosed with acute promyelocytic leukemia (APL), 266 had chronic lymphocytic leukemia (CLL), and 164 had acute myeloid leukemia (AML). The leukemia data set is a high-dimensional data set with natural clusters specified by the illness status and defined by discontinuities (for details, see chapters 3 and 9).

Figure 11.1 shows a visualization of the healthy patients and the patients diagnosed with these three major types of leukemia. The four groups are well separated by mountains, with the subjects represented by points of different colors. Magenta points indicate healthy subjects, whereas points of other colors indicate ill subjects. The automatic clustering of DBS is able to separate the four groups with an accuracy of 99.6%. Two outliers can be seen in Figure 11.1, marked with red arrows. These green and yellow outliers cannot be explained without deanonymization of the patients, which was not feasible for the author. They may be misclassified, but a future publication will address this diagnostic problem[72].

11.2 World Gross Domestic Product (World GDP)

The World GDP data set, published in [Leister, 2016], consists of data on the gross domestic product (GDP) per capita for 160 countries over the past 40 years (see chapter 9 for details). The dynamic time warping (DTW) distances were calculated using the R package dtw [Giorgino, 2009], which computes the optimal alignment between two time series [Giorgino, 2009]. The homogeneity of the cluster structures of DBS is visualized in a silhouette plot in Figure 11.4, the result of the DBS method in Figure 11.2 shows this clear cluster structure and it is confirmed by the heatmap in Figure 11.3.

As the rules deduced through Classification and Regression Tree (CART) analysis show in Figure 11.5, the clusters are defined by a tragic event that occurred in 2001, the crashing of airplanes into the World Trade Center. In its aftermath, “the world economy was experiencing its first synchronized global recession in a quarter-century” [Makinen, 2002, p. 17].

[72] It should be remarked that a data-driven DBS clustering does not reproduce the classification(s) of AML (like FAB subtypes) or CLL of research in this area, e.g. [Bene et al., 1995; Bennett et al., 1985; Vardiman et al., 2009; Haferlach et al., 2010], for CLL see [Rosenwald et al., 2001]. See also p. 30 fn. 19.

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_11

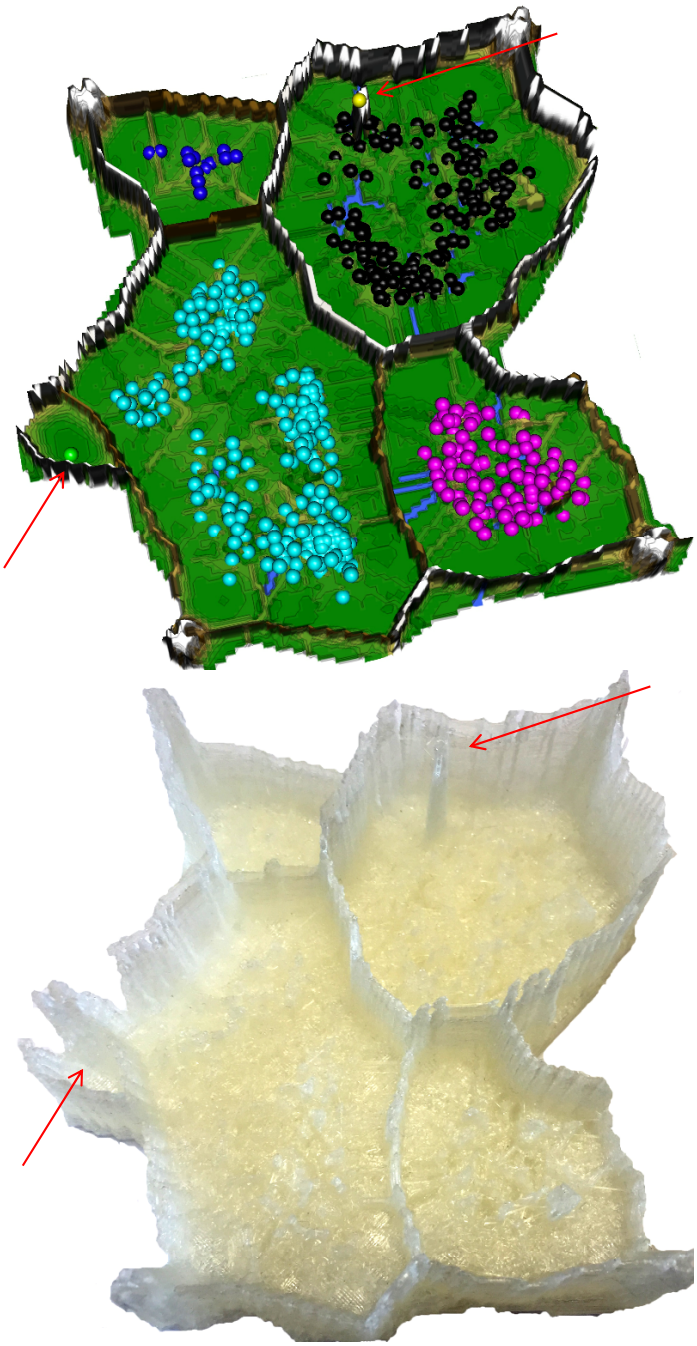

Figure 11.1: Topographic map with DBS clustering results for the leukemia data set, showing six clusters and an accuracy of 99.6% in comparison with the prior classification of four leukemia statuses.
Top: healthy (magenta), AML (cyan), APL (blue), and CLL (black). Two outliers are marked with red arrows: an APL outlier (green) and a CLL outlier (yellow).
Bottom: 3D print (see [Thrun et al., 2016a]), colors are not available yet due to technical limitations.

Therefore, the first cluster consists mostly of African and Asian countries, which were generally unaffected by this event, and the second cluster consists of American and European countries, which were affected. The outlier is Equatorial Guinea, where the first Parliamentary elections since 1968 were held in 1983. Equatorial Guinea shows the smallest variance in its GDP, which is mostly based on oil — this small country, with an area of 28,000 square kilometers, is one of sub-Saharan Africa's largest oil producers.

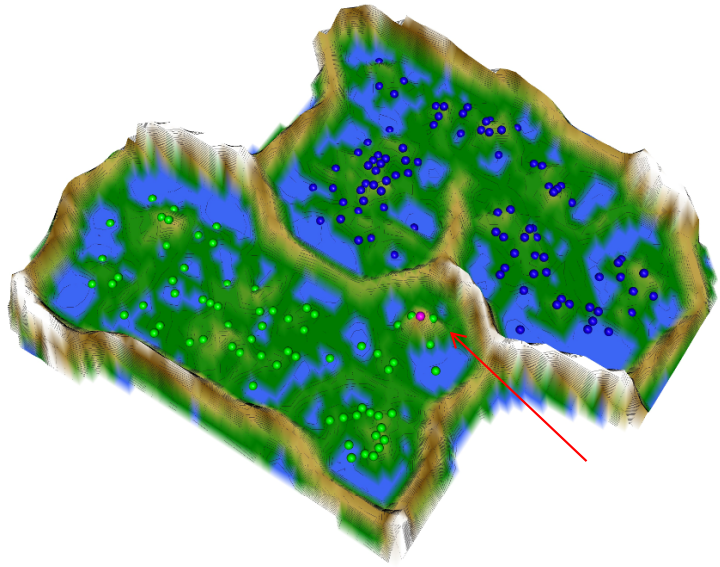

Figure 11.2: Topographic map of the DBS clustering of the World GDP data set shows two distinctive clusters. There is one outlier, colored in magenta and marked with a red arrow.

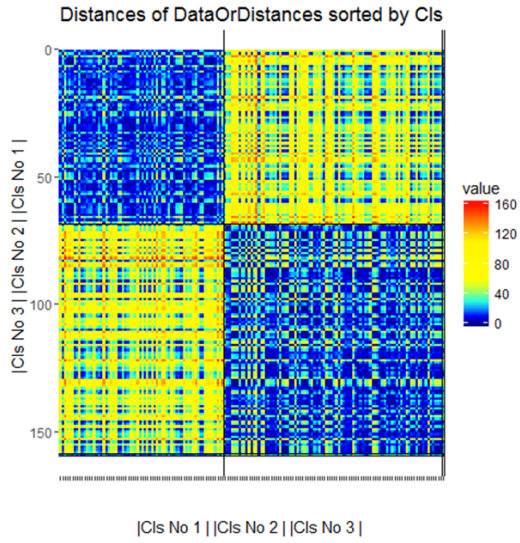

Figure 11.3: Heatmap of the dynamic time warping (DTW) distances for the World GDP data set shows a small variance of intracluster distance.

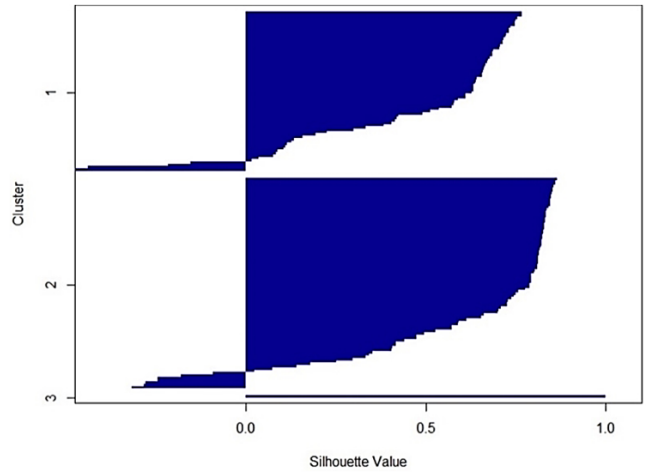

Figure 11.4: Silhouette plot of the DBS clustering results for the World GDP data set indicates that data points (y-axis) above a value of 0.5 (x-axis) have been assigned to an appropriate cluster.

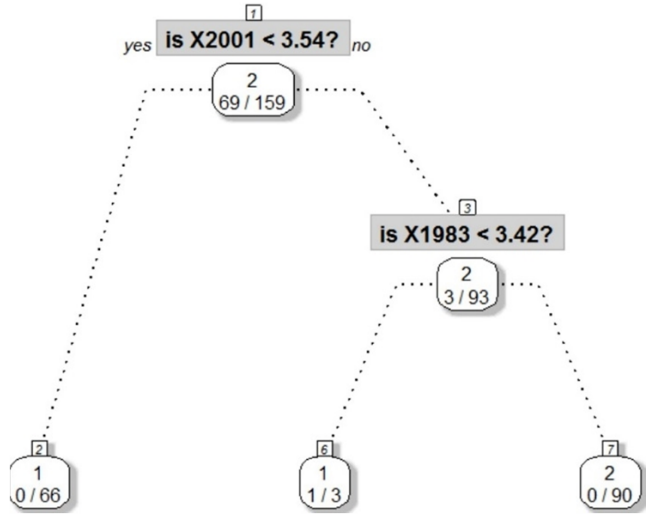

Figure 11.5: Classification and Regression Tree (CART) analysis rules for the clusters. The two main clusters are defined only by an event in 2001.

11.3 Tetragonula Bees

The Tetragonula data set was published in [Franck et al., 2004] and contains the genetic data of 236 Tetragonula bees from Australia and Southeast Asia, expressed using 13 variables (for details, see chapter 9), with a specific distance definition.
The shared allele distance is described in [Hausdorf/Hennig, 2010, p. 493] as follows:

> *"[The distance is] defined as one minus the proportion of alleles shared by 2 individuals averaged over loci. Loci with missing values are not considered in the pairwise distance calculation. In the presence of missing values, this distance measure is not necessarily a metric."*

For the distance calculation, the R package fpc of [Hausdorf/Hennig, 2010] was used with the distance introduced by [Bowcock et al., 1994].
The first DBS visualization implied the existence of 8 clusters and two pairs of outliers. Hence, 100 trials of Pswarm projection and DBS clustering with k=10 clusters were generated, and the best one (i.e., the one with the smallest Delaunay clustering error (DCE)) was chosen (Figure 11.7). The silhouette plot indicates a hyperspherical cluster structure (Figure 11.6) and the heatmap of the distances in Figure 11.9 confirmed the DBS clustering. This application of DBS illustrated the possibility of using multiple swarms by means of parallel computing, for which the term *deep swarming* (see [Ultsch, 2016b]) is introduced in this work in analogy to deep learning [Goodfellow et al., 2016]. Additionally, using the prabclus package, the largest within-cluster gap, the cluster separation, and the average within-cluster dissimilarity of [Hennig, 2014] were calculated to be 0.5, 0.33 and 0.29, respectively. These values are the minima reported in [Hennig, 2014], presented there in Fig. 4. Seven clusters of the average linkage hierarchical clustering with ten clusters ([Hennig, 2014, p. 5]) could be reproduced (see supplement H) with a total accuracy of 93%. Finally, as Figure 11.8 shows, the clusters strongly depend on the geographic origins of the bees:

> *"Longitude (x-axis) and latitude (y-axis) of locations of individuals in decimal format, i.e. one number is latitude (negative values are South), with minutes and seconds converted to fractions. The other number is longitude (negative values are West)" (see [Hennig, 2014] and the prabclus package).*

After the transformation into a two-dimensional plane Figure 11.8 shows that the first eight clusters (96% of data) are consistent with the geography (top) except for the Outliers in Queensland (bottom). The dependency on geography was also illustrated in [Franck et al., 2004, p. 2319].

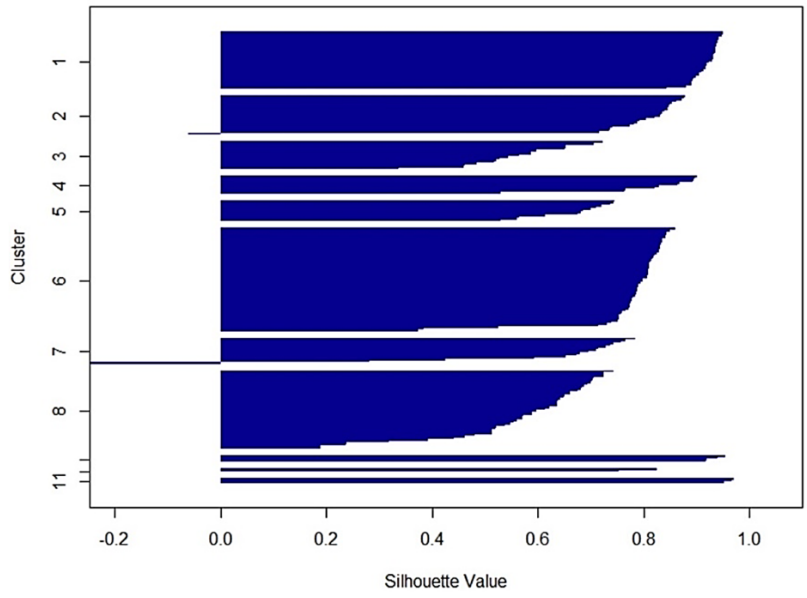

Figure 11.6: Silhouette plot of the Tetragonula data set, showing very homogeneous cluster structures because most of the data points (y-axis) are above a value of 0.5 (x-axis).

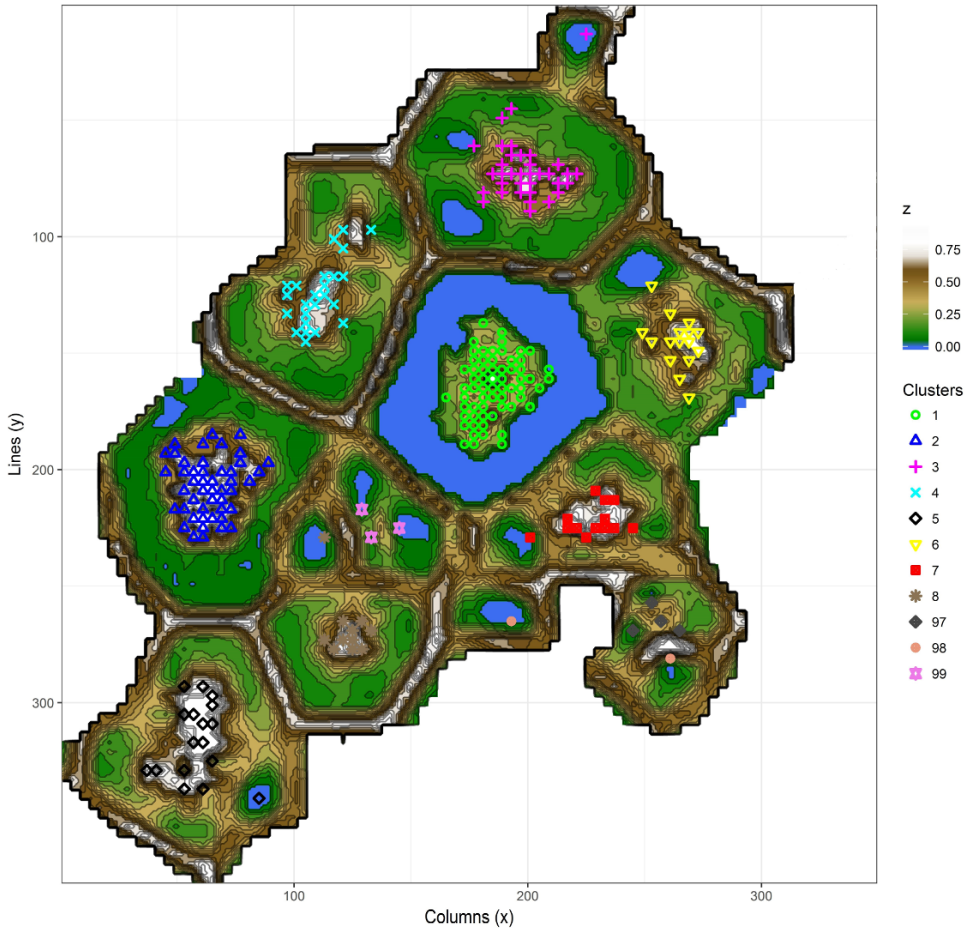

Figure 11.7: Topographic map of the DBS clustering of the Tetragonula data set with the best DCE shows eight clusters and three groups of outliers. The cluster labels are colored as shown on the right, and a similar color code is used in Figure 11.8 below. Clusters are ordered sequentially by the number of samples such that in cluster 1 lies the bee species with the highest occurrence.

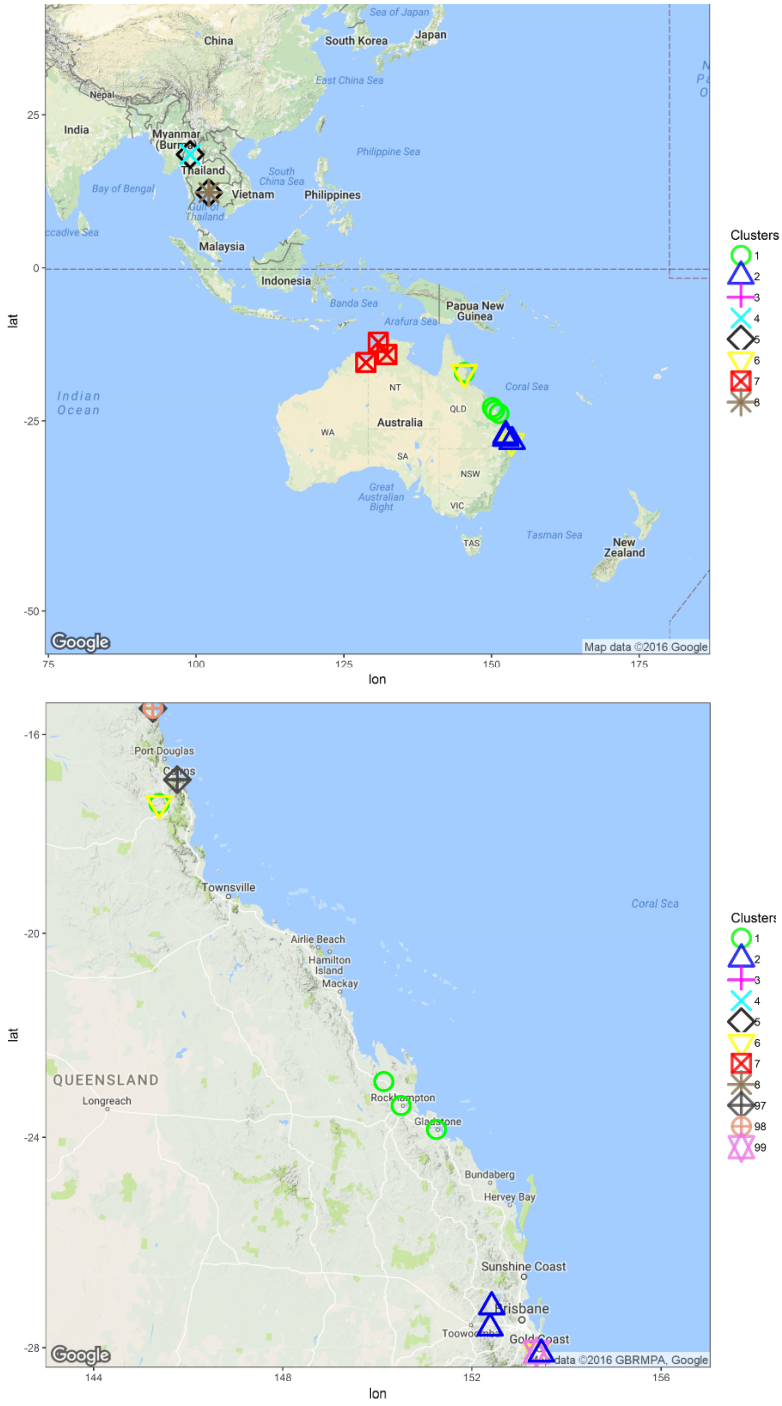

Figure 11.8: Clustering is consistent with the geographic origins: The first eight clusters (96% of data) are consistent with the geography (top) except for the Outliers in Queensland (bottom). Pictures were generated using the ggmap CRAN package.

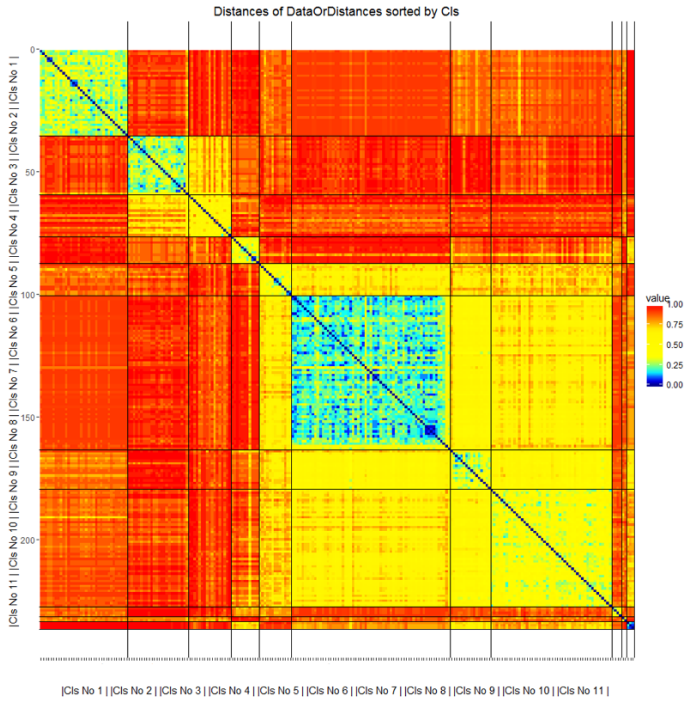

Figure 11.9: Heatmap of the distances for the Tetragonula data set shows large intercluster distances.

12 Knowledge Discovery with DBS

In contrast to chapter 11, in which Databionic swarm (DBS) clustering was applied to recognize more or less obvious knowledge, this chapter shows that DBS is also able to discover new knowledge. A hydrological data set of multivariate time series [Aubert et al., 2016] and a data set consisting of pain genes [Ultsch et al., 2016b] are used for this purpose. In [Aubert et al., 2016], a high-frequency time series analysis was performed, but no prediction could be made. Here, the focus is placed on daily frequency.
The analysis of [Ultsch et al., 2016b] concentrated on chronic pain, and for that reason, it required searching for candidate genes that modulate pain chronification. This chapter, however, focuses on defining the distances between genes and grouping genes by semantic similarity, which can be explained based on overrepresentation analysis (ORA) [Backes et al., 2007].

12.1 Hydrology

> *"Human activities modify the global nitrogen cycle, particularly through farming. These practices have unintended consequences; for example, nitrate lost from terrestrial runoff to streams and estuaries can impact aquatic life" [Aubert et al., 2016].*

A greater understanding of water quality variations can improve the evaluation of the state of water bodies and lead to better recommendations for appropriate and efficient management practices [Cirmo/McDonnell, 1997]**.** Accordingly, the objective here is to predict water quality in the Schwingbach catchment[73] using the currently available variables related to chemical water quality: nitrate and (electrical) conductivity (*N&C*) which is a part of the science of hydrology. Electrical conductivity is a measure that reflects the water quality as a whole; this is because it indicates the variations in the presence of ions other than nitrate in the water body [Aubert, 2015]. Nitrate in water bodies is partially responsible for the phenomenon of eutrophication [Diaz, 2001]. Eutrophication occurs when an excess of nutrients (i.e., nitrate) leads to uncontrollable growth of aquatic plant life, followed by a depletion of the dissolved oxygen [Diaz, 2001; Howarth et al., 1996]. For this reason, the nitrate concentration is one of the parameters used to evaluate water quality.

> *"The available dataset contained in total 32,196 data points for each of the 14 variables (in total, 4% missing data). For technical reasons, no nitrate data were available during winter, so the actual time span of nitrate monitoring was 05 March 2013 12:45 to 24 September 2013 12:30 and 27 April 2014 00:00 to 23 October 13:15. Data were analyzed as a whole, without differentiating between the hydrological years"* [Aubert et al., 2016].

Conductivity, in particular, will be explained using another set of variables, which are indicators of hydrological and biological conditions. In contrast to the temporal high-frequency analysis (with 15-minute intervals) of [Aubert et al., 2016], here, the daily courses for each variable were calculated as the sums of all daily measurements, resulting in a low-frequency analysis. The missing values were imputed using the seven-nearest-neighbors approach. All variables were linearly decorrelated, and the logarithms of the variables q13 and q18 were calculated. Subsequently, all variables, with the exception of rain, were normalized to values between zero

[73] A catchment is a dynamic system, and current observations depend on previous hydrological states [Aubert et al., 2016].

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_12

and one through robust normalization. The outliers in the rain variable were detected via ABC analysis [Ultsch/Lötsch, 2015]: in the ABC analysis, rain was normalized with respect to the minimum value in group A and then all points in group A were set to a value of 1.1 for rain, and. After feature selection the data set had in 12 variables over 343 days.

The preprocessed daily courses are shown in Figure 12.1. The preprocessing resulted in Euclidean distances with a multimodal distribution (Figure 12.2). The first mode represents the intracluster distances, and the second mode represents the intercluster distances (see also chapter 3, Figure 3.1).

DBS was used for visualization and clustering. The outliers were marked interactively, resulting in five classes (Figure 12.4). The clusters have small intracluster distances and high intercluster distances, as visualized using DBS (Figure 12.4) and confirmed by the heatmap (Figure 12.4). The silhouette plot shows that all clusters can be well modeled as hyperspheres (Figure 12.3).

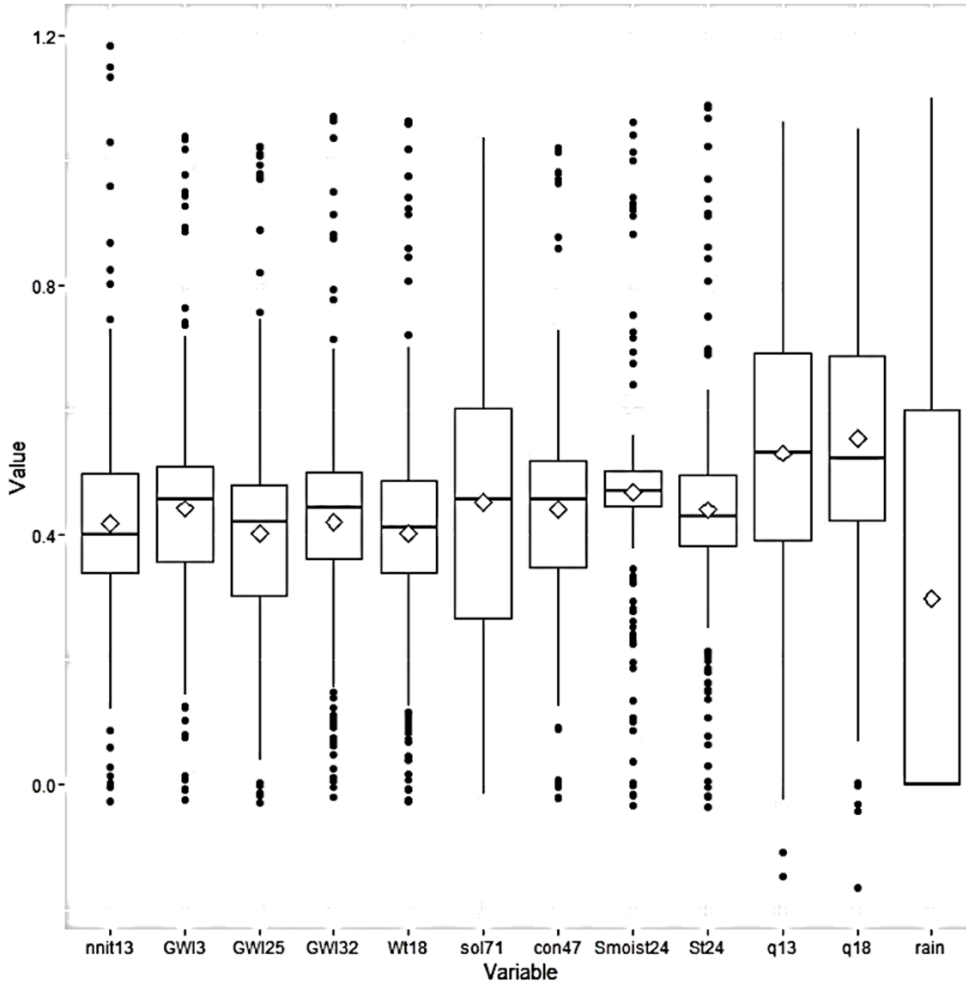

Figure 12.1: Variances of variables after preprocessing and feature extraction visualized using boxplots after the preprocessing of the hydrology data set.

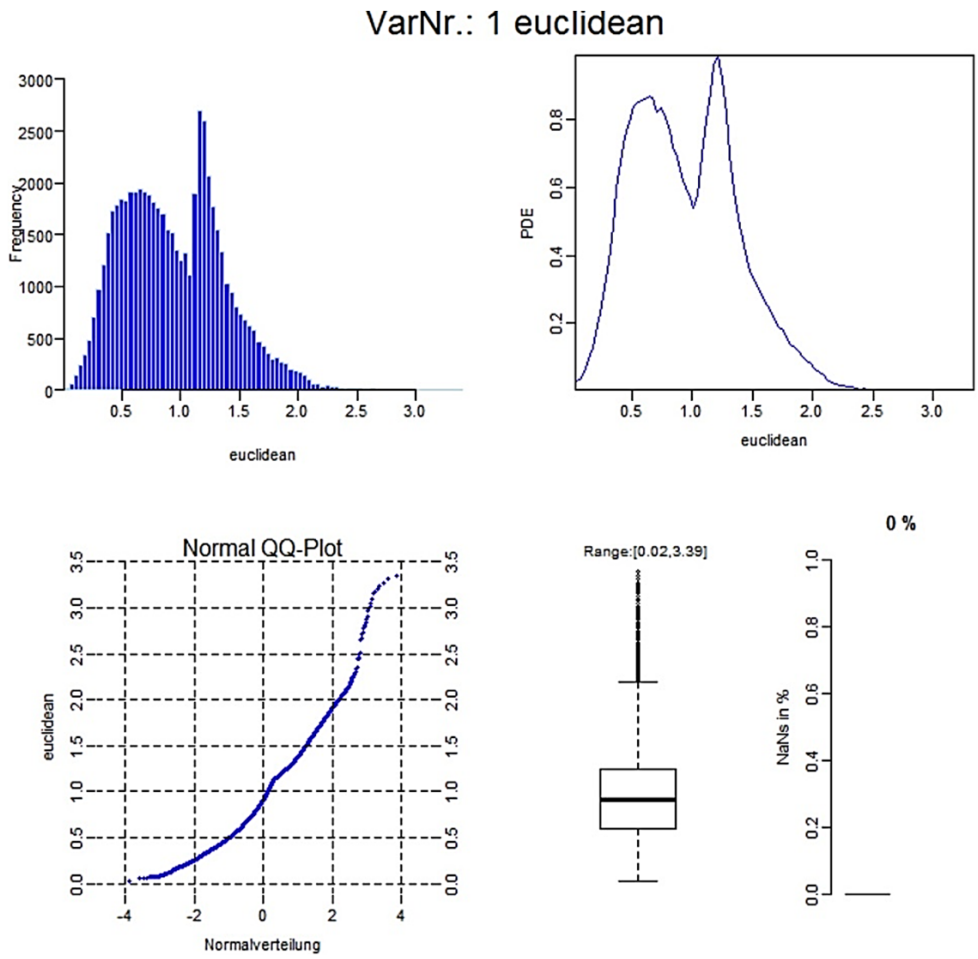

Figure 12.2: Distribution analysis of the distances. The first mode represents the intracluster distances, and the second mode represents the intercluster distances (for further explanation see chapter 3, Figure 3.1).

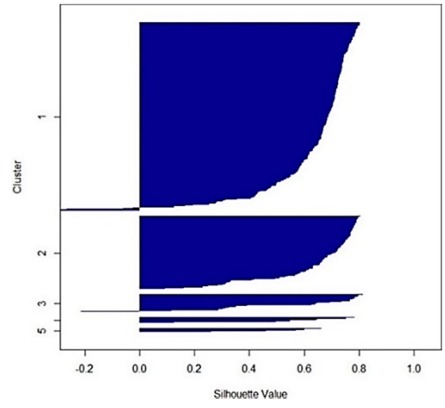

Figure 12.3: Silhouette plot of the DBS clustering set indicates that data points (y-axis) above a value of 0.5 (x-axis) have been assigned to an appropriate cluster.

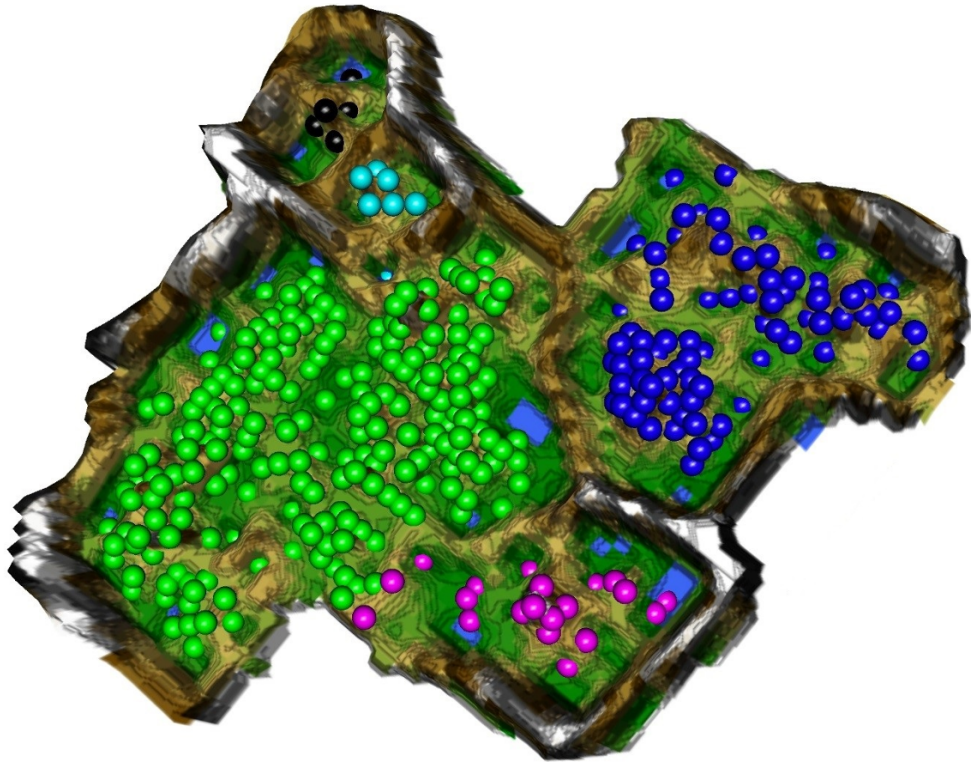

Figure 12.4: Five clusters are shown in the topographic map of DBS of the Hydrology data set. For 3D print see supplement G, Figure G.24.

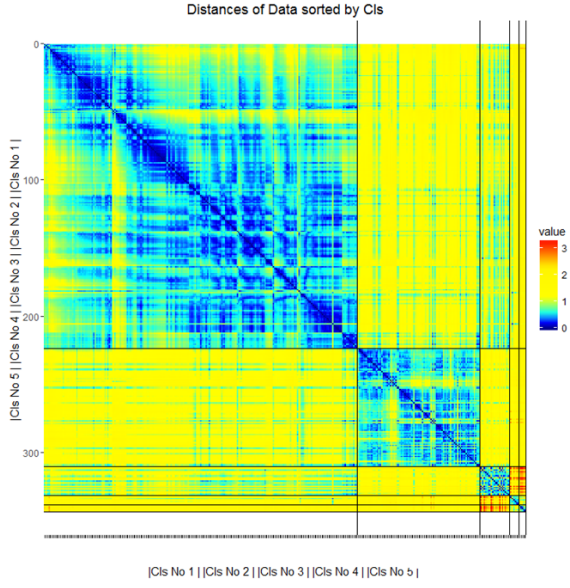

Figure 12.5: The five clusters have clearly distinctive distances, as shown by the heatmap; there are small distances within each cluster and large distances between the clusters.

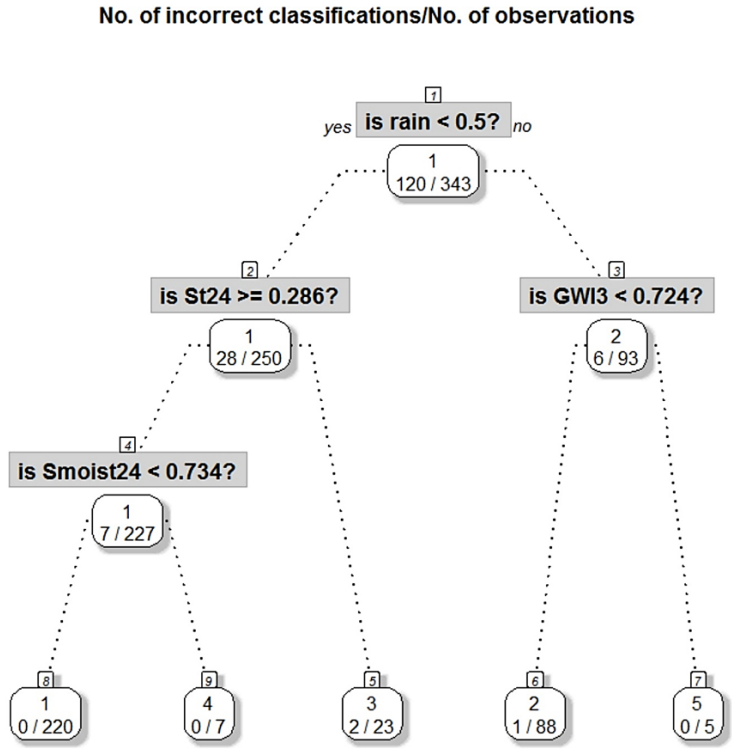

Figure 12.6: Classification and Regression Tree (CART) analysis rules for the hydrology data set with the five clusters identified by DBS. Applying the rules to the clustering combined with the data set results in three misclassified points (0.9%). Abbreviations: rainfall intensity (rain), soil temperature (St24), soil moisture (Smoist24), groundwater level at point 3 (GWl3). All values are expressed as percentages.

12.1.1 Knowledge Acquisition and Prediction in the Hydrology Data Set

Here, the rules extracted from the Classification and Regression Tree (CART) decision tree, as shown in Figure 12.6, were applied to the clustering. In comparison to the DBS clustering, the application of the CART rules to the data set results in the misclassification of three data points (0.9%). Based on this finding, it can be said that the rules precisely classify the data set (Figure 12.6). The generated rules are listed in Table 12.1.

Table 12.1: The CART rules based on Figure 12.6, in which the clusters of Figure 12.4 are used. Abbreviations: rainfall intensity (rain), soil temperature (St24), soil moisture (Smoist24), groundwater level at point 3 (GWl3). All values are expressed as percentages.

Rule No.	DBS Cluster No.	No. of Days	Rule
R1	1	223	if rain < 0.5 and St24 ≥ 0.29 and Smoist24 < 0.73
R2	4	7	if rain < 0.5 and St24≥ 0.29 and Smoist24 ≥ 0.73
R3	3	21	if rain < 0.5 and St24 < 0.29
R4	2	87	if rain ≥ 0.5 and GWl3 < 0.72
R5	5	5	if rain ≥ 0.5 and GWl3 ≥ 0.72

The N&C measurements can be described by two variables related to biological processes, namely, soil temperature and soil moisture, and two variables related to hydrological processes, namely, rainfall intensity and groundwater level at point 3, which represents downslope conditions. Temperature influences the activities of living organisms, such as soil microbial organisms [Zak et al., 1999]. Soil moisture determines microbial activities, such as long-term inactivity in dried soil followed by wetting [Borken/Matzner, 2009]. The groundwater level (or head, in m) is the main factor driving discharge in a catchment [Orlowski et al., 2014]. Rainfall intensity triggers discharge and affects soil moisture as well as leaching of nutrients [Orlowski et al., 2014].

A thorough examination of the CART results based on the five distinguishing rules R (Tab. 1) yields the following classes C:

C1/R1: Low rain, higher soil temperature, lower soil moisture => *DryDays WetHotGround*
C2/R4: High rain, lower downslope groundwater level => *Rain Shower*
C3/R3: Low rain, low soil temperature => *DryDays Cold Ground*
C4/R2: Low rain, higher soil temperature, high soil moisture => *DryDays DryHotGround*
C5/R5: High rain, high downslope groundwater level => *Rainy Days*

With regard to N&C, these classes can be distinguished as follows: the first two classes (green and blue) are responsible for normal N&C, the third class (magenta) is associated with low N&C, and the fourth and fifth classes (teal and black) are responsible for high N&C (Figure 12.7).

After a rain shower or on dry days when the ground is wet and hot, the N&C concentrations are normal. The N&C concentrations are high (above 50%) on rainy days, when the downslope groundwater level is above 72%. The N&C concentration is low (<25%) on dry days (below 50% rain) when the ground is cold (below 29% of the maximum ground temperature). These definitions enable future predictions of daily N&C concentrations.

It is assumed here that the structures associated with the 5 clusters described by these classes are defined by discontinuities. Consequently, the clusters should contain samples of different natures and based on different processes. Given this assumption, it is valid to statistically test whether the N&C distributions significantly differ between clusters. The Kolmogorov–Smirnov test (KS test) is a nonparametric two-sample test of the null hypothesis that two variables are drawn from the same continuous distribution [Conover, 1971, pp. 309-314], and it is implemented in the R language [R Development Core Team, 2008].

The statistical results are shown in supplement F, Tab. 1 and 2. All N&C distributions significantly differ between clusters, with the exception of cluster 4 compared with 5, for both variables.

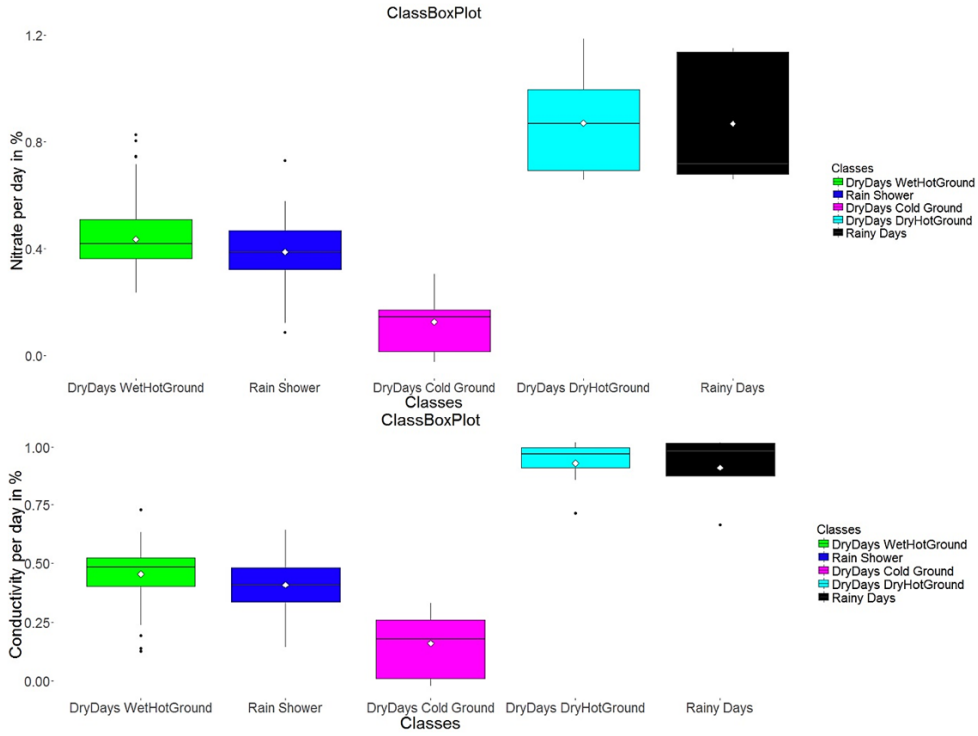

Figure 12.7: Boxplots of the five classes with regard to nitrate N (top) and conductivity C (bottom). All values are expressed as percentages.

12.2 Pain Genes

In [Ultsch et al., 2016b], a set of genes with relevance to pain[74] was obtained from four sources, and the search of several databases and studies (e.g., the Pain Genes Database, the PubMed database) was described in detail. This search yielded a set of n = 535 genes, subsequently referred to as *pain genes* in [Ultsch et al., 2016b].

After accessing the Gene Ontology (GO) database in this work, 528 of the pain genes were found to be annotated, and the remaining seven genes were disregarded in the subsequent analysis (feature selection). Various types of annotation (evidence codes) are possible. When the inverse document frequency *idf* is used [Sparck Jones, 1972], the distances between these genes are defined as follows (as discussed in [Ultsch, 2014b]):

Let the documents be represented by GO terms T, and let the terms used to calculate *idf* be represented by the genes G, which are coded with numbers defined by the National Center for Biotechnology Information (NCBI) [NCBI, 2013]; the term frequency *tf* is then the frequency of occurrence of a gene in a given document divided by the maximal occurrence of the gene in any document:

[74] "An unpleasant sensory and emotional experience associated with actual or potential tissue damage, or described in terms of such damage" [Merskey/Bogduk, 1994].

$$tf(G,T) = \frac{f(G,T)}{\max(f)} \qquad (1)$$

If only manually curated evidence codes are used for annotation, then $tf(G,T) = 1$.
Let N be the number of GO terms to which the pain genes are annotated, and let n_i be the number of GO terms to which a pain gene with a given NCBI number is annotated; then, the inverse document frequency is defined as

$$idf_i = log\left(1 + \frac{N}{n_i}\right) \qquad (2)$$

and the term frequency–inverse document frequency is defined as

$$tfidf = tf(G,T) * idf_i = 1 * idf_i \qquad (3)$$

A gene that is annotated to only some GO terms is more meaningful than one that is annotated to almost every or only a few GO terms . Hence, the inverse document frequency reduces the weights of genes that occur very frequently among the GO terms and increases the weight of genes that occur rarely. The distance D between two genes l and j is defined as the absolute distance in terms of *idf*:

$$D(l,j) = abs(idf_l - idf_j) \qquad (4)$$

This distance was used to generate the DBS visualization shown in Figure 12.9, and clustering was automatically performed after the identification of 8 clusters in the visualization. The clusters are verified by the heatmap presented in Figure 12.10 and the Silhouette plot in Figure 12.8.

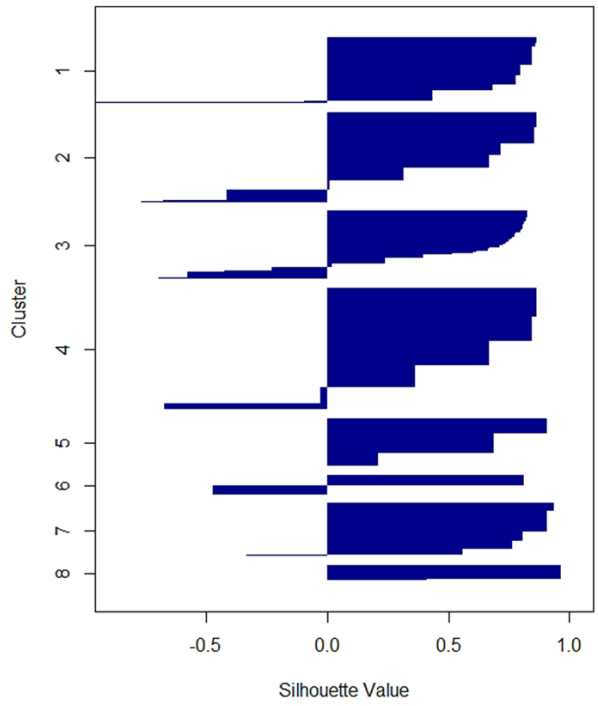

Figure 12.8: Silhouette plot of the DBS clustering of pain genes. Most of clusters of pain genes can be modeled as hyperspheres. However, cluster 6 has a different high-dimensional structure.

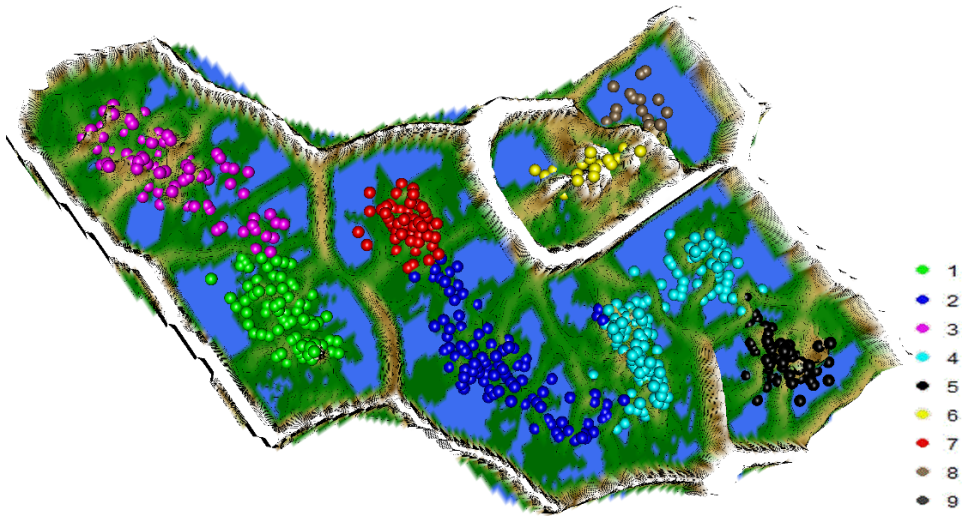

Figure 12.9: Topographic map of DBS clustering of 528 pain genes. Clusters 1 and 3 and clusters 2 and 4 are very similar to each other. Cluster 6, labeled in yellow, consists of outliers. The counts per cluster, from 1 to 8, are 72, 99, 75, 133, 53, 21, 58, and 17. For 3D print see supplement G, Figure G.25.

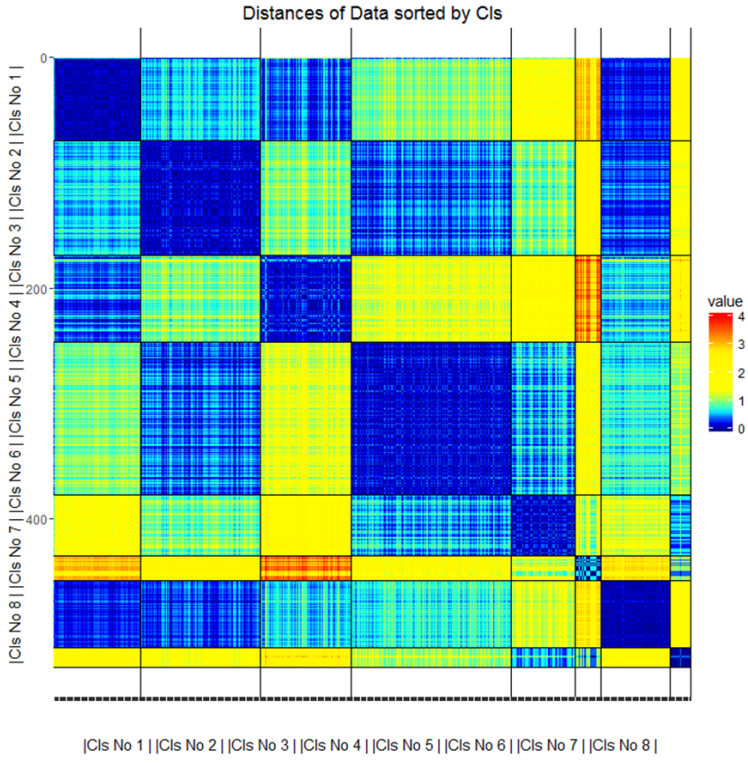

Figure 12.10: Heatmap of the distances with regard to the 8 identified clusters of pain genes, which verifies that the clustering is sound. Clusters 1 and 3 and clusters 2 and 4 are very similar to each other. Cluster 6 is clearly defined by outliers.

12.2.1 Prior Knowledge

The pain genes were analyzed by means of ORA, revealing several important functions, as listed below. If the distance definition and DBS clustering were applied correctly to the pain genes data set, it should be possible to rediscover structures that are already known from two main publications on this topic. [Lötsch et al., 2013] defined twelve functions of pain for 460 pain genes (Figure 12.11) [Lötsch et al., 2013]:

1.) regulation of localization
2.) behavior
3.) response to wounding
4.) response to organic substance
5.) cellular ion homeostasis
6.) ion transport
7.) synaptic transmission
8.) G protein-coupled receptor protein signaling pathway
9.) intracellular signal transduction
10.) positive regulation of biological process
11.) regulation of system process
12.) anatomical structure development

Additionally, in 2016, twelve chronification functions of 535 pain genes were identified [Ultsch et al., 2016b]:

1.) single-organism cellular process
2.) biological regulation
3.) cell communication
4.) cellular response to stimulus
5.) localization
6.) response to stress
7.) phosphorus metabolic process
8.) nervous system development
9.) cell death
10.) single-organism behavior
11.) cellular ion homeostasis
12.) rhythmic process

With the aim of reproducing the knowledge listed above, for every cluster in Figure 12.9, ORA was performed using the R package ORA [Lippmann et al., 2016]. The resulting p-values were filtered via ABC analysis, and thereafter, only group A was considered for interpretation (see chapter 9 for further details).

12.2.2 Knowledge Acquisition in Clusters of Pain Genes

DBS identified eight clusters[75] of genes (Figure 12.9). For each cluster, an ORA was performed. In contrast to the standard approach, in which the Bonferroni correction [Perneger, 1998] is

[75] After inspection of the functional areas in the eight ORA results, the eight clusters could be reduced to six (for details, see Tab. 2

often used, here, the p-values of the GO terms in the ORA results were filtered via ABC analysis [Ultsch/Lötsch, 2015]. The Bonferroni correction reduces the alpha error of significance, but it may cause valid results to be disregarded because the beta error simultaneously increases (for extensive discussions, see [Button et al., 2013; Nuzzo, 2014; Perneger, 1998]. Here, it is argued that in the special case of ORA, the p-values also represent the effect strength. Therefore, the adjustments to the significance threshold made by the Bonferroni correction are unnecessary. In contrast to the standard approach, ABC analysis was used to identify the most important GO terms as those assigned to group A, which had the highest effect strength. After the reduction of the directed acyclic graph (DAG) using this approach, the functional areas identified in [Lötsch et al., 2013] and [Ultsch et al., 2016b] were found to be associated with three of the classes (Table 12.2).

Considering the prior knowledge regarding pain functions and pain chronification, the following clusters could be combined: cluster 1 and cluster 3 were combined to class C1*, and cluster 2 and cluster 4 were combined into class C2*, because they showed similar functions and were separated only by low borders in the topographic map with hypsometric tints (Figure 12.9). Hence, it was possible to identify five classes with different semantic characterizations, plus one class of outliers (Tab. 2). Class C1* predominantly describes the pain functions of cells and reproduces knowledge presented in section 11.2.1. The main class (C2*) describes the molecular transport and signaling of pain, also reproducing prior knowledge about the pain genes. class C5 represents the downregulation of metabolic processes and the upregulation of the creatine metabolic process, which is a new discovery enabled by the DBS clustering. Class C6 describes outliers that are not relevant to the ORA-based DAG — these outliers are surrounded by very large hills in Figure 12.9. Class C7 characterizes the response and regulation systems as well as the upregulation of the phosphorus metabolic process, effectively reproducing the results of [Lötsch et al., 2013] and [Ultsch et al., 2016b]. The final class, C8, could represent hematopoietic stem cell differentiation. In summary, these clusters reproduce the previously identified functions of pain genes as described in section 11.2.1. In addition, new insights can also be found from class C5 and perhaps class C8.

Figure 12.11: The biological process of pain with the twelve functions of pain genes [Lötsch et al., 2013].

Table 12.2: Semantic characterization of the eight clusters of pain genes and the connections to prior knowledge. Downregulation is indicated as underlined, and new functional areas [Ultsch/Lötsch, 2014] are indicated in italics. The following clusters in Figure 12.9 were combined with the aid of prior knowledge: C1 and C3 were combined into C1*, and C2 and C4 were combined into C2*.

ORA Parameters	**Class.**	**No. of Genes**	**Semantic Meaning as Defined by GO Terms in ORA**	**Semantic Characterization**
RAW and Bonferroni, minimum number of genes=10	C1*	147	single-organism cellular process cell communication cellular response to stimulus localization cell death cellular ion homeostasis nervous system development single-organism behavior rhythmic process intracellular signal transduction anatomical structure development cellular ion homeostasis	Pain functions of cells
RAW and Bonferroni, minimum number of genes=10	C2*	232	synaptic transmission ion transport G protein-coupled receptor signaling pathway *transmembrane transport*	Molecular transport and signaling
RAW	C5	53	*creatine metabolic process* *metabolic process*	Downregulation of metabolic processes and upregulation of the creatine metabolic process
RAW	C6	21	None	Outliers
RAW and Bonferroni, minimum number of genes=2	C7	58	response to stress phosphorus metabolic process behavior positive regulation of biological process response to organic substance response to wounding regulation of localization regulation of system process	Response and regulation systems as well as upregulation of the phosphorus metabolic process
RAW	C8	17	*hematopoietic stem cell differentiation*	Hematopoietic stem cell differentiation

13 Discussion

This work examined and analyzed patterns in high-dimensional data characterized by discontinuity. Such distance- or density-based patterns are either compact or connected structures. If the structures are compact, inter- versus intracluster distances are relevant. If they are connected, then density relations and neighborhoods play an important role. Here, it was demonstrated that the neighborhood of a point can always be defined based on graph theory. If the neighborhoods are defined based only on distance, then the structure is compact and a Euclidean graph can be used. If the structure is connected, then two subtypes can be deduced from graph theory: direction-based and unidirectional neighborhoods.

In the context of cluster analysis, structures induced by discontinuities lead to natural clusters, as elaborated in chapter 3. The definition of discontinuity in high-dimensional data, presented in chapter 2, enables the generalization of spatial separation, which was described by [Handl et al.] as a third category of clustering criteria [Handl et al., 2005, p. 3202]. Here, in contrast to [Handl et al., 2005], it is argued that there is no distinction between connected and spatially separated structures or between compact and spatially separated structures[76]. Instead, the third category (spatial separation) can be generalized as the prerequisite for natural clusters defined by either compact or connected structures. It was discussed in chapter 3 that, through the application of basic principles founded on graph theory, clustering algorithms usually search for clusters with a predefined structure. However, it is not always clear which structures are sought because the objective functions that are optimized can be mathematically very difficult to understand. An extensive evaluation of the objective functions found in the literature supports this argument and implies two subtypes of structures sought by common clustering algorithms, called direction-based and unidirectional structures. The assumptions put forward in chapter 3 (Figure 3.5) were verified in chapter 10 (Table 10.1) using data sets from the Fundamental Clustering Problems Suite (FCPS). A question arises regarding how one can choose a clustering algorithm that assumes the correct type of cluster structure for a high-dimensional data set without prior knowledge. Here, it is suggested that dimensionality reduction methods for generating (two-dimensional) projections may help solve this problem.

This work has demonstrated that the objective functions used in clustering and projection methods and the quality measures (QMs) used to evaluate them are based on the fundamental distinction between connected and compact structures. The conclusion is that when the task is to achieve a structure-preserving visualization or clustering, the optimization of an objective function could yield misleading results if the underlying structures of the high-dimensional data of interest are unknown. Hence, a completely different approach is required, which, in chapter 7, motivates an extensive review of the application of artificial intelligence in data science. In chapter 7, two interesting concepts are addressed, called self-organization and swarm intelligence. Through self-organization, the irreducible structures of high-dimensional data can emerge, in a process defined as emergence in chapter 7. If properly applied using a swarm of intelligent agents, the approach presented in this work can outperform the optimization of an objective function for the tasks of clustering and dimensionality reduction.

[76] In [Handl et al.], the three categories of clustering criteria were called connectedness, compactness and spatial separation [Handl et al., 2005, p. 3202].

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_13

The Databionic Swarm (DBS) method

> *"[A clustering approach] must be adaptive or exhibit 'plasticity,' possibly allowing for the creation of new clusters, if the data warrants it. On the other hand, if the cluster structures are unstable [...], then it is difficult to ascribe much significance to any particular clustering. This general problem has been called 'the stability/plasticity dilemma' " [Duda et al., 2001, p. 559].*

The work presented herein introduces a clustering algorithm based on a swarm-based projection method combined with a human-understandable visualization technique. In terms of stability and plasticity (chapter 10, Figure 9.1), the Databionic swarm (DBS) framework outperforms common algorithms in clustering tasks on the FCPS.

> *"One source of this dilemma is that with clustering based on a global criterion, every sample can have an influence on the location of a cluster center, regardless of how remote it might be" [Duda et al., 2001, p. 559].*

In contrast to standard approaches, swarm techniques are known for their properties of flexibility and robustness [Bonabeau/Meyer, 2001; Şahin, 2004]. As a swarm technique, DBS clustering is robust with respect to outliers (see chapter 10).
DBS is a flexible and robust clustering framework that consists of three independent modules. The first module is the parameter-free projection method Pswarm, which exploits the concepts of self-organization and emergence, game theory, swarm intelligence and symmetry considerations. The second module is a parameter-free high-dimensional data visualization technique, which generates projected points on a topographic map with hypsometric colors, called the generalized U-matrix. The third module is a clustering method with no sensitive parameters. The clustering can be verified by the visualization and vice versa. The term DBS refers to the method as a whole. DBS enables even a non-professional in the field of data mining to apply its algorithms for visualization and/or clustering to data sets with completely different structures drawn from diverse research fields, simply by downloading the corresponding R package [Thrun, 2017].
Each module of DBS was compared with various competing algorithms, and in the majority of cases, the modules outperformed those algorithms. However, the author of this work concurs with [Coretto/Hennig, 2016] that despite one's best intentions and efforts to conduct fair comparisons of various methods of visualization, projection and clustering, "ultimately it would be good to have comparisons of methods run by researchers who did not have their hand in the design of any of the methods"; this is because "(simulation) studies can always be designed that make any method 'win.' " The author also agrees with [Coretto/Hennig, 2016] that "readers need to make up their own mind about to what extent our study covered situations that are important to them."
With these considerations in mind, DBS was particularly designed to be flexible and to allow the modules to be interchangeable. An expert in the field of data mining may prefer a method with a clear optimization strategy or may not require the entire DBS framework for his/her application. The interchangeability of the modules is useful in such a case. For example, it is possible to use the visualization and clustering module with NeRV instead of Pswarm. Alternatively, a user could cluster a data set using his/her preferred clustering algorithm and then verify the clusters visually using Pswarm and the generalized U-matrix. As another example, a user could use Pswarm and its clustering algorithm with no visualization, by setting the number of clusters with the aid of the dendrogram of the swarm-defined distances. In summary, the

projection based clustering framework proposed here is a user-friendly platform for the visualization of high-dimensional structures and/or for clustering with no sensitive parameters.[77]

Clustering with DBS

> *"[T]he majority of clustering algorithms […] impose a clustering structure on the data set X, even though X may not possess such a structure" [Theodoridis/Koutroumbas, 2009, p. 863].*

Additionally, they may return meaningless results in the absence of natural clusters [Cormack, 1971, pp. 345-346; Handl et al., 2005, p. 3203; Jain/Dubes, 1988, p. 75]. The results presented in this work illustrate that the DBS algorithm does not suffer from these two disadvantages.

The DBS algorithm makes it possible to apply the abstract U-matrix (AU-matrix) [Lötsch/Ultsch, 2014] to a Pswarm projection instead of an emergent self-organizing map (ESOM) projection. The new clustering approach of DBS is defined by using the shortest-path distances [Dijkstra, 1959] of the AU-matrix and a hierarchical approach to clustering. In contrast to swarm-organized projection (SOP) and ESOM, this approach does not require any parameters except the number of clusters and a two-option parameter that specifies the cluster structure as being either compact or connected (see chapter 3 for details). "One of the most difficult decisions to make is the number of clusters" [Everitt et al., 2001, p. 179]. In DBS, the number of clusters and the cluster structure can be easily estimated from a careful examination of the topographic map (by counting the valleys) and with the help of a dendrogram. If the number of clusters and the cluster structure are chosen properly, then the clusters in the topographic map will be well separated by mountains.

It is argued here that DBS clustering should be semi-interactive and requires user supervision to achieve the best possible results. Nevertheless, the results of automatic DBS clustering with no user intervention were also compared with the results of the common clustering algorithms k-means [MacQueen, 1967], partitioning around medoids (PAM) [L. Kaufman/Rousseeuw, 1990], single linkage (SL) [Florek et al., 1951] and spectral clustering [Ng et al., 2002] as well as two state-of-the-art clustering algorithms: the mixture of Gaussians (MoG) method [Fraley/Raftery, 2002] and the Ward algorithm [Ward Jr, 1963]. "Several of the comparative studies […] conclude that Ward's method […] outperforms other hierarchical clustering methods" [Jain/Dubes, 1988, p. 81]. MoG clustering, which is also known as model-based clustering, serves as the reference technique [Bouveyron/Brunet-Saumard, 2014]. Clustering algorithms such as DBscan [Ester et al., 1996] or the ESOM/U-matrix approach [Ultsch et al., 2016a] require additional sensitive and continuous parameters and were omitted from the comparison for that reason. Every clustering algorithm was applied using the default parameter settings and the correct number of clusters. Calculations were performed for 100 trials on the FCPS data sets [Ultsch, 2005c].

The main result achieved in the work presented herein concerns the error rates of the clustering algorithms tested in these trials. As already stated throughout this work, clustering algorithms often predefine the structure of the clusters they seek; e.g., for PAM and k-means, the shape is round, and thus, the structure is compact. Therefore, these algorithms failed on the Chainlink and Atom data sets. In addition, the k-means and spectral clustering algorithms showed large

[77] After this work it was also made available in [Thrun et al., 2017, Thrun/Ultsch, 2017a].

variances in their results on the Hepta and Target data sets. It is known that the k-means algorithm sometimes strongly depends on the order of objects in a data set [L. R. Kaufman/Rousseeuw, 2005, p. 114], which may be the cause of the large variance in the results. This variance was shown through several examples for the spectral clustering algorithm, in which case the results were strongly trial-dependent, even when the parameter settings remain unchanged. The MoG method yielded results of comparably good quality to those of DBS, but it still failed in the case of the Lsun3D data set (in the sense that it showed a large variance) and in the case of the Target data set and its outliers. The MoG approach uses the expectation maximization (EM) algorithm, which is known to be subject to such problems on univariate data sets [Ultsch et al., 2015]. Notably, only "if the underlying distribution comes from a mixture of component densities described by a set of unknown parameters" can it be estimated using MoG approaches [Duda et al., 2001, e.g. p. 581]. This is the case for the FCPS data sets, resulting in high performance of the MoG algorithm. However, natural data sets do not necessarily satisfyhave to meet this assumption. Additionally, the MoG method fails if the dimensionality of the data set is too high (chapter 3).
The automatic DBS clustering showed a small variance in its results and yielded good accuracy for all data sets. In contrast to all other approaches, in every trial in which the clustering accuracy of DBS was worse than that of some other algorithm, its performance could be improved by using the semi-interactive approach. The reason for this ability to improve the results of DBS lies in the main advantage of DBS clustering, namely, the possibility of verifying the clustering results through visualization, as described below. For a clustering algorithm, it is relevant to test for the absence of a cluster structure [Everitt et al., 2001, p. 180], or the clustering tendency [Theodoridis/Koutroumbas, 2009, p. 896]. Usually, tests for the clustering tendency rely on statistical tests [Theodoridis/Koutroumbas, 2009, p. 896]. Unlike other hierarchical clustering algorithms (except for ESOM/U-matrix clustering [Ultsch et al., 2016a]), the DBS algorithm finds no clusters if no natural clusters exist. The clustering tendency is visualized by the generalized U-matrix.

Generalized U-matrix visualization and structure preservation

The technique of producing visualizations in the form of a two-dimensional scatter plot of projected points currently remains the state of the art in cluster analysis (e.g., [Hennig et al., 2015, pp. 119-120, 683-684; Ritter, 2014, p. 223]). However, such a two-dimensional visualization can lead to a misleading interpretation of the underlying structures because the low-dimensional similarities do not completely represent the high-dimensional distances in two dimensions. Two types of error have been identified in the literature (see chapter 5): forward projection error (FPE) and backward projection error (BPE) [Aupetit, 2007; Ultsch/Herrmann, 2005; Venna et al., 2010]. In addition to these errors, this work introduces the concept of structure preservation, which is the preservation of high-dimensional discontinuities such that no points are allowed to intrude into the discontinuity regions of the two dimensional projection.
The FPEs and BPEs were visualized for various projection methods using a two-dimensional gray-scale U-matrix visualization in [Ultsch/Mörchen, 2006]. Such a gray-scale U-matrix is the most commonly used method for displaying dissimilarities in SOMs [K. Tasdemir/Merenyi, 2009, p. 550; Kadim Tasdemir/Merényi, 2012, p. 3]. Here, the idea was to "apply Self-Organizing Map training without changing the best matching unit [prototype] assignment"

[Ultsch/Mörchen, 2006, pp. 3-4] through the transformation of projected points into best matching units, as introduced in this work. Unlike the approach of Ultsch and Mörchen, the newly proposed simplified ESOM (sESOM) algorithm does not require a learning rate, and the cooling scheme is defined by a special neighborhood function based on symmetry considerations, which results in a parameter-free algorithm (cf. [Ultsch/Mörchen, 2006, p. 4]). This makes it possible to visualize SOMs as topographic maps with hypsometric tints [Thrun et al., 2016a], which serves as a basis for a visualization technique that can be applied in combination with any projection method. The third dimension is used to visualize the local BPE and FPE around each projected point in precisely defined height-dependent colors, thereby giving rise to the generalized U-matrix, which is a generalization of the U-map concept [Ultsch, 2003a].

Here, it is argued that the generalized U-matrix visualization of a topographic map (second DBS module) is able to visualize both compact and connected structures. In terms of the preservation of high-dimensional structures, it is a suitable approach for visualizing the BPEs, FPEs and discontinuities in a data set. However, as shown in Fig. 5.6 in chapter 5, this visualization technique has certain limitations. If additional gaps with intruding points are added by the projection method, then the generalized U-matrix is not able to distinguish identical clusters from distinct ones. To the author's knowledge, the only visualization that shows whether clusters have been disrupted uses a linear gray-scale approach based on a holistic solution called the proximity measure [Aupetit, 2007]. In the two-dimensional projected space, Voronoi cells are filled with brighter or darker luminances depending on their high-dimensional distances D to a reference point. "Points with bright cells are connected in the original space" [Aupetit, 2007, p. 17]. However, cluster disruption can only be successfully visualized when the user selects the correct reference point. To estimate the correct reference point for a projected space, additional visualizations of other measures, as introduced in this paper, must be used. Consequently, this process is both time-consuming and challenging and requires user supervision.

Many quality criteria exist for evaluating the visualization of a scatter plot. Chapter 6 addressed the question of whether the currently existing QMs are able to measure structure preservation. By using a generalized, graph-theory-based definition for a neighborhood of points, it is possible to group the QMs based on their semantic characterization. Here, 19 common QMs were reviewed and grouped, and they were compared with regard to their ability to measure the structure preservation of a projection. It is argued here that the QMs that have been presented in the literature have difficulty correctly capturing the discontinuities in high-dimensional data because of their inherent assumptions regarding the underlying high-dimensional structures. This was shown using the Hepta and Chainlink data sets in supplement *A*.

Otherwise, an objective function could be defined using the "best" QM, and it would always be possible to obtain a structure-preserving two-dimensional visualization by optimizing this objective function. In this work, no answer could be found to the question of how the quality of structure preservation can be automatically measured or visualized without prior knowledge.

However, when a prior classification of the data is available, it can be used to evaluate the quality of structure preservation. The structures that should be preserved are defined by such a classification. A QM called the Delaunay classification error (DCE) was developed based on this concept; it allows projections to be ranked and normalized compared with a baseline and also enables statistical testing.

In summary, structure preservation depends on the chosen projection method; however, the task of choosing the correct projection method is challenging because the optimization of an objective function requires the predefinition of the structures to be visualized. The generalized U-matrix is able to visualize the similarities and dissimilarities among high-dimensional data points in a scatter plot of the projected points (BPEs and FPEs), but it is unable to visualize the disruption of clusters, based on which the quality of structure preservation is defined.

The projection method Pswarm

The first module of the DBS framework is called Pswarm. Pswarm is a projection method that does not rely on an objective function. Similarly to SOP, Pswarm uses stigmergy and a swarm of DataBots because swarm techniques are known for their properties of flexibility and robustness [Bonabeau/Meyer, 2001; Şahin, 2004]. However, in contrast to SOP, which uses an ESOM-like grid space, the environment of the DataBots in Pswarm has been redefined based on symmetry considerations [Feynman et al., 2007, pp. 147-153, 745], resulting in the use of polar coordinates on a toroidal hexagonal grid. The combination of symmetry considerations with game theory concepts endows the polar swarm (Pswarm) with a parameter-free annealing process and an automatically selected, data-driven grid size.

The insights presented in chapter 7 demonstrate that Pswarm exhibits both self-organization and swarm intelligence. In the swarm-based techniques presented in the available literature, the swarms used for projection and/or clustering do not take advantage of both concepts (chapter 7.3, Figure 7.4). Moreover, no other reported swarm method exploits game theory or the phenomenon of emergence (as defined in chapter 7, section 3, after [Ultsch, 2007]). Here, the focus is placed on a subfield of dimensionality reduction in which projection methods are used for visualizing high-dimensional data in a two-dimensional space, as opposed to manifold learning methods, which are designed only to find manifolds, not to compress them into two-dimensional space [Venna et al., 2010, p. 2].

Of the methods of projecting high-dimensional data into two-dimensional space, two stand out: Neighborhood Retrieval Visualizer (NeRV) [Venna et al., 2010] and ESOM [Ultsch, 1999]. NeRV optimizes the objective function that quantifies the cost, defined as information retrieval, with the goal of visualizing the similarity relationships between data points. NeRV attempts to achieve a faithful representation of the data in two dimensions by minimizing the BPE and FPE. The cost is a tradeoff between the FPE and BPE[78], which is defined by the parameter λ. ESOM is an unsupervised neural learning algorithm and can be used as a projection method if a large number of neurons is specified. ESOM remains a reference tool for two-dimensional visualization [Lee/Verleysen, 2007, p. 244]. Instead of an objective function, ESOM uses the powerful concept of emergence [Ultsch, 2007] in addition to the 3D visualization technique of [Thrun et al., 2016a], which is based on the U-matrix [Ultsch, 2003a]. Both NeRV and ESOM are state-of-the-art methods for the visualization of high-dimensional data.

Pswarm was compared with the following common projection methods: principal component analysis (PCA), curvilinear component analysis (CCA), t-distributed stochastic neighbor embedding (t-SNE), ESOM, NeRV and the multidimensional scaling (MDS) technique of Sammon mapping. Five artificial three-dimensional data sets from the FCPS were used to compare these projection methods because of their clearly defined natural clusters. Typically, the QMs

[78] In information retrieval terms, precision and recall.

discussed in the literature indirectly assume that a projection method has a deterministic outcome. A problem that has, thus far, remained undiscussed is the stochastic outcomes of some common projection methods, such as t-SNE and CCA. Therefore, the DCEs were calculated for 100 trials per projection method and data set. Thus, the outcomes of the projection methods could be statistically compared. To enable an unbiased comparison, the DCE requires a prior classification that defines the structures in a data set. However, as discussed by [Färber et al., 2010], natural data sets may have more than one useful classification, depending on the context and the algorithm applied, because no universal definition of a cluster exists [Hennig, 2015b, p. 705]. Therefore, the evaluation of different projections methods by DCE only makes sense on artificial data sets with predefined natural clusters (see chapter 9). This is a major limitation of the DCE QM.

It was shown that the two-dimensional projections generated by Pswarm are comparable to those produced by the state-of-the-art methods NeRV and ESOM. To the author's knowledge, every projection method considered here (except ESOM and SOP) optimizes an objective function, which may lead to the disadvantages discussed above. Moreover, some projection methods, such as ESOM and CCA, use a sophisticated annealing scheme that may be sensitive to one or more parameters or have one or more sensitive parameters themselves (e.g., λ in NeRV). Examples are given in chapter 10.2, Tab. 10.1. In contrast to NeRV, Pswarm is not sensitive to any parameter or, as in the case of ESOM, to an annealing scheme and lattice size. It was shown that a projection with minimal BPE and FPE values does not necessarily achieve structure preservation. In the case of NeRV, it was shown that this algorithm is sensitive to its random initialization process (chapter 5, Fig. 5.6, and chapter 10). Venna et al. also proposed an alternative PCA-based initialization [Venna et al., 2010, p. 459], which in itself makes prior assumptions regarding the relevant structures of the high-dimensional data[79], as illustrated by the baseline used to analyze the DCE results (see chapter 10.2 Figure 10.5). Unlike NeRV, Pswarm does not visualize cluster structures if such structures do not exist in the data, as in the case of the Golf Ball data set (or the various continuous data sets presented in supplement D); moreover, because Pswarm is a swarm-based technique, it is more robust to the random initialization process (e.g., the DBS visualization of the leukemia data set in chapter 11, Figure 10.1).

In the third section of chapter 10, the SOP algorithm is emphasized because it is another method based on a swarm of DataBots, as introduced in [Herrmann, 2009]. In [Herrmann, 2011], it was shown that SOP is nearly as good as or even better than the best of its carefully parameterized competitor methods, namely, CCA, t-SNE and ESOM, in terms of the 1-nearest-neighbor classification accuracy and the specially formulated dispersion measure of [Herrmann, 2011, p. 101]. It was also noted that these methods resulted in severe misrepresentations of the structures for several data sets, which was not the case for SOP (see also the scatter plots in section A2 of [Herrmann, 2011, pp. 158-161]).

Notably, the annealing process of the SOP algorithm is not truly self-adaptive; rather, it is parameterized, which can lead to severe errors in the projections. In the best case, the choice of the lattice size and, therefore, the maximal neighborhood radius as well as the choices of the two magic numbers (the jumping DataBots threshold and the maximum number of iterations) in the SOP algorithm have only a minor effect on the visualization of the high-dimensional

[79] PCA maximizes the variance.

structures (as in the cases of the Atom and Chainlink data sets). In the worst case, as for the EngyTime or Iris data set, all structures are prevented from emerging. Moreover, in the case of EngyTime, it was shown that when there is no restriction ensuring that no more than one DataBot can occupy each lattice position, the information about the high-dimensional structure is lost. Unlike the dispersion measure and 1-nearest-neighbor classification approach of Herrmann, in comparison with SOP and based on a topographic map of projected points, the visualizations presented in this work illustrate important improvements achieved by Pswarm, which are described in the last section of chapter 10.

Several examples were presented to demonstrate that the process leading to emergence is disrupted in the SOP algorithm. Other swarms do not exhibit self-organization but instead rely on the optimization of an objective function, which makes emergence impossible. To the author's knowledge, the game theory approach to behavior-based systems remains undiscussed in the available literature on artificial intelligence in data science. The naturally clustered Wine, Swiss Banknotes and Iris data sets all illustrate the importance of consistent and appropriate definitions of the neighborhoods, scents, grid or lattice size and data-driven annealing scheme used for clustering and projection. If these definitions are oblique, as is the case for SOP, then the self-organization of the DataBots is disrupted. The ultimate disruption of the process leading to emergence may be minor (Swiss Banknotes) or major (Wine, Iris), depending on the data set and the specific trial. For the Wine data set, Pswarm gains an advantage because of the ability to choose different a distance whereas the SOP algorithm does not. [Herrmann, 2011, p. 65]. Pswarm allows the user to define a non-metric distance method without any restrictions.

The correct selection of the parameters for the annealing scheme requires an experienced user. For example, it was shown that with the default settings, the ESOM algorithm sometimes projects three, instead of two, clusters for the Atom data set (chapter 5, Fig. 5.6). To further substantiate this argument, additional ESOM projections generated with the default parameters are presented in Supplement E. For example, it is necessary to change the lattice type from toroidal (default) to planar to achieve a correct projection of the Wing Nut data set. If the default parameters are not changed, the structures are very difficult to see. Disruption of the clusters can be seen in the ESOM/U-matrix visualizations of the Iris, Wine, and Swiss Banknotes data sets, in which one or more of the other eight parameters play an important role (see supplement C for these U-matrix visualizations).

Thus, it is argued here that the ESOM/U-matrix projections of the EngyTime, Wing Nut, Iris, Wine and Swiss Banknotes data sets may be misleading because the toroidal ESOM projections are computed without accounting for symmetry considerations, which results in unwanted boundary effects. For example, the maximal radius is set to the diagonal length[80] $\sqrt{L^2 + C^2}$ instead of $L / 2$, which leads to overlapping of the neighborhoods if the neighborhood function is defined as Gaussian. Several examples illustrate that the uniform distribution used in the ESOM and SOP algorithms has no advantages; however, it may have some disadvantages. The attempt to distribute the projected points uniformly on the lattice is useful only if a visualization method is able to reveal the high-dimensional structures of the data. For this reason, the U-matrix visualization [Ultsch, 2003a] is mandatory for ESOM projections. In other cases, uni-

[80] L is the number of lines in the grid, and C is the number of columns.

formly distributed projected points do not lead to new knowledge about the data set. By contrast, for the generalized U-matrix, there is no requirement for the projected points to be uniformly distributed. Consequently, Pswarm outperforms ESOM on density-based data sets such as EngyTime.

Being a swarm-based method, DBS suffers from the disadvantage of high computational costs. When the number of DataBots[81] is greater than 4000, the use of Pswarm is impractical because of the long calculation time. Further research is necessary on the application of game theory as the foundation for a data-driven annealing scheme. At this point, it can be proven only that a weak Nash equilibrium will be found [Nash, 1951], which may be the reason for the high variance observed in the DCE results (chapter 10, section 2). Only with DBS clustering can the variance of the results be noticeably improved. The structures of 14 of the investigated data sets were preserved using Pswarm (chapters 10 and 11).

The main drawbacks of the proposed approach are as follows. If no prior classification is available for a data set, then the use of DCE measure is limited. Thus, it is very difficult to evaluate whether Pswarm and the generalized U-matrix produce a structure-preserving visualization or whether the clusters are disrupted in the visualization. Additionally, the variance of the results remains high: because it is a stochastic projection method, two different trials of Pswarm could yield different visualizations of the same data set. If the number of clusters is known beforehand, *deep swarming* may be able to solve this problem, as the Tetragonula data set demonstrated[82]. Moreover, it should be possible for the swarm to iteratively add new data points during or after the algorithm following a well-defined process**.** At present, the Pswarm algorithm is unable to do this. Briefly, it was demonstrated in sections 2 and 3 of chapter 10 that finding the correct grid or lattice size and annealing scheme for ESOM/SOP may be challenging. It should be emphasized that unlike SOP and, especially, ESOM (see supplement C and E), Pswarm is able to successfully project density-based data sets. The comparison between Pswarm and the other common projection methods with their default parameter settings resulted in two major findings. First, the state-of-the-art methods ESOM and NeRV do not outperform Pswarm, and second, Pswarm has one important advantage, namely, that it is parameter-free. However, if prior knowledge of the data set to be analyzed is available, then a projection method that is appropriately chosen with regard to the structures that should be preserved will always outperform Pswarm. Furthermore, other projection methods may also outperform Pswarm if their settings are carefully selected by an experienced user. In summary, to the author's knowledge, Pswarm is the first swarm-based technique to show emergent properties while simultaneously combining swarm intelligence, self-organization and game theory.

Knowledge discovery with DBS

Up to this point, mainly artificial data sets have been used to assess the capabilities of DBS. In the case of natural data sets, only the prior classifications were considered. However, the introduction of a new clustering method is necessary only if it is useful. Therefore, three complex real world data sets were first analyzed using DBS to confirm its ability to reproduce known knowledge. Subsequently, two high-dimensional data sets were clustered using DBS to obtain

[81] Which is equal to the number of high-dimensional data points.
[82] for details see next section or chapter 11, section 3.

new knowledge. The silhouette plots and the heatmaps, which showed small intracluster distances and large intercluster distances, indicated that the clustering results for all five data sets were valid.

The visualization and connected clustering of the high-dimensional[83] leukemia data set, which contains clearly defined natural clusters (see chapter 3), successfully reproduced the diagnoses of three types of leukemia: acute myeloid leukemia (AML), acute promyelocytic leukemia (APL) and chronic lymphocytic leukemia (CLL). Aside from two outliers (patients), the prior classification of healthy patients and patients diagnosed with the three leukemia subtypes was reproduced by the DBS clustering and visualization. The two outlier patients may be misdiagnosed; however, a future publication will address this diagnostic problem. Chapter 6 showed that aside from ESOM, no other common projection method was able to visualize the predefined cluster structure of this data set. Similarly, in chapter 3, it was demonstrated that common clustering algorithms failed to correctly cluster the leukemia data set, with the exception of the Ward algorithm, which was not able to find the two outliers.

When the dynamic time-warping distance definition was applied on a data set consisting of the gross domestic product (GDP) per capita in 190 countries for the years 1970–2010, two clusters and one outlier were found using DBS. Upon the application of Classification and Regression Tree (CART) analysis, it was found that the two clusters could be explained as being distinguished by the influence of the tragic event of planes crashing into the World Trade Center in 2001.

DBS found 10 clusters in the Tetragonula data set, as verified by the heatmap and silhouette plot. When the largest within-cluster gap, the cluster separation, and the average within-cluster dissimilarity of [Hennig, 2014] were calculated, the resulting values were the minima reported in [Hennig, 2014], presented there in Fig. 4. The 10 identified clusters strongly depended on the locations of the bees (chapter 11, Figure 11.8). Additionally, the application of DBS to this data set illustrated the possibility of using multiple swarms by means of parallel computing, for which the term *deep swarming* (see [Ultsch, 2016b]) is introduced here in analogy to deep learning [Goodfellow et al., 2016]. Here, deep swarming was applied with a DCE-based objective function, but it can also be applied in combination with any arbitrary objective function.

For the hydrology data set, the daily courses were analyzed. After preprocessing, DBS identified five distinct clusters (chapter 12, Figure 11.4), which were verified by the heatmap and silhouette plot. The rules extracted from a CART decision tree were applied to the clustering of this data set and found to result in the misclassification of 0.9% of the points (chapter 12, Figure 12.6). Five different water quality states in terms of nitrate concentration and electrical conductivity were identified based on a semantic characterization of these clusters (chapter 12, Figure 12.7). The extracted rules enable the prediction of future nitrate and electrical conductivity conditions.

For the pain gene data set, focus was placed on the task of clustering the pain genes. The distances between genes were defined based on the inverse document frequency (idf) [Sparck Jones, 1972] and the information available in the Gene Ontology (GO) database. The DBS clustering resulted in eight clusters (Figure 12.9). Five clusters reproduced the previously known functions of the pain genes (Tab 12.2), as described in section 12.2.1. Outliers were

[83] Containing 7747 variables.

found in two clusters, and one cluster yielded new discoveries regarding the functions of pain genes (Tab 12.2, C5). This cluster was characterized by the downregulation of metabolic processes and the upregulation of the creatine metabolic process.

> *"The experience from many knowledge discovery tasks ([Behnisch/Ultsch, 2009; Kupas et al., 2004; Lötsch/Ultsch, 2013; Mörchen et al., 2005]) is that about 80% of clusters coincide with known processes. Typically about 10% may be attributed to erroneous data, while the remaining 10% may generate entirely new knowledge" [Behnisch/Ultsch, 2015, p. 68].*

This experience is consistent with the findings obtained in the above examples. Two domain experts found the results presented above to be valid and useful.

14 Conclusion

A new and data-driven approach for cluster analysis and visualization is introduced in this work. The projection based clustering combines structures preserved in two dimensions with underlying high-dimensional structures (see also [Thrun et al., 2017, Thrun/Ultsch, 2017a]). It is a flexible and robust approach for cluster analysis that consists of three independent modules which can be optionally combined into the Databionic swarm (DBS). Here, the attention is focused on data for which the generation process is complete and for which the size and amount of information can be managed using a personal computer with standard hardware; consequently, the realm of Big Data is not discussed here. To the author's knowledge, DBS is the first swarm-based technique showing emergent properties while simultaneously exploiting the concepts of swarm intelligence, self-organization and the Nash equilibrium concept from game theory, which results in the elimination of a global objective function and of the setting of parameters.

Alternatively, the visualization by the generalized Umatrix and the DBS clustering can be applied to every projection method for connected or compact structures based on discontinuities of high-dimensional data [Thrun/Ultsch, 2017a]. Through the use of the generalized Umatrix visualization, results of common clustering methods can be verified by the structures found by the data-driven Pswarm or any other projection method.

This work introduced the fundamental principle of considering compact versus connected structures in the clustering of data. However, in this context, only unsupervised indices, called QMs for projection methods, were analyzed. A similar analysis of supervised indices should be conducted in the future with the help of the FCPS. There is sufficient literature available to do so (e.g., [Charrad et al., 2012; Dimitriadou et al., 2002; Handl et al., 2005]).

Another goal of future research should be to find a strong Nash equilibrium. However, a strong Nash equilibrium is mathematically difficult to prove. In the opinion of the author, if each DataBot were able to assess all possible jump positions in a given neighborhood instead of only four, then a strong Nash equilibrium could be achieved. However, the time complexity of this approach is too high for practical testing unless the algorithm is parallelized. Additionally, deep swarming should be extensively tested.

Symmetry considerations were applied to the two-dimensional toroidal output space, resulting in the use of polar coordinates in the DBS framework. Additionally, it should be possible to explore and exploit connections with solid-state physics. Perhaps it would be beneficial to define the Bravais lattice, apply a Fourier transformation to the reciprocal lattice [Hunklinger, 2009, pp. 83-88], and perform calculations in the reciprocal space, where boundary effects could be easily eliminated and a low computational time complexity could be achieved.

Further research on these possibilities is required.

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_14

References

[Abraham et al., 2006] **Abraham, A., Guo, H., & Liu, H.**: Swarm intelligence: foundations, perspectives and applications, In Nedjah, N. & Mourelle, L. d. M. (Eds.), *Swarm Intelligent Systems*, pp. 3-25, Springer, **2006**.

[Aeberhard et al., 1992] **Aeberhard, S., Coomans, D., & De Vel, O.**: Comparison of classifiers in high dimensional settings, *Dept. Math. Statist., James Cook Univ., North Queensland, Australia, Tech. Rep, Vol.* (92-02), **1992.**

[Aho et al., 2014] **Aho, K., Derryberry, D., & Peterson, T.**: Model selection for ecologists: the worldviews of AIC and BIC, *Ecology, Vol. 95*(3), pp. 631-636, **2014.**

[Aikins, 1983] **Aikins, J. S.**: Prototypical knowledge for expert systems, *Artificial intelligence, Vol. 20*(2), pp. 163-210, **1983.**

[Akkucuk/Carroll, 2006] **Akkucuk, U., & Carroll, J. D.**: PARAMAP vs. Isomap: a comparison of two nonlinear mapping algorithms, *Journal of Classification, Vol. 23*(2), pp. 221-254, **2006.**

[Alhoniemi, et al., 2005] **Alhoniemi E., Himberg J., Parhankangas, J. & Vesanto J.:** SOM Toolbox 2.1, in Matlab under GPL, *http://www.cis.hut.fi/projects/somtoolbox/*, Version 2.1, Retrieved 10.10.2015, **2005.**

[Anderson, 1935] **Anderson, E.**: The Irises of the Gaspé Peninsula, *Bulletin of the American Iris Society, Vol. 59*, pp. 2-5, **1935.**

[Aparna/Nair, 2014] **Aparna, K., & Nair, M. K.**: Enhancement of K-Means algorithm using ACO as an optimization technique on high dimensional data, Proc. International Conference on Electronics and Communication Systems (ICECS), pp. 1-5, IEEE, **2014**.

[Arabie et al., 1996] **Arabie, P., Hubert, L. J., & De Soete, G.**: *Clustering and classification*, Singapore, World Scientific, ISBN: 9810212879, **1996**.

[Arimond/Elfessi, 2001] **Arimond, G., & Elfessi, A.**: A clustering method for categorical data in tourism market segmentation research, *Journal of Travel Research, Vol. 39*(4), pp. 391-397. **2001.**

[Arumugam et al., 2005] **Arumugam, M. S., Chandramohan, A., & Rao, M.**: Competitive approaches to PSO algorithms via new acceleration co-efficient variant with mutation operators, Proc. Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05), pp. 225-230, IEEE, **2005**.

[Asar et al., 2014] **Asar, Ö., Ilk, O., & Dag, O.**: Estimating Box-Cox power transformation parameter via goodness of fit tests, *Communications in Statistics-Simulation and Computation, Vol.*46.1, pp. 91-105, **2014.**

[Ashburner et al., 2000] **Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., . . . Eppig, J. T.**: Gene Ontology: tool for the unification of biology, *Nature genetics, Vol. 25*(1), pp. 25-29, **2000.**

[Aubert, 2015] **Aubert, A. H.**: "Introduction into hydrology variables", personal correspondence, 21.04.2015, **2015**.

[Aubert et al., 2016] **Aubert, A. H., Thrun, M. C., Breuer, L., & Ultsch, A.**: Knowledge discovery from high-frequency stream nitrate concentrations: hydrology and biology contributions, *Scientific reports, Vol. 6*(31536), doi 10.1038/srep31536, **2016.**

[Aupetit, 2003] **Aupetit, M.**: Robust Topology Representing Networks, Proc. ESANN, pp. 45-50, **2003**.

[Aupetit, 2007] **Aupetit, M.**: Visualizing distortions and recovering topology in continuous projection techniques, *Neurocomputing, Vol. 70*(7), pp. 1304-1330, **2007.**

[Backes et al., 2007] **Backes, C., Keller, A., Kuentzer, J., Kneissl, B., Comtesse, N., Elnakady, Y. A., . . . Lenhof, H.-P.**: GeneTrail—advanced gene set enrichment analysis, *Nucleic acids research, Vol. 35*(suppl 2), pp. W186-W192, **2007.**

[Baggenstoss, 2002] **Baggenstoss, p. M.**: Statistical modeling using gaussian mixtures and HMMs with Matlab, *Naval Undersea Warfare Center, Newport RI*, **2002.**

[Basu et al., 2008] **Basu, S., Davidson, I., & Wagstaff, K.**: *Constrained clustering: Advances in algorithms, theory, and applications*, CRC Press, ISBN: 1584889977, **2008**.

[Bauer et al., 1999] **Bauer, H.-U., Herrmann, M., & Villmann, T.**: Neural maps and topographic vector quantization, *Neural networks, Vol. 12*(4), pp. 659-676, **1999.**

[Bauer/Pawelzik, 1992] **Bauer, H.-U., & Pawelzik, K. R.**: Quantifying the neighborhood preservation of self-organizing feature maps, *Neural Networks, IEEE Transactions on, Vol. 3*(4), pp. 570-579, **1992.**

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9

[Beaton et al., 2010] **Beaton, D., Valova, I., & MacLean, D.**: CQoCO: A measure for comparative quality of coverage and organization for self-organizing maps, *Neurocomputing, Vol. 73*(10), pp. 2147-2159, **2010.**

[Beckers et al., 1994] **Beckers, R., Holland, O., & Deneubourg, J.-L.**: From local actions to global tasks: Stigmergy and collective robotics, Proc. Artificial life IV, Vol. 181, pp. 189, **1994**.

[Behnisch/Ultsch, 2009] **Behnisch, M., & Ultsch, A.**: Urban data-mining: spatiotemporal exploration of multidimensional data, *Building Research & Information, Vol. 37*(5-6), pp. 520-532, **2009.**

[Behnisch/Ultsch, 2015] **Behnisch, M., & Ultsch, A.**: Knowledge Discovery in Spatial Planning Data: A Concept for Cluster Understanding, *Computational Approaches for Urban Environments*, pp. 49-75, Springer, **2015**.

[Bellman, 1957] **Bellman, R.**: Dynamic programming: Princeton Univ. press, Princeton, **1957**.

[Bene et al., 1995] **Bene, M., Castoldi, G., Knapp, W., Ludwig, W., Matutes, E., Orfao, A., & Van't Veer, M.**: Proposals for the immunological classification of acute leukemias. European Group for the Immunological Characterization of Leukemias (EGIL), *Leukemia, Vol. 9*(10), pp. 1783-1786. **1995.**

[Bennett et al., 1985] **Bennett, J. M., Catovsky, D., Daniel, M. T., Flandrin, G., Galton, D. A., Gralnick, H. R., & Sultan, C.**: Proposed revised criteria for the classification of acute myeloid leukemia A report of the French-American-British Cooperative Group, *Annals of internal medicine, Vol. 103*(4), pp. 620-625. **1985.**

[Beni, 2004] **Beni, G.**: From swarm intelligence to swarm robotics, Proc. International Workshop on Swarm Robotics, pp. 1-9, Springer, **2004**.

[Beni/Wang, 1989] **Beni, G., & Wang, J.**: Swarm Intelligence in Cellular Robotic Systems, Proc. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, **1989**.

[Beni/Wang, 1993] **Beni, G., & Wang, J.**: Swarm intelligence in cellular robotic systems, *Robots and Biological Systems: Towards a New Bionics?*, pp. 703-712, Springer, **1993**.

[Benyus, 2002] **Benyus, J.**: Biomimicry: innovation inspired by design, New York: Harper Perennial, **2002**.

[Bezdek/Pal, 1993] **Bezdek, J. C., & Pal, N. R.**: An index of topological preservation and its application to self-organizing feature maps, Proc. Neural Networks, 1993. IJCNN'93-Nagoya. Proceedings of 1993 International Joint Conference on, Vol. 3, pp. 2435-2440, IEEE, **1993**.

[Bezdek/R Pal, 1995] **Bezdek, J. C., & R Pal, N.**: An index of topological preservation for feature extraction, *Pattern Recognition, Vol. 28*(3), pp. 381-391, **1995.**

[Bishop, 2006] **Bishop, C. M.**: Pattern recognition, *Machine Learning, Vol. 128*, **2006.**

[Bock, 1974] **Bock, H. H.**: Automatische Klassifikation: Theoret. u. prakt. Methoden z. Gruppierung u. Strukturierung von Daten (Cluster-Analyse), (Vol. XXIV), Göttingen, Germany, Vandenhoeck & Ruprecht, ISBN: 3-525-40130-2, **1974**.

[Bogon, 2013] **Bogon, T.**:*Agentenbasierte Schwarmintelligenz,* (Phd Dissertation), Springer-Verlag, Trier, Germany**, 2013**.

[Bonabeau et al., 1999] **Bonabeau, E., Dorigo, M., & Theraulaz, G.**: *Swarm intelligence: from natural to artificial systems*, New York, Oxford University Press, ISBN: 978-0-19-513159-8, **1999**.

[Bonabeau/Meyer, 2001] **Bonabeau, E., & Meyer, C.**: Swarm intelligence: A whole new way to think about business, *Harvard business review, Vol. 79*(5), pp. 106-115, **2001.**

[Borken/Matzner, 2009] **Borken, W., & Matzner, E.**: Reappraisal of drying and wetting effects on C and N mineralization and fluxes in soils, *Global Change Biology, Vol. 15*(4), pp. 808-824, **2009.**

[Bouveyron/Brunet-Saumard, 2014] **Bouveyron, C., & Brunet-Saumard, C.**: Model-based clustering of high-dimensional data: A review, *Computational Statistics & Data Analysis, Vol. 71*, pp. 52-78, **2014.**

[Bouveyron et al., 2012] **Bouveyron, C., Hammer, B., & Villmann, T.**: Recent developments in clustering algorithms, Proc. ESANN, Citeseer, **2012**.

[Bowcock et al., 1994] **Bowcock, A. M., Ruiz-Linares, A., Tomfohrde, J., Minch, E., Kidd, J. R., & Cavalli-Sforza, L. L.**: High resolution of human evolutionary trees with polymorphic microsatellites, *Nature, Vol. 368*(6470), pp. 455-457, **1994.**

[Breiman et al., 1984] **Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A.**: *Classification and regression trees*, CRC press, ISBN: 0412048418, **1984**.

[Brendel, 2016] **Brendel, C.**: "About the Statements of Consent", personal correspondence, 18.11.2016, **2016**.

[Brito et al., 1997] **Brito, M., Chávez, E., Quiroz, A., & Yukich, J.**: Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection, *Statistics & Probability Letters, Vol. 35*(1), pp. 33-42, **1997.**

[Bronstein et al., 2005] **Bronstein, I. N., Hromkovic, J., Luderer, B., Schwarz, H.-R., Blath, J., Schied, A., . . . Zeidler, E.**: *Taschenbuch der mathematik*, (6th edition ed. Vol. 1), Springer-Verlag, ISBN: 3-8171-2006-0, **2005**.

[Brooks, 1991] **Brooks, R. A.**: Intelligence without representation, *Artificial intelligence, Vol. 47*(1), pp. 139-159, **1991.**

[Buhl et al., 2006] **Buhl, J., Sumpter, D. J., Couzin, I. D., Hale, J. J., Despland, E., Miller, E., & Simpson, S. J.**: From disorder to order in marching locusts, *Science, Vol. 312*(5778), pp. 1402-1406, **2006.**

[Bunte et al., 2012] **Bunte, K., Biehl, M., & Hammer, B.**: A general framework for dimensionality-reducing data visualization mapping, *Neural Computation, Vol. 24*(3), pp. 771-804, **2012.**

[Button et al., 2013] **Button, K. S., Ioannidis, J. P., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., & Munafò, M. R.**: Power failure: why small sample size undermines the reliability of neuroscience, *Nature Reviews Neuroscience, Vol. 14*(5), pp. 365-376, **2013.**

[Camon et al., 2003] **Camon, E., Magrane, M., Barrell, D., Binns, D., Fleischmann, W., Kersey, P., . . . Cox, A.**: The gene ontology annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro, *Genome research, Vol. 13*(4), pp. 662-672, **2003.**

[Camon et al., 2004] **Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., . . . Apweiler, R.**: The gene ontology annotation (goa) database: sharing knowledge in uniprot with gene ontology, *Nucleic acids research, Vol. 32*(suppl 1), pp. D262-D266, **2004.**

[Cao et al., 1997] **Cao, Y. U., Fukunaga, A. S., & Kahng, A.**: Cooperative mobile robotics: Antecedents and directions, *Autonomous robots, Vol. 4*(1), pp. 7-27, **1997.**

[Charrad et al., 2012] **Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A.**: NbClust Package: finding the relevant number of clusters in a dataset, *Journal of statistical Software, Vol. 61*(6), doi 10.18637/jss.v061.i06, **2012.**

[Chen/Buja, 2006] **Chen, L., & Buja, A.**:*Local multidimensional scaling for nonlinear dimensionality reduction, graph layout, and proximity analysis,* (PhD thesis), University of Pennsylviana, USA**, 2006**.

[Chen/Buja, 2009] **Chen, L., & Buja, A.**: Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis, *Journal of the American Statistical Association, Vol. 104*(485), pp. 209-219, **2009.**

[Cheng, 1997] **Cheng, Y.**: Convergence and ordering of Kohonen's batch map, *Neural Computation, Vol. 9*(8), pp. 1667-1676, **1997.**

[Cirmo/McDonnell, 1997] **Cirmo, C. P., & McDonnell, J. J.**: Linking the hydrologic and biogeochemical controls of nitrogen transport in near-stream zones of temperate-forested catchments: a review, *Journal of Hydrology, Vol. 199*(1), pp. 88-120, **1997.**

[Clark et al., 1990] **Clark, B. N., Colbourn, C. J., & Johnson, D. S.**: Unit disk graphs, *Discrete mathematics, Vol. 86*(1-3), pp. 165-177, **1990.**

[Colorimetry, 2004] **Colorimetry**.C.I.E., CIE Publication,Central Bureau of the CIE, Vienna, **2004**.

[Conover, 1971] **Conover, W. J.**: *Practical nonparametric statistics*, New York, USA, John Wiley & Sons, ISBN, **1971**.

[Coretto/Hennig, 2016] **Coretto, P., & Hennig, C.**: Robust improper maximum likelihood: tuning, computation, and a comparison with other methods for robust Gaussian clustering, J*ournal of the American Statistical Association,* Vol. 111(516), pp. 1648-1659, **2016.**

[Cormack, 1971] **Cormack, R. M.**: A review of classification, *Journal of the Royal Statistical Society. Series A (General),* pp. 321-367, **1971.**

[Cottrell, 1996] **Cottrell, M.**: A {K} ohonen Map Representation to Avoid Misleading Interpretations, Proc., D facto conference services, **1996**.

[Cottrell et al., 2016] **Cottrell, M., Olteanu, M., Rossi, F., & Villa-Vialaneix, N.**: Theoretical and applied aspects of the self-organizing maps, *Advances in Self-Organizing Maps and Learning Vector Quantization*, pp. 3-26, Springer, **2016**.

[de Buitléir et al., 2012] **de Buitléir, A., Russell, M., & Daly, M.**: Wains: A pattern-seeking artificial life species, *Artificial Life, Vol. 18*(4), pp. 399-423, **2012.**

[Delaunay, 1934] **Delaunay, B.**: Sur la sphere vide, *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk, Vol. 7*(793-800), pp. 1-2, **1934.**

[Demartines/Hérault, 1995] **Demartines, P., & Hérault, J.**: CCA:" Curvilinear component analysis", Proc. 15° Colloque sur le traitement du signal et des images, FRA, 1995, Vol. 199, GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, **1995**.

[Deneubourg et al., 1991] **Deneubourg, J.-L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., & Chrétien, L.**: The dynamics of collective sorting robot-like ants and ant-like robots, Proc. Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats, pp. 356-363, **1991**.

[Diaz, 2001] **Diaz, R. J.**: Overview of hypoxia around the world, *Journal of environmental quality, Vol. 30*(2), pp. 275-281, **2001.**

[Dijkstra, 1959] **Dijkstra, E. W.**: A note on two problems in connexion with graphs, *Numerische mathematik, Vol. 1*(1), pp. 269-271, **1959.**

[Dimitriadou et al., 2002] **Dimitriadou, E., Dolničar, S., & Weingessel, A.**: An examination of indexes for determining the number of clusters in binary data sets, *Psychometrika, Vol. 67*(1), pp. 137-159, **2002.**

[Dimitriadou/Hornik 2017] **Dimitriadou, E. & Kurt Hornik**: cclust: Convex Clustering Methods and Clustering Indexes, in R under GPL-2, Version 0.6-21, https://cran.r-project.org/web/packages/cclust/index.html, **2017**.

[Dirac, 1981] **Dirac, p. A. M.**: *The principles of quantum mechanics*, Oxford university press, ISBN: 0198520115, **1981**.

[Doherty et al., 2006] **Doherty, K., Adams, R., & Davey, N.**: Topological correlation, Proc. ESANN, pp. 125-130, Citeseer, **2006**.

[Drygas, 1978] **Drygas, H.**: Über multidimensionale Skalierung, *Statistical Papers, Vol. 19*(1), pp. 63-66, **1978.**

[Duda et al., 2001] **Duda, R. O., Hart, p. E., & Stork, D. G.**: *Pattern classification*, (Second Edition ed.), Ney York, USA, John Wiley & Sons, ISBN: 0-471-05669-3, **2001**.

[Durbin/Mitchison, 1990] **Durbin, R., & Mitchison, G.**: A dimension reduction framework for understanding cortical maps, *Nature, Vol. 343*(6259), pp. 644-647, **1990.**

[Eberhart et al., 2001] **Eberhart, R. C., Shi, Y., & Kennedy, J.**: Swarm Intelligence (The Morgan Kaufmann Series in Evolutionary Computation)*, Elsevier,* **2001.**

[Esmin et al., 2015] **Esmin, A. A., Coelho, R. A., & Matwin, S.**: A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data, *Artificial Intelligence Review, Vol. 44*(1), pp. 23-45, **2015.**

[Ester et al., 1996] **Ester, M., Kriegel, H.-P., Sander, J., & Xu, X.**: A density-based algorithm for discovering clusters in large spatial databases with noise, Proc. Kdd, Vol. 96, pp. 226-231, **1996**.

[Everitt et al., 2001] **Everitt, B. S., Landau, S., & Leese, M.**: *Cluster analysis*, (McAllister, L. Ed. Fourth Edition ed.), London, Arnold, ISBN: 0 340 76119 9, **2001**.

[Färber et al., 2010] **Färber, I., Günnemann, S., Kriegel, H.-P., Kröger, P., Müller, E., Schubert, E., . . . Zimek, A.**: On using class-labels in evaluation of clusterings, Proc. MultiClust: 1st international workshop on discovering, summarizing and using multiple clusterings held in conjunction with KDD, pp. 1, **2010**.

[Fathian/Amiri, 2008] **Fathian, M., & Amiri, B.**: A honeybee-mating approach for cluster analysis, *The International Journal of Advanced Manufacturing Technology, Vol. 38*(7-8), pp. 809-821, **2008.**

[Fayyad et al., 1996] **Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R.**: *Advances in knowledge discovery and data mining*, (Vol. 21), Menlo Park, California, USA, American Association for Artificial Intelligence press, ISBN: 0-262-56ß97-6, **1996**.

[Feynman et al., 2006] **Feynman, R. P., Leighton, R. B., & Sands, M.**: *Quantenmechanik*, (Köhler, K.-H., Schröder, K.-E. & Beckmann, W. B. P., Trans. 5th Edition: Definitive Edition ed. Vol. 3), Munich, Germany, Oldenbourg Verlag, ISBN: 978-3-486-58108-6, **2006**.

[Feynman et al., 2007] **Feynman, R. P., Leighton, R. B., & Sands, M.**: *Mechanik, Strahlung, Wärme*, (Köhler, K.-H., Schröder, K.-E. & Beckmann, W. B. P., Trans. 5th Edition: Definitive Edition ed. Vol. 1), Munich, Germany, Oldenbourg Verlag, ISBN: 978-3-486-58108-9, **2007**.

[Fisher, 1936] **Fisher, R. A.**: The use of multiple measurements in taxonomic problems, *Annals of eugenics, Vol. 7*(2), pp. 179-188, **1936.**

[Florek et al., 1951] **Florek, K., Łukaszewicz, J., Perkal, J., Steinhaus, H., & Zubrzycki, S.**: Sur la liaison et la division des points d'un ensemble fini, Proc. Colloquium Mathematicae, Vol. 2, pp. 282-285, Institute of Mathematics Polish Academy of Sciences, **1951**.

[Flury/Riedwyl, 1988] **Flury, B., & Riedwyl, H.**: *Multivariate statistics, a practical approach*, London, Chapman and Hall, ISBN, **1988**.

[Forest, 1990] **Forest, S.**: Emergent computation: self-organizing, collective, and cooperative phenomena in natural and artificial computing networks, *Physica D, Vol. 42*, pp. 1-11, **1990.**

[Fort et al., 2001] **Fort, J.-C., Cottrell, M., & Letremy, P.**: Stochastic on-line algorithm versus batch algorithm for quantization and self organizing maps, Proc. Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop, pp. 43-52, IEEE, **2001**.

[Fraley/Raftery, 2002] **Fraley, C., & Raftery, A. E.**: Model-based clustering, discriminant analysis, and density estimation, *Journal of the American Statistical Association, Vol. 97*(458), pp. 611-631, **2002.**

[Fraley/Raftery, 2006] **Fraley, C., & Raftery, A. E.**MCLUST version 3: an R package for normal mixture modeling and model-based clustering,DTIC Document, **2006**.

[Fraley et al., 2017] **Fraley, C., Raftery, A.E., Scrucca, L., Murphy, T.B. & Fop, M.**: mclust: Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation, in R under GPL-3, Version 5.3, https://cran.r-project.org/web/packages/mclust/index.html, **2017**.

[Franck et al., 2004] **Franck, P., Cameron, E., Good, G., RASPLUS, J. Y., & Oldroyd, B.**: Nest architecture and genetic differentiation in a species complex of Australian stingless bees, *Molecular Ecology, Vol. 13*(8), pp. 2317-2331, **2004.**

[Fraser, 2006] **Fraser, G.**: *The New Physics: For the Twenty-First Century*, USA, Cambridge University Press, ISBN: 978-0-521-81600-9, **2006**.

[Gabriel/Sokal, 1969] **Gabriel, K. R., & Sokal, R. R.**: A new statistical approach to geographic variation analysis, *Systematic Biology, Vol. 18*(3), pp. 259-278, **1969.**

[Garnier et al., 2007] **Garnier, S., Gautrais, J., & Theraulaz, G.**: The biological principles of swarm intelligence, *Swarm Intelligence, Vol. 1*(1), pp. 3-31, **2007.**

[Ge et al., 2007] **Ge, R., Ester, M., Jin, W., & Davidson, I.**: Constraint-driven clustering, Proc. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 320-329, ACM, **2007**.

[Giorgino, 2009] **Giorgino, T.**: Computing and visualizing dynamic time warping alignments in R: the dtw package, *Journal of statistical Software, Vol. 31*(7), pp. 1-24, **2009.**

[Giraldo et al., 2011] **Giraldo, L. F., Lozano, F., & Quijano, N.**: Foraging theory for dimensionality reduction of clustered data, *Machine Learning, Vol. 82*(1), pp. 71-90, **2011.**

[Goeman/Mansmann, 2008] **Goeman, J. J., & Mansmann, U.**: Multiple testing on the directed acyclic graph of gene ontology, *Bioinformatics, Vol. 24*(4), pp. 537-544, **2008.**

[Goldstein, 1999] **Goldstein, J.**: Emergence as a construct: History and issues, *Emergence, Vol. 1*(1), pp. 49-72.,**1999.**

[Goodfellow et al., 2016] **Goodfellow, I., Bengio, Y., & Courville, A.**: *Deep learning*, Cambridge, Massachusetts, USA, MIT press, ISBN, **2016**.

[Goodhill et al., 1995] **Goodhill, G. J., Finch, S., & Sejnowski, T. J.**: Quantifying neighbourhood preservation in topographic mappings*, Proceedings of the 3rd Joint Symposium on Neural Computation, Vol. 6*, pp. 61-82, **1995.**

[Gracia et al., 2014] **Gracia, A., González, S., Robles, V., & Menasalvas, E.**: A methodology to compare Dimensionality Reduction algorithms in terms of loss of quality, *Information Sciences, Vol. 270*, pp. 1-27, **2014.**

[Grassberger/Procaccia, 1983] **Grassberger, P., & Procaccia, I.**: Estimation of the Kolmogorov entropy from a chaotic signal, *Physical review A, Vol. 28*(4), pp. 2591-2593, **1983.**

[Grassé, 1959] **Grassé, P.-P.**: La reconstruction du nid et les coordinations interindividuelles chezBellicositermes natalensis etCubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs, *Insectes sociaux, Vol. 6*(1), pp. 41-80, **1959.**

[Grosan et al., 2006] **Grosan, C., Abraham, A., & Chis, M.**: Swarm intelligence in data mining, *Swarm Intelligence in Data Mining*, pp. 1-20, Springer, **2006**.

[Haferlach et al., 2010] **Haferlach, T., Kohlmann, A., Wieczorek, L., Basso, G., Te Kronnie, G., Béné, M.-C., . . . Mills, K. I.**: Clinical utility of microarray-based gene expression profiling in the diagnosis and subclassification of leukemia: report from the International Microarray Innovations in Leukemia Study Group, *Journal of Clinical Oncology, Vol. 28*(15), pp. 2529-2537, **2010.**

[Häkkinen/Koikkalainen, 1997] **Häkkinen, E., & Koikkalainen, P.**: SOM based visualization in data analysis, *Artificial Neural Networks—ICANN'97*, pp. 601-606, Springer, **1997**.

[Hamel/Brown, 2011] **Hamel, L., & Brown, C. W.**: Improved interpretability of the unified distance matrix with connected components, Proc. 7th International Conference on Data Mining (DMIN'11), pp. 338-343, **2011**.

[Handl et al., 2006] **Handl, J., Knowles, J., & Dorigo, M.**: Ant-based clustering and topographic mapping, *Artificial Life, Vol. 12*(1), pp. 35-62, **2006.**

[Handl et al., 2005] **Handl, J., Knowles, J., & Kell, D. B.**: Computational cluster validation in post-genomic data analysis, *Bioinformatics, Vol. 21*(15), pp. 3201-3212, **2005.**

[Handl/Meyer, 2007] **Handl, J., & Meyer, B.**: Ant-based and swarm-based clustering, *Swarm Intelligence, Vol. 1*(2), pp. 95-113, **2007.**

[Hartigan, 1981] **Hartigan, J. A.**: Consistency of single linkage for high-density clusters, *Journal of the American Statistical Association, Vol. 76*(374), pp. 388-394, **1981.**

[Hatna/Benenson, 2012] **Hatna, E., & Benenson, I.**: The Schelling model of ethnic residential dynamics: Beyond the integrated-segregated dichotomy of patterns, *Journal of Artificial Societies and Social Simulation, Vol. 15*(1), pp. 6, **2012.**

[Haug/Koch, 2004] **Haug, H., & Koch, S. W.**: *Quantum theory of the optical and electronic properties of semiconductors*, (Edition, F. Ed. Vol. 5), Singapore, World Scientific, ISBN: 13 978-981-283-883-4, **2004**.

[Hausdorf/Hennig, 2010] **Hausdorf, B., & Hennig, C.**: Species delimitation using dominant and codominant multilocus markers, *Systematic Biology, Vol. 59.5*, pp. 491-503, **2010.**

[Havens et al., 2008] **Havens, T. C., Spain, C. J., Salmon, N. G., & Keller, J. M.**: Roach infestation optimization, Proc. Swarm Intelligence Symposium, 2008. SIS 2008. IEEE, pp. 1-7, IEEE, **2008**.

[Haykin, 1994] **Haykin, S.**: *Neural Networks: A comprehensive Foundation*, Uppder Saddle River, NJ, Prentice-Hall, ISBN: 0023527617, **1994**.

[Hennig, 2014] **Hennig, C.**: How many bee species? A case study in determining the number of clusters, *Data Analysis, Machine Learning and Knowledge Discovery*, pp. 41-49, Springer, **2014**.

[Hennig, 2015a] **Hennig, C.**: Clustering strategy and method selection, *arXiv preprint arXiv:1503.02059*, **2015.**

[Hennig, 2015b] **Hennig, C.**: What are the true clusters?, *Pattern Recognition Letters, Vol. 64*, pp. 53-62, **2015.**

[Hennig et al., 2015] **Hennig, C., Meila, M., Murtagh, F., & Rocci, R.**: *Handbook of cluster analysis*, New York, USA, CRC Press, ISBN: 9781466551893, **2015**.

[Herrmann, 2009] **Herrmann, L.** Swarm-Organized Projection for Topographic Mapping, technical report, Philipps-Universität Marburg, Marburg, **2009**.

[Herrmann, 2011] **Herrmann, L.**:*Swarm-Organized Topographic Mapping,* Doctoral dissertation, Philipps-Universität Marburg, Marburg**, 2011**.

[Herrmann/Ultsch, 2008a] **Herrmann, L., & Ultsch, A.**: The architecture of ant-based clustering to improve topographic mapping, *Ant Colony Optimization and Swarm Intelligence*, pp. 379-386, Springer, **2008**.

[Herrmann/Ultsch, 2008b] **Herrmann, L., & Ultsch, A.**: An artificial life approach for semi-supervised Learning, *Data Analysis, Machine Learning and Applications*, pp. 139-146, Springer, **2008**.

[Herrmann/Ultsch, 2008c] **Herrmann, L., & Ultsch, A.**: Explaining Ant-Based Clustering on the basis of Self-Organizing Maps, Proc. ESANN, pp. 215-220, Citeseer, **2008**.

[Herrmann/Ultsch, 2009] **Herrmann, L., & Ultsch, A.**: Clustering with swarm algorithms compared to emergent SOM, Proc. International Workshop on Self-Organizing Maps, pp. 80-88, Springer, **2009**.

[Heskes, 1999] **Heskes, T.**: Energy functions for self-organizing maps, *Kohonen maps, Vol.*, pp. 303-316, **1999.**

[Heston et al., 2012] **Heston, A., Summers, R., & Aten, B.**: Penn World Table Version 7.1 Center for International Comparisons of Production, *Income and Prices at the University of Pennsylvania*, **2012.**

[Hinton/Roweis, 2002] **Hinton, G. E., & Roweis, S. T.**: Stochastic neighbor embedding, Proc. Advances in neural information processing systems, pp. 833-840, **2002**.

[Hinton/Salakhutdinov, 2006] **Hinton, G. E., & Salakhutdinov, R. R.**: Reducing the dimensionality of data with neural networks, *Science, Vol. 313*(5786), pp. 504-507, **2006.**

[Hollander/Wolfe, 1973] **Hollander, M., & Wolfe, D. A.**: *Nonparametric statistical methods*, (Second Edition ed.), New York, USA, John Wiley & Sons, ISBN: 1118553292, **1973**.

[Hotelling, 1933] **Hotelling, H.**: Analysis of a complex of statistical variables into principal components, *Journal of educational psychology, Vol. 24*(6), pp. 417, **1933.**

[Howarth et al., 1996] **Howarth, R. W., Billen, G., Swaney, D., Townsend, A., Jaworski, N., Lajtha, K., . . . Jordan, T.**: Regional nitrogen budgets and riverine N & P fluxes for the drainages to the North Atlantic Ocean: Natural and human influences, *Nitrogen cycling in the North Atlantic Ocean and its watersheds*, pp. 75-139, Springer, **1996**.

[Hunklinger, 2009] **Hunklinger, S.**: *Festkörperphysik*, Oldenbourg Verlag, ISBN: 3486596411, **2009**.

[Hyvärinen, 1997] **Hyvärinen, A.**: Independent Component Analysis by Minimization of Mutual Information*, Helsinki University of Technology, Laboratory of Computer and Information Science,* Finland, Report A 46, **1997.**

[Hyvärinen et al., 2004] **Hyvärinen, A., Karhunen, J., & Oja, E.**: *Independent component analysis*, (Vol. 46), John Wiley & Sons, ISBN: 0471464198, **2004**.

[Jackson, 1999] **Jackson, J. D.**: *Classical Electrodynamics*, (4th edition ed.), New York, John Wiley $ Sons, ISBN: 9783110189704, **1999**.

[Jafar/Sivakumar, 2010] **Jafar, O. M., & Sivakumar, R.**: Ant-based clustering algorithms: A brief survey, *International Journal of Computer Theory and Engineering, Vol. 2*(5), pp. 787, **2010.**

[Jain, 2010] **Jain, A. K.**: Data clustering: 50 years beyond K-means, *Pattern Recognition Letters, Vol. 31*(8), pp. 651-666, **2010.**

[Jain/Dubes, 1988] **Jain, A. K., & Dubes, R. C.**: *Algorithms for Clustering Data*, (Vol. 3), Englewood Cliffs, New Jersey, USA, Prentice Hall College Div, ISBN: 9780130222787, **1988**.

[Jain et al., 1999] **Jain, A. K., Murty, N. M., & Flynn, p. J.**: Data clustering: a review, *ACM computing surveys (CSUR), Vol. 31*(3), pp. 264-323. **1999.**

[Janich/Duncker, 2011] **Janich, P., & Duncker, H.-R.**: *Emergenz-Lückenbüssergottheit für Natur-und Geisteswissenschaften*, F. Steiner, ISBN: 978-3-515-09871-7, **2011**.

[Jardine/Sibson, 1968] **Jardine, N., & Sibson, R.**: The construction of hierarchic and non-hierarchic classifications, *The Computer Journal, Vol. 11*(2), pp. 177-184, **1968.**

[Jennings et al., 1998] **Jennings, N. R., Sycara, K., & Wooldridge, M.**: A roadmap of agent research and development, *Autonomous agents and multi-agent systems, Vol. 1*(1), pp. 7-38, **1998.**

[Jungnickel, 2013] **Jungnickel, D.**: *Graphs, networks and algorithms*, (4th ed ed. Vol. 5), Berlin, Heidelberg, Germany, Springer, ISBN: 978-3-642-32278-5, **2013**.

[Kämpf/Ultsch, 2006] **Kämpf, D., & Ultsch, A.**: An Overview of Artificial Life Approaches for Clustering, *From Data and Information Analysis to Knowledge Engineering*, pp. 486-493, Springer, **2006**.

[Kandel, 2012] **Kandel, E.**: Neurowissenschaften. Eine Einführung, *Padiatrische Praxis, Vol. 79*(4), pp. 672, **2012.**

[Karaboga, 2005] **Karaboga, D.**An idea based on honey bee swarm for numerical optimization,Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, **2005**.

[Karaboga/Akay, 2009] **Karaboga, D., & Akay, B.**: A survey: algorithms simulating bee swarm intelligence, *Artificial Intelligence Review, Vol. 31*(1-4), pp. 61-85, **2009.**

[Karaboga/Ozturk, 2011] **Karaboga, D., & Ozturk, C.**: A novel clustering approach: Artificial Bee Colony (ABC) algorithm, *Applied Soft Computing, Vol. 11*(1), pp. 652-657, **2011.**

[Karatzoglou et al., 2016] **Karatzoglou, A., Smola, A., Hornik, K**.: kernlab: Kernel-Based Machine Learning Lab, R under GPL-2, Version 0.9-25,https://cran.r-project.org/web/packages/kernlab/, **2016**.

[Karbauskaitė/Dzemyda, 2009] **Karbauskaitė, R., & Dzemyda, G.**: Topology preservation measures in the visualization of manifold-type multidimensional data, *Informatica, Vol. 20*(2), pp. 235-254, **2009.**

[Kaski et al., 2003] **Kaski, S., Nikkilä, J., Oja, M., Venna, J., Törönen, P., & Castrén, E.**: Trustworthiness and metrics in visualizing similarity of gene expression, *BMC bioinformatics, Vol. 4*(1), pp. 48, **2003.**

[Kaski et al., 2000] **Kaski, S., Venna, J., & Kohonen, T.**: Coloring that reveals cluster structures in multivariate data, *Australian Journal of Intelligent Information Processing Systems, Vol. 6*(2), pp. 82-88, **2000.**

[Kaufman/Rousseeuw, 1990] **Kaufman, L., & Rousseeuw, p. J.**: Partitioning around medoids (program pam), *Finding groups in data: an introduction to cluster analysis, Vol.*, pp. 68-125, **1990.**

[Kaufman/Rousseeuw, 2005] **Kaufman, L. R., & Rousseeuw, P.**: *Finding groups in data: An introduction to cluster analysis*, Hoboken New York, John Wiley & Sons Inc, ISBN: 0-471-73578-7, **2005**.

[Kaur/Rohil, 2015] **Kaur, P., & Rohil, H.**: Applications of Swarm Intelligence in Data Clustering: A Comprehensive Review, *International Journal of Research in Advent Technology (IJRAT), Vol. 3*(4), pp. 85-95, **2015.**

[Kelso, 1997] **Kelso, J. A. S.**: *Dynamic patterns: The self-organization of brain and behavior*, Cambridge, Massachusetts, London, England, MIT press, ISBN: 0262611317, **1997**.

[Kennedy/Eberhart, 1995] **Kennedy, J., & Eberhart, R.**: *Particle Swarm Optimization*, IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948, IEEE Service Center,, Piscataway, **1995.**

[Khatri/Drăghici, 2005] **Khatri, P., & Drăghici, S.**: Ontological analysis of gene expression data: current tools, limitations, and open problems, *Bioinformatics, Vol. 21*(18), pp. 3587-3595, **2005.**

[Khatri et al., 2012] **Khatri, P., Sirota, M., & Butte, A. J.**: Ten years of pathway analysis: current approaches and outstanding challenges, *PLoS Comput Biol, Vol. 8*(2), pp. e1002375, **2012.**

[Kim, 2006] **Kim, J.**: Emergence: Core ideas and issues, *Synthese, Vol. 151*(3), pp. 547-559. **2006.**

[Kirsch, 1978] **Kirsch, A.**: Bemerkung zu H. Drygas,"Über multidimensionale Skalierung", *Statistical Papers, Vol. 19*(3), pp. 211-212, **1978.**

[Kiviluoto, 1996] **Kiviluoto, K.**: Topology preservation in self-organizing maps, Proc. International Conference on Neural Networks, Vol. 1, pp. 294-299, **1996**.

[Kleinberg, 2003] **Kleinberg, J.**: An impossibility theorem for clustering, *Advances in neural information processing systems, Proc. of the 15th International Conference on Neural Information Processing Systems (NIPS'02)*, pp. 463-470, **2003.**

[Kohlhof, 2010] **Kohlhof, J.**:Untersuchung der Leistungsfähigkeit topologieerhaltender Schwarmalgorithmen zur explorativen Clusteranalyse, (Diplom Diplomarbeit), Phillips-Universität Marburg, Gießen**, 2010**.

[Kohonen, 1982a] **Kohonen, T.**: Analysis of a simple self-organizing process, *Biological cybernetics, Vol. 44*(2), pp. 135-140. **1982.**

[Kohonen, 1982b] **Kohonen, T.**: Self-organized formation of topologically correct feature maps, *Biological cybernetics, Vol. 43*(1), pp. 59-69, **1982.**

[Kohonen, 1995] **Kohonen, T.**: *Self-Organizing Maps*, (Vol. 30), Berlin Heidelberg, Springer, ISBN: 978-3-642-97610-0, **1995**.

[Kohonen/Somervuo, 2002] **Kohonen, T., & Somervuo, P.**: How to make large self-organizing maps for nonvectorial data, *Neural networks, Vol. 15*(8), pp. 945-952, **2002.**

[König, 2000] **König, A.**: Interactive visualization and analysis of hierarchical neural projections for data mining, *Neural Networks, IEEE Transactions on, Vol. 11*(3), pp. 615-624, **2000.**

[König et al., 1994] **König, A., Bulmahn, O., & Glesner, M.**: Systematic Methods for Multivariate Data Visualization and Numerical Assessment of Class Separability and Overlap in Automated Visual Industrial Quality Control, Proc. BMVC, pp. 1-10, **1994**.

[Kraaijveld et al., 1995] **Kraaijveld, M., Mao, J., & Jain, A. K.**: A nonlinear projection method based on Kohonen's topology preserving maps, *Neural Networks, IEEE Transactions on, Vol. 6*(3), pp. 548-559, **1995.**

[Kruskal, 1964a] **Kruskal, J. B.**: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, *Psychometrika, Vol. 29*(1), pp. 1-27, **1964.**

[Kruskal, 1964b] **Kruskal, J. B.**: Nonmetric multidimensional scaling: a numerical method, *Psychometrika, Vol. 29*(2), pp. 115-129, **1964.**

[Kupas et al., 2004] **Kupas, K., Ultsch, A., & Klebe, G.**: Comparison of substructural epitopes in enzyme active sites using self-organizing maps, *Journal of computer-aided molecular design, Vol. 18*(11), pp. 697-708, **2004.**

[Lee et al., 2014] **Lee, J. A., Peluffo-Ordonez, D. H., & Verleysen, M.**: Multiscale stochastic neighbor embedding: Towards parameter-free dimensionality reduction, Proc. Proceedings of 22st European Symposium on Artificial Neural Networks, Computational Intelligence And Machine Learning (ESANN), **2014**.

[Lee/Verleysen, 2007] **Lee, J. A., & Verleysen, M.**: *Nonlinear dimensionality reduction*, New York, USA, Springer, ISBN: 978-0-387-39350-6, **2007**.

[Lee/Verleysen, 2008] **Lee, J. A., & Verleysen, M.**: Rank-based quality assessment of nonlinear dimensionality reduction, Proc. ESANN, pp. 49-54, **2008**.

[Lee/Verleysen, 2009] **Lee, J. A., & Verleysen, M.**: Quality assessment of dimensionality reduction: Rank-based criteria, *Neurocomputing, Vol. 72*(7), pp. 1431-1443, **2009.**

[Lee/Verleysen, 2010] **Lee, J. A., & Verleysen, M.**: Scale-independent quality criteria for dimensionality reduction, *Pattern Recognition Letters, Vol. 31*(14), pp. 2248-2257, **2010.**

[Legg/Hutter, 2007] **Legg, S., & Hutter, M.**: A collection of definitions of intelligence, *Frontiers in Artificial Intelligence and applications, Vol. 157*, pp. 17, **2007.**

[Leister, 2016] **Leister, A. M.**: Hidden Markov models: Estimation theory and economic applications*, Doctoral dissertation, Philipps University Marburg Marburg*, **2016.**

[Li/Xiao, 2008] **Li, J., & Xiao, X.**: Multi-swarm and multi-best particle swarm optimization algorithm, Proc. Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on, pp. 6281-6286, IEEE, **2008**.

[Libbrecht, 2016] **Libbrecht, K. G.**: A monster snow crystal, In c1230c035B1.jpg (Ed.), The Snowflake: Winter's Frozen Artistry, Kenneth Libbrecht and Rachel Wing, **2016**.

[Linde et al., 1980] **Linde, Y., Buzo, A., & Gray, R.**: An algorithm for vector quantizer design, *IEEE Transactions on communications, Vol. 28*(1), pp. 84-95, **1980.**

[Lippmann et al., 2016] **Lippmann, C., Thrun, M. C., & Ultsch, A.:** Overrepresentation Analysis (Version 1.2.5), in preparation. R package, requires CRAN packages: Matrix, ggm, AnnotationDbi, **2016.**

[Lötsch et al., 2013] **Lötsch, J., Doehring, A., Mogil, J. S., Arndt, T., Geisslinger, G., & Ultsch, A.**: Functional genomics of pain in analgesic drug development and therapy, *Pharmacology & therapeutics, Vol. 139*(1), pp. 60-70, **2013.**

[Lötsch/Ultsch, 2013] **Lötsch, J., & Ultsch, A.**: A machine-learned knowledge discovery method for associating complex phenotypes with complex genotypes. Application to pain, *Journal of biomedical informatics, Vol. 46*(5), pp. 921-928, **2013.**

[Lötsch/Ultsch, 2014] **Lötsch, J., & Ultsch, A.**: Exploiting the Structures of the U-Matrix, in Villmann, T., Schleif, F.-M., Kaden, M. & Lange, M. (eds.), Proc. Advances in Self-Organizing Maps and Learning Vector Quantization, pp. 249-257, Springer International Publishing, Mittweida, Germany, **2014**.

[Lueks et al., 2011] **Lueks, W., Mokbel, B., Biehl, M., & Hammer, B.**: How to Evaluate Dimensionality Reduction?-Improving the Co-ranking Matrix, *arXiv preprint arXiv:1110.3917*, **2011.**

[Lumer/Faieta, 1994] **Lumer, E. D., & Faieta, B.**: Diversity and adaptation in populations of clustering ants, Proc. Proceedings of the third international conference on Simulation of adaptive behavior: from animals to animats 3: from animals to animats 3, pp. 501-508, MIT Press, **1994**.

[Lyapunov, 1992] **Lyapunov, A. M.**: The general problem of the stability of motion, *International Journal of Control, Vol. 55*(3), pp. 531-534, doi 10.1080/00207179208934253, **1992.**

[MacQueen, 1967] **MacQueen, J.**: Some methods for classification and analysis of multivariate observations, Proc. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, pp. 281-297, Oakland, CA, USA, **1967**.

[Maechler et al., 2017] **Maechler, M., Rousseeuw, P.** (Fortran original), **Struyf,, A.** (S original), **Hubert, M.** (S original), **Hornik , K.** (port to R; maintenance(1999-2000)), **Studer, M., Roudier, P. & Gonzalez, J**.: cluster: "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al., in R under GPL-3, Version 2.0.6, https://cran.r-project.org/web/packages/cluster/index.html, **2017**.

[Makinen, 2002] **Makinen, G.**: The economic effects of 9/11: A retrospective assessment, Proc., DTIC Document, **2002**.

[Marinakis et al., 2007] **Marinakis, Y., Marinaki, M., & Matsatsinis, N.**: A hybrid clustering algorithm based on honey bees mating optimization and greedy randomized adaptive search procedure, Proc. International Conference on Learning and Intelligent Optimization, pp. 138-152, Springer, **2007**.

[Martens et al., 2011] **Martens, D., Baesens, B., & Fawcett, T.**: Editorial survey: swarm intelligence for data mining, *Machine Learning, Vol. 82*(1), pp. 1-42, **2011.**

[McDonnell, 1995] **McDonnell, R.**: *International GIS dictionary* Retrieved from http://support.esri.com/other-resources/gis-dictionary/term/boundary%20effect, 30.11.2016 11:10, **1995.**

[Merkl/Rauber, 1997] **Merkl, D., & Rauber, A.**: Alternative ways for cluster visualization in self-organizing maps, Proc. Proc. of the Workshop on Self-Organizing Maps (WSOM97), pp. 4-6, Citeseer, **1997**.

[Merskey/Bogduk, 1994] **Merskey, H., & Bogduk, N.**: *Classification of chronic pain, IASP Task Force on Taxonomy*, https://www.iasp-pain.org/Taxonomy, **1994**.

[Miller, 1956] **Miller, G. A**: *The magical number seven, plus or minus two: Some limits on our capacity for processing information*, Psychological Review. Vol. 63 (2), pp. 81–97, PMID 13310704, doi:10.1037/h0043158, **1956.**

[Milligan/Cooper, 1988] **Milligan, G. W., & Cooper, M. C.**: A study of standardization of variables in cluster analysis, *Journal of Classification, Vol. 5*(2), pp. 181-204, **1988.**

[Mirkin, 2005] **Mirkin, B.**: *Clustering: a data recovery approach*, Boca Raton, FL, USA, CRC Press, ISBN: 978-1.58488-534-4, **2005**.

[Mitchison, 1995] **Mitchison, G.**: A type of duality between self-organizing maps and minimal wiring, *Neural Computation, Vol. 7*(1), pp. 25-35, **1995.**

[Mlot et al., 2011] **Mlot, N. J., Tovey, C. A., & Hu, D. L.**: Fire ants self-assemble into waterproof rafts to survive floods, *Proceedings of the National Academy of Sciences, Vol. 108*(19), pp. 7669-7673, **2011.**

[Mokbel et al., 2013] **Mokbel, B., Lueks, W., Gisbrecht, A., & Hammer, B.**: Visualizing the quality of dimensionality reduction, *Neurocomputing, Vol. 112*, pp. 109-123, **2013.**

[Mörchen, 2006] **Mörchen, F.**: *Time series knowledge mining*, Marburg, Germany, Citeseer/Görich & Weiershäuser, ISBN: 3897036703, **2006**.

[Mörchen et al., 2005] **Mörchen, F., Ultsch, A., & Hoos, O.**: Extracting interpretable muscle activation patterns with time series knowledge mining, *International Journal of Knowledge-based and Intelligent Engineering Systems, Vol. 9*(3), pp. 197-208, **2005.**

[Moutarde/Ultsch, 2005] **Moutarde, F., & Ultsch, A.**: U* F clustering: a new performant" cluster-mining" method based on segmentation of Self-Organizing Maps, Proc. Workshop on Self-Organizing Maps (WSOM'2005), **2005**.

[Murphy, 2012] **Murphy, K. P.**: *Machine learning: a probabilistic perspective*, MIT press, ISBN: 0262304325, **2012**.

[Murtagh/Hernández-Pajares, 1995] **Murtagh, F., & Hernández-Pajares, M.**: The Kohonen self-organizing map method: an assessment, *Journal of Classification, Vol. 12*(2), pp. 165-190, **1995.**

[Nash, 1950] **Nash, J. F.**: Equilibrium points in n-person games, *Proc. Nat. Acad. Sci. USA, Vol. 36*(1), pp. 48-49, **1950.**

[Nash, 1951] **Nash, J. F.**: Non-cooperative games, *Annals of mathematics, Vol.*, pp. 286-295, **1951.**

[Natarajan, 2014] **Natarajan, B. K.**: *Machine learning: a theoretical approach*, Morgan Kaufmann, ISBN: 0080510531, **2014**.

[NCBI, 2013] **NCBI, R. C.**: Database resources of the National Center for Biotechnology Information, *Nucleic acids research, Vol. 41*(Database issue), pp. D8, **2013.**

[Neumann/Morgenstern, 1953] **Neumann, L. J., & Morgenstern, O.**: *Theory Of Games And Economic Behavior* (Third Edition ed. Vol. 60), Princeton, USA, Princeton University Press, ISBN, **1953**.

[Ng et al., 2002] **Ng, A. Y., Jordan, M. I., & Weiss, Y.**: On spectral clustering: Analysis and an algorithm, *Advances in neural information processing systems, Vol. 2*, pp. 849-856, **2002.**

[Nisan et al., 2007] **Nisan, N., Roughgarden, T., Tardos, E., & Vazirani, V. V.**: *Algorithmic Game Theory*, (Nisan, N. Ed.), New York, USA, Cambridge University Press, ISBN: 978-0-521-87282-9, **2007**.

[Nöcker et al., 2006] **Nöcker, M., Mörchen, F., & Ultsch, A.**: An algorithm for fast and reliable ESOM learning, Proc. ESANN, pp. 131-136, **2006**.

[Nolting, 2001] **Nolting, W.**: *Grundkurs Theoretische Physik 5/1*, (7th edition ed. Vol. 5), Berlin, Heidelberg, New York, pringer-Verlag ISBN: 3540688686, **2001**.

[Nuzzo, 2014] **Nuzzo, R.**: Statistical errors, *Nature, Vol. 506*(7487), pp. 150-152, **2014.**

[Nybo et al., 2007] **Nybo, K., Venna, J., & Kaski, S.**: The self-organizing map as a visual neighbor retrieval method, Proc. Proc. of the Sixth Int. Workshop on Self-Organizing Maps, **2007**.

[Nybo/ Venna, 2015] **Nybo, K. & Venna J.**: dredviz: dimensionality reduction for information visualization, in ANSI/ISO C++ under LGPL, Version 1.0.2, http://research.cs.aalto.fi/pml/software/dredviz/, Retrieved 15.10.2016 **2015**.

[O'Connor/Wong, 2015] **O'Connor, T., & Wong, H. Y.**: Emergent properties, In Zalta, E. N. (Ed.), *The Stanford Encyclopedia of Philosophy*, Stanford, Kalifornien, Metaphysics Research Lab, Stanford University, **2015**.

[O'Neill/Brabazon, 2008] **O'Neill, M., & Brabazon, A.**: Self-organising swarm (SOSwarm), *Soft Computing, Vol. 12*(11), pp. 1073-1080, **2008.**

[Omar et al., 2013] **Omar, E., Badr, A., & Hegazy, A. E.-F.**: Hybrid AntBased Clustering Algorithm with Cluster Analysis Techniques, Proc. Journal of Computer Science, Vol. 9, pp. 780-793, Citeseer, **2013**.

[Orlowski et al., 2014] **Orlowski, N., Lauer, F., Kraft, P., Frede, H.-G., & Breuer, L.**: Linking spatial patterns of groundwater table dynamics and streamflow generation processes in a small developed catchment, *Water, Vol. 6*(10), pp. 3085-3117, **2014.**

[Ouadfel/Batouche, 2007] **Ouadfel, S., & Batouche, M.**: An Efficient Ant Algorithm for Swarm-Based Image Clustering 1, *Journal of Computer Science, Vol. 3*(3), pp. 2-167, 162, **2007.**

[Parpinelli/Lopes, 2011] **Parpinelli, R. S., & Lopes, H. S.**: New inspirations in swarm intelligence: a survey, *International Journal of Bio-Inspired Computation, Vol. 3*(1), pp. 1-16, **2011.**

[Pasquier, 1987] **Pasquier, V.**: Lattice derivation of modular invariant partition functions on the torus, *Journal of Physics A: Mathematical and General, Vol. 20*(18), pp. L1229, **1987.**

[Passino, 2013] **Passino, K. M.**: Modeling and Cohesiveness Analysis of Midge Swarms, *International Journal of Swarm Intelligence Research (IJSIR), Vol. 4*(4), pp. 1-22, **2013.**

[Patterson/Kelso, 2004] **Patterson, T., & Kelso, N. V.**: Hal Shelton revisited: Designing and producing natural-color maps with satellite land cover data, *Cartographic Perspectives, Vol.* (47), pp. 28-55, **2004.**

[Pearson, 1901] **Pearson, K.**: LIII. On lines and planes of closest fit to systems of points in space, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, Vol. 2*(11), pp. 559-572, **1901.**

[Perneger, 1998] **Perneger, T. V.**: What's wrong with Bonferroni adjustments, *Bmj, Vol. 316*(7139), pp. 1236-1238, **1998.**

[Pham et al., 2007] **Pham, D., Otri, S., Afify, A., Mahmuddin, M., & Al-Jabbouli, H.**: Data clustering using the bees algorithm, Proc. Proceedings of 40th CIRP international manufacturing systems seminar, **2007**.

[Pölzlbauer, 2004] **Pölzlbauer, G.**: *Survey and comparison of quality measures for self-organizing maps*, In Fifth Workshop on Data Analysis (WDA'04), pp. 67-82, **2004**.

[Poundstone, 1992] **Poundstone, W.**: Prisoner's Dilemma: John von Neuman, Game Theory, and the Puzzle of the Bomb (Kindle Edition, Anchor 2011 ed., pp. 294, Doubleday, New York, **1992**.

[R Development Core Team, 2008] **R Development Core Team:** R: A Language and Environment for Statistical Computing (Version 3.2.5), Vienna, Austria, R Foundation for Statistical Computing. Retrieved from http://www.R-project.org, **2008.**

[Rana et al., 2011] **Rana, S., Jasola, S., & Kumar, R.**: A review on particle swarm optimization algorithms and their applications to data clustering, *Artificial Intelligence Review, Vol. 35*(3), pp. 211-222, **2011.**

[Reutterer, 1998] **Reutterer, T.**: Competitive market structure and segmentation analysis with self-organizing feature maps, Proc. Proceedings of the 27th EMAC Conference, pp. 85-115, Citeseer, **1998**.

[Revuelta et al., 2004] **Revuelta, F. F., Chamizo, J. M. G., Rodríguez, J. G., & Sáez, A. H.**: Geodesic topographic product: An improvement to measure topology preservation of self-organizing neural networks, *Advances in Artificial Intelligence–IBERAMIA 2004*, pp. 841-850 Springer, **2004**.

[Reynolds, 1987] **Reynolds, C. W.**: Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH computer graphics, Vol. 21*(4), pp. 25-34, **1987.**

[Ritter, 2014] **Ritter, G.**: *Robust cluster analysis and variable selection*, CRC Press, ISBN: 1439857962, **2014**.

[Ritter et al., 1992] **Ritter, H., Martinetz, T., Schulten, K., Barsky, D., Tesch, M., & Kates, R.**: *Neural computation and self-organizing maps: an introduction*, Addison-Wesley Reading, MA, ISBN: 0201554429, **1992**.

[Rosenwald et al., 2001] **Rosenwald, A., Alizadeh, A. A., Widhopf, G., Simon, R., Davis, R. E., Yu, X., . . . Powell, J.**: Relation of gene expression phenotype to immunoglobulin mutation genotype in B cell chronic lymphocytic leukemia, *Journal of Experimental Medicine, Vol. 194*(11), pp. 1639-1648. **2001.**

[Roweis/Saul, 2000] **Roweis, S. T., & Saul, L. K.**: Nonlinear dimensionality reduction by locally linear embedding, *Science, Vol. 290*(5500), pp. 2323-2326, **2000.**

[Russell et al., 2003] **Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., & Edwards, D. D.**: *Artificial intelligence: a modern approach*, (Vol. 2), New Jersey, USA, Prentice-Hall, Upper Saddle River, ISBN: 0-13-103805-2, **2003**.

[Safavian/Landgrebe, 1990] **Safavian, S. R., & Landgrebe, D.**: A survey of decision tree classifier methodology, *IEEE transactions on systems, man, and cybernetics, Vol. 21(3)*, pp. 660-674, **1990.**

[Şahin, 2004] **Şahin, E.**: Swarm robotics: From sources of inspiration to domains of application, Proc. International workshop on swarm robotics, pp. 10-20, Springer, **2004**.

[Sammon, 1969] **Sammon, J. W.**: A nonlinear mapping for data structure analysis, *IEEE Transactions on computers, Vol. 18*(5), pp. 401-409, doi doi:10.1109/t-c.1969.222678, **1969.**

[Sarlin/Rönnqvist, 2013] **Sarlin, P., & Rönnqvist, S.**: Cluster coloring of the Self-Organizing Map: An information visualization perspective, Proc. Information Visualisation (IV), 17th International Conference, pp. 532-538, IEEE, **2013.**

[Schelling, 1969] **Schelling, T. C.**: Models of segregation, *The American Economic Review, Vol. 59*(2), pp. 488-493, **1969.**

[Schelling, 1971] **Schelling, T. C.**: Dynamic models of segregation†, *Journal of mathematical sociology, Vol. 1*(2), pp. 143-186, **1971.**

[Schiele, 2016] Schiele, J. (2016, 07.09.2016, 06:06 Uh). Irgendwas mit Daten *Frankfurter Allgemeine Zeitung,* p. 2. Retrieved from http://www.faz.net/aktuell/beruf-chance/arbeitswelt/digitalisierung-was-macht-eigentlich-ein-data-scientist-14416564.html, **2016.**

[Schmid, 1980] **Schmid, F.**: Über ein Problem der mehrdimensionalen Skalierung, *Statistical Papers, Vol. 21*(2), pp. 140-144, **1980.**

[Schneirla, 1971] **Schneirla, T.**: *Army ants, a study in social organization*, San Francisco, USA, W.H. Freeman and Company, ISBN: 0-7167-0933-3, **1971**.

[Shepard, 1980] **Shepard, R. N.**: Multidimensional scaling, tree-fitting, and clustering, *Science, Vol. 210*(4468), pp. 390-398, **1980.**

[Siegel/Castellan, 1988] **Siegel, S., & Castellan, N. J.**: Nonparametric statistics for the behavioural sciences, *New York, McGraw-Hill*, **1988.**

[Sokal/Sneath, 1963] **Sokal, R. R., & Sneath, P.**: *Principles of Numerical Taxonomy*, London, Freeman, ISBN, **1963**.

[Sparck Jones, 1972] **Sparck Jones, K.**: A statistical interpretation of term specificity and its application in retrieval, *Journal of documentation, Vol. 28*(1), pp. 11-21, **1972.**

[Steinhaus, 1956] **Steinhaus, H.**: Sur la division des corp materiels en parties, *Bull. Acad. Polon. Sci, Vol. 1*(804), pp. 801, **1956.**

[Stephan, 1999] **Stephan, A.**: *Emergenz von der Unvorhersagbarkeit zur Selbstorganisation*, (First Edition ed.), Germany, mentis, ISBN: 3897854392 or 3933168090, **1999**.

[Stephens/Krebs, 1986] **Stephens, D. W., & Krebs, J. R.**: *Foraging theory*, New Jersey, USA, Princeton University Press, ISBN: 0691084424, **1986**.

[Stöcker et al., 2007] **Stöcker, H., Best, C., Kutz, H., Pitka, R., Griepenkerl, K., Bohrmann, S., . . . Spieles, C.**: *Taschenbuch der Physik*, (5th edition ed.), Frankfurt am Main, Germany, Harri Deutsch, ISBN: 3817117205, **2007**.

[Su et al., 2009] **Su, M.-C., Su, S.-Y., & Zhao, Y.-X.**: A swarm-inspired projection algorithm, *Pattern Recognition, Vol. 42*(11), pp. 2764-2786, **2009.**

[Tan et al., 2006] **Tan, S. C., Ting, K. M., & Teng, S. W.**: Reproducing the results of ant-based clustering without using ants, Proc. 2006 IEEE International Conference on Evolutionary Computation, pp. 1760-1767, IEEE, **2006**.

[Tasdemir/Merenyi, 2009] **Tasdemir, K., & Merenyi, E.**: Exploiting Data Topology in Visualization and Clustering of Self-Organizing Maps, *IEEE Transactions on Neural Networks, Vol. 20*(4), pp. 549-562, doi 10.1109/tnn.2008.2005409, **2009.**

[Tasdemir/Merényi, 2012] **Tasdemir, K., & Merényi, E.**: SOM-based topology visualisation for interactive analysis of high-dimensional large datasets, *Machine Learning Reports, Vol. 1*, pp. 13-15, **2012.**

[Tejada et al., 2003] **Tejada, E., Minghim, R., & Nonato, L. G.**: On improved projection techniques to support visual exploration of multi-dimensional data sets, *Information Visualization, Vol. 2*(4), pp. 218-231, **2003.**

[Tenenbaum et al., 2000] **Tenenbaum, J. B., Silva, V. d., & Langford, J. C.**: A Global Geometric Framework for Nonlinear Dimensionality Reduction, *Science,Vol. 290*(5500), pp. 2319-2323, doi 10.1126/science.290.5500.2319, **2000.**

[Theodoridis/Koutroumbas, 2009] **Theodoridis, S., & Koutroumbas, K.**: *Pattern Recognition*, (Fourth Edition ed.), Canada, Elsevier, ISBN: 978-1-59749-272-0, **2009**.

[Thorel et al., 1990] **Thorel, M.-F., Krichevsky, M., & Lévy-Frébault, V. V.**: Numerical taxonomy of mycobactin-dependent mycobacteria, emended description of Mycobacterium avium, and description of Mycobacterium avium subsp. avium subsp. nov., Mycobacterium avium subsp. paratuberculosis subsp. nov., and Mycobacterium avium subsp. silvaticum subsp. nov, International Journal of Systematic and Evolutionary Microbiology, Vol. 40(3), pp. 254-260, **1990.**

[Thrun/Ultsch, 2017a] **Thrun, M.C., Ultsch, A**.: Projection based Clustering, Conf. Int. Federation of Classification Societies (IFCS), Tokyo, Japan, DOI:10.13140/RG.2.2.13124.53124, **2017.**

[Thrun/Ultsch, 2017b] **Thrun, M. C.:** GeneralizedUmatrix (Version 0.9.5), Marburg. R package, requires CRAN packages: Rcpp, RcppArmadillo, Suggests: matrixStats, rgl, ggplot2, grid, mgcv, Retrieved from https://cran.r-project.org/web/packages/GeneralizedUmatrix/index.html, **2017.**

[Thrun, 2017] **Thrun, M. C.:** DatabionicSwarm (Version 0.9.7), Marburg. R package, requires CRAN packages: Rcpp, RcppArmadillo, deldir, GeneralizedUmatrix, Suggests: plotrix, geometry, sp, spdep, AdaptGauss, ABCanalysis, parallel, matrixStats, rgl, png, Retrieved from https://cran.r-project.org/web/packages/DatabionicSwarm/index.html, **2017.**

[Thrun et al., 2017] **Thrun, M. C., Lerch, F., & Pape, F.:** ProjectionBasedClustering (Version 1.0.4), Marburg. R package, requires CRAN packages: Rcpp, ggplot2, stats, graphics, vegan, deldir, geometry, GeneralizedUmatrix, shiny, shinyjs; Suggests: fastICA, tsne, FastKNN, MASS, pcaPP, spdep, methods, pracma, grid, mgcv, fields, png, reshape2, Retrieved from https://cran.r-project.org/web/packages/ProjectionBasedClustering/index.html, **2017.**

[Thrun et al., 2016a] **Thrun, M. C., Lerch, F., Lötsch, J., & Ultsch, A.**: *Visualization and 3D Printing of Multivariate Data of Biomarkers*, in Skala, V. (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Vol. 24, pp. 7-16, ISBN: 976-80-86943-68-9, Plzen, **2016.**

[Thrun et al., 2016b] **Thrun, M. C., Lerch, F., & Ultsch, A.:** Umatrix (Version 2.0.0), Marburg. R package, requires CRAN packages: Rcpp, ggplot2, shiny, ABCanalysis, shinyjs, reshape2, fields, plyr, abind, tcltk, png, tools, grid, rgl, Retrieved from www.uni-marburg.de/fb12/datenbionik/software-en, **2016.**

[Thrun et al., 2015] **Thrun, M. C., Lötsch, J., & Ultsch, A.:** ABCanalysis (Version 1.1.2), Marburg. R package, requires CRAN packages: plotrix, Retrieved from https://cran.r-project.org/web/packages/ABCanalysis/index.html, **2015.**

[Thrun/Ultsch, 2015] **Thrun, M. C., & Ultsch, A.**: *Models of Income Distributions for Knowledge Discovery*, Proc. European Conference on Data Analysis (ECDA)*,* DOI 10.13140/RG.2.1.4463.0244, pp. 136-137, Colchester, **2015**.

[Timm, 2006] **Timm, I. J.**: Strategic management of autonomous software systems, TZI-Bericht Center for Computing Technologies, University of Bremen, Bremen, **2006.**

[Torgerson, 1952] **Torgerson, W. S.**: Multidimensional scaling: I. Theory and method, *Psychometrika, Vol. 17*(4), pp. 401-419, **1952.**

[Toussaint, 1980] **Toussaint, G. T.**: The relative neighbourhood graph of a finite planar set, *Pattern Recognition, Vol. 12*(4), pp. 261-268, **1980.**

[Tsai et al., 2004] **Tsai, C.-F., Tsai, C.-W., Wu, H.-C., & Yang, T.**: ACODF: a novel data clustering approach for data mining in large databases, *Journal of Systems and Software, Vol. 73*(1), pp. 133-145, **2004.**

[Tukey, 1977] **Tukey, J. W.**: *Exploratory data analysis*, (1 edition ed.), Pearson, ISBN: 978-0201076165, **1977**.

[Tung et al., 2001] **Tung, A. K., Han, J., Lakshmanan, L. V., & Ng, R. T.**: Constraint-based clustering in large databases, Proc. ICDT, Vol. 1, pp. 405-419, Springer, **2001**.

[Uber_Pix, 2015] **Uber_Pix.** A spherical school of fish, 567289_1280_1024.jpg, 1280x1024, https://twitter.com/Uber_Pix/status/614068525766995969, twitter, **2015**.

[Ultsch, 1987] **Ultsch, A.**:*Control for knowledge-based information retrieval,* Doctoral Dissertation, Techn. Wiss. ETH Zürich, Zürich, (8353)**, 1987**.

[Ultsch, 1994] **Ultsch, A.**: The integration of neural networks with symbolic knowledge processing, *New Approaches in Classification and Data Analysis*, pp. 445-454, Springer, **1994**.

[Ultsch, 1995] **Ultsch, A.**: Self organizing neural networks perform different from statistical k-means clustering, Proc. Society for Information and Classification (GFKL), Vol. 1995, Basel 8th-10th March, **1995**.

[Ultsch, 1999] **Ultsch, A.**: Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series, In Oja, E. & Kaski, S. (Eds.), *Kohonen maps*, (1 ed.), pp. 33-46, Elsevier, **1999**.

[Ultsch, 2000a] **Ultsch, A.**: *Clustering with DataBots*, Int. Conf. Advances in Intelligent Systems Theory and Applications (AISTA), pp. p. 99-104, IEEE ACT Section, Canberra, Australia, **2000a.**

[Ultsch, 2000b] **Ultsch, A.**: The Neuronal Data Mine, Proc. Proceedings 2nd Int. ICSC Symposium on Neural Computation NC, Berlin, **2000**.

[Ultsch, 2000c] **Ultsch, A.**: Visualisation and Classification with Artificial Life, *Data Analysis, Classification, and Related Methods*, pp. 229-234, Springer, **2000**.

[Ultsch, 2003a] **Ultsch, A.**: Maps for the visualization of high-dimensional data spaces, Proc. Workshop on Self organizing Maps (WSOM), pp. 225-230, Kyushu, Japan, **2003**.

[Ultsch, 2003b] **Ultsch, A.**Optimal density estimation in data containing clusters of unknown structure, technical report, Vol. 34, University of Marburg, Department of Mathematics and Computer Science, **2003**.

[Ultsch, 2003c] **Ultsch, A.**: *U*-matrix: a tool to visualize clusters in high dimensional data*, Fachbereich Mathematik und Informatik, ISBN, **2003.**

[Ultsch, 2005a] **Ultsch, A.**: Clustering wih SOM: U* C, Proc. Proceedings of the 5th Workshop on Self-Organizing Maps, Vol. 2, pp. 75-82, **2005**.

[Ultsch, 2005b] **Ultsch, A.**: Pareto density estimation: A density estimation for knowledge discovery, In Baier, D. & Werrnecke, K. D. (Eds.), *Innovations in classification, data science, and information systems*, Vol. 27, pp. 91-100, Berlin, Germany, Springer, **2005**.

[Ultsch, 2005c] **Ultsch, A.**: U* C: Self-organized Clustering with Emergent Feature Maps, Proc. Lernen, Wissensentdeckung und Adaptivität (LWA/FGML), pp. 240-244, Saarbruecken, Germany, **2005**.

[Ultsch, 2006] **Ultsch, A.**: Analysis and practical results of U* C clustering, *Advances in data analysis, Proceedings 30th annual conference of the german classification society (GfKl), Berlin, Germany* pp. 6, **2006.**

[Ultsch, 2007] **Ultsch, A.**: Emergence in Self-Organizing Feature Maps, Proc. 6th International Workshop on Self-Organizing Maps (WSOM), ISBN: 978-3-00-022473-7, Bielefeld, Germany, **2007**.

[Ultsch, 2014a] **Ultsch, A.**: "About the Effektsize of pValues in ORA", personal correspondence, 09.04.2014 11:26, **2014**.

[Ultsch, 2014b] **Ultsch, A.**: "about the term frequency-inverse document frequency metrik", personal correspondence, 05.06.2014 16:17, **2014**.

[Ultsch, 2015] **Ultsch, A.**: "About the grid size for the generelized Umatrix", personal correspondence, 07.04.2015, **2015**.

[Ultsch, 2016a] **Ultsch, A.**: "About the Delaunay Classification Error", personal correspondence, 10.10.2016, **2016**.

[Ultsch, 2016b] **Ultsch, A.**: "Discussion about the Application of Muliple Swarms", personal correspondence, 22.11.2016, **2016**.

[Ultsch, 2016c] **Ultsch, A.**: "Lectures: "Introduction into Artificial Intelligence"", personal correspondence, 08.02.2016, **2016**.

[Ultsch et al., 2016a] **Ultsch, A., Behnisch, M., & Lötsch, J.**: ESOM Visualizations for Quality Assessment in Clustering, In Merényi, E., Mendenhall, J. M. & O'Driscoll, p. (Eds.), *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 11th International Workshop WSOM 2016, Houston, Texas, USA, January 6-8, 2016*, doi: 10.1007/978-3-319-28518-4_3, pp. 39-48, Cham, Springer International Publishing, **2016**.

[Ultsch et al., 1994] **Ultsch, A., Guimaraes, G., Korus, D., & Li, H.**: Knowledge extraction from artificial neural networks and applications, *Parallele Datenverarbeitung mit dem Transputer*, pp. 148-162, Springer, **1994**.

[Ultsch/Herrmann, 2005] **Ultsch, A., & Herrmann, L.**: The architecture of emergent self-organizing maps to reduce projection errors, Proc. ESANN, pp. 1-6, **2005**.

[Ultsch/Herrmann, 2010] **Ultsch, A., & Herrmann, L.**: Self Organized Swarms for cluster preserving Projections of high-dimensional Data, *Electronic Communications of the EASST, Vol. 27*, **2010.**

[Ultsch/Korus, 1993] **Ultsch, A., & Korus, D.**: Automatic acquisition of symbolic knowledge from subsymbolic neural networks, Proc. Proceedings of the 3rd European Congress on Intelligent Techniques and Soft Computing, EUFIT, Vol. 3, pp. 326-331, **1993**.

[Ultsch et al., 2016b] **Ultsch, A., Kringel, D., Kalso, E., Mogil, J. S., & Lötsch, J.**: A data science approach to candidate gene selection of pain regarded as a process of learning and neural plasticity, *Pain, Vol. 157(12)*, pp. 2747-2757, **2016.**

[Ultsch/Lötsch, 2014] **Ultsch, A., & Lötsch, J.**: Functional abstraction as a method to discover knowledge in gene ontologies, *PloS one, Vol. 9*(2), pp. e90191. **2014.**

[Ultsch/Lötsch, 2015] **Ultsch, A., & Lötsch, J.**: Computed ABC Analysis for Rational Selection of Most Informative Variables in Multivariate Data, *PloS one, Vol. 10*(6), pp. e0129767. doi 10.1371/journal.pone.0129767, **2015.**

[Ultsch/Lötsch, 2016] **Ultsch, A., & Lötsch, J.**: Machine-learned cluster identification in high-dimensional data, *Journal of biomedical informatics, Vol. 66,* pp. 95-104, **2016.**

[Ultsch/Mörchen, 2005] **Ultsch, A., Moerchen, F.**: ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM, Technical Report Dept. of Mathematics and Computer Science, University of Marburg, Germany, No. 46, **2005**.

[Ultsch/Mörchen, 2006] **Ultsch, A., & Mörchen, F.**: U-maps: topograpic visualization techniques for projections of high dimensional data, Proc. Proc. 29th Annual Conference of the German Classification Society, Citeseer, **2006**.

[Ultsch/Siemon, 1990] **Ultsch, A., & Siemon, H. P.**: *Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis*, International Neural Network Conference, pp. 305-308, Kluwer Academic Press, Paris, France, **1990.**

[Ultsch et al., 2015] **Ultsch, A., Thrun, M. C., Hansen-Goos, O., & Lötsch, J.**: Identification of Molecular Fingerprints in Human Heat Pain Thresholds by Use of an Interactive Mixture Model R Toolbox (AdaptGauss), *International journal of molecular sciences, Vol. 16*(10), pp. 25897-25911, **2015.**

[Ultsch/Vetter, 1995] **Ultsch, A., & Vetter, C.**Self-Organizing-Feature-Maps versus statistical clustering methods: a benchmark,Dept. of Mathematics and Computer Science,, University of Marburg Germany, **1995**.

[Uriarte/Martín, 2005] **Uriarte, E. A., & Martín, F. D.**: Topology preservation in SOM, *International Journal of Mathematical and Computer Sciences, Vol. 1*(1), pp. 19-22, **2005.**

[Van der Maaten/Hinton, 2008] **Van der Maaten, L., & Hinton, G.**: Visualizing Data using t-SNE, *Journal of Machine Learning Research, Vol. 9*(11), pp. 2579-2605, **2008.**

[Van der Maaten et al., 2009] **Van der Maaten, L., Postma, E. O., & Van den Herik, J.**: Dimensionality Reduction: A Comparative Review. from Tilburg Centre for Creative Computing, **2009**.

[Van der Maaten et al., 2009] **van der Maaten, L. J., Postma, E. O., & van den Herik, H. J.**: Dimensionality reduction: A comparative review, *Journal of Machine Learning Research, Vol. 10*(1-41), pp. 66-71, **2009.**

[Van der Merwe/Engelbrecht, 2003] **Van der Merwe, D., & Engelbrecht, A. P.**: Data clustering using particle swarm optimization, Proc. Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, Vol. 1, pp. 215-220, IEEE, **2003**.

[Vardiman et al., 2009] **Vardiman, J. W., Thiele, J., Arber, D. A., Brunning, R. D., Borowitz, M. J., Porwit, A., . . . Tefferi, A.**: The 2008 revision of the World Health Organization (WHO) classification of myeloid neoplasms and acute leukemia: rationale and important changes, *Blood, Vol. 114*(5), pp. 937-951. **2009.**

[Venna/Kaski, 2001] **Venna, J., & Kaski, S.**: Neighborhood preservation in nonlinear projection methods: An experimental study, *Artificial Neural Networks—ICANN 2001*, pp. 485-491, Springer, **2001**.

[Venna/Kaski, 2007] **Venna, J., & Kaski, S.**: Comparison of visualization methods for an atlas of gene expression data sets, *Information Visualization, Vol. 6*(2), pp. 139-154, **2007.**

[Venna et al., 2010] **Venna, J., Peltonen, J., Nybo, K., Aidos, H., & Kaski, S.**: Information retrieval perspective to nonlinear dimensionality reduction for data visualization, *The Journal of Machine Learning Research, Vol. 11*, pp. 451-490, **2010.**

[Verleysen et al., 2003] **Verleysen, M., Francois, D., Simon, G., & Wertz, V.**: On the effects of dimensionality on data analysis with neural networks, *Artificial Neural Nets Problem solving methods*, pp. 105-112, Springer, **2003**.

[Vesanto, 1999] **Vesanto, J.**: SOM-based data visualization methods, *Intelligent data analysis, Vol. 3*(2), pp. 111-126, **1999.**

[Vesanto et al., 1998] **Vesanto, J., Himberg, J., Siponen, M., & Simula, O.**: Enhancing SOM based data visualization, Proc. Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems. Methodologies for the Conception, Design and Application of Soft Computing, Vol. 1, pp. 64-67, **1998**.

[Villmann et al., 1994] **Villmann, T., Der, R., Herrmann, M., & Martinetz, T. M.**: A novel approach to measure the topology preservation of feature maps, *ICANN'94*, pp. 298-301, Springer, **1994**.

[Villmann et al., 1997] **Villmann, T., Der, R., Herrmann, M., & Martinetz, T. M.**: Topology preservation in self-organizing feature maps: exact definition and measurement, *Neural Networks, IEEE Transactions on, Vol. 8*(2), pp. 256-266, **1997.**

[Vinković/Kirman, 2006] **Vinković, D., & Kirman, A.**: A physical analogue of the Schelling model, *Proceedings of the National Academy of Sciences, Vol. 103*(51), pp. 19261-19265, **2006.**

[Walck, 2007] **Walck, C.**Handbook on statistical distributions for experimentalists, Vol. 96-01 SUF-PFY,University of Stockholm, Stockholm, Sweden, **2007**.

[Ward Jr, 1963] **Ward Jr, J. H.**: Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association, Vol. 58*(301), pp. 236-244, **1963.**

[Wilkinson/Friendly, 2009] **Wilkinson, L., & Friendly, M.**: The history of the cluster heat map, *The American Statistician, Vol. 63*(2), pp. 179-184, **2009.**

[Wilkinson/Friendly, 2012] **Wilkinson, L., & Friendly, M.**: The history of the cluster heat map, *The American Statistician, Vol. 63(2)*, pp. 179-184, **2012.**

[Wong et al., 2014] **Wong, K.-C., Peng, C., Li, Y., & Chan, T.-M.**: Herd clustering: A synergistic data clustering approach using collective intelligence, *Applied Soft Computing, Vol. 23*, pp. 61-75, **2014.**

[Xu et al., 2007] **Xu, X., Chen, L., & He, P.**: A novel ant clustering algorithm based on cellular automata, *Web Intelligence and Agent Systems: An International Journal, Vol. 5*(1), pp. 1-14, **2007.**

[Yang, 2009] **Yang, X.-S.**: Firefly algorithms for multimodal optimization, Proc. International Symposium on Stochastic Algorithms, pp. 169-178, Springer, **2009**.

[Yang/He, 2013] **Yang, X.-S., & He, X.**: Bat algorithm: literature review and applications, *International Journal of Bio-Inspired Computation, Vol. 5*(3), pp. 141-149, **2013.**

[Yin, 2007] **Yin, H.**: Nonlinear dimensionality reduction and data visualization: a review, *International Journal of Automation and Computing, Vol. 4*(3), pp. 294-303, **2007.**

[Zak et al., 1999] **Zak, D. R., Holmes, W. E., MacDonald, N. W., & Pregitzer, K. S.**: Soil temperature, matric potential, and the kinetics of microbial respiration and nitrogen mineralization, *Soil Science Society of America Journal, Vol. 63*(3), pp. 575-584, **1999.**

[Zhang et al., 2010] **Zhang, S., Cao, J., Kong, Y. M., & Scheuermann, R. H.**: GO-Bayes: Gene Ontology-based overrepresentation analysis using a Bayesian approach, *Bioinformatics, Vol. 26*(7), pp. 905-911, **2010.**

[Zhong, 2010] **Zhong, Y.**: Advanced intelligence: definition, approach, and progresses, *International Journal of Advanced Intelligence, Vol. 2*(1), pp. 15-23, **2010.**

[Zou et al., 2010] **Zou, W., Zhu, Y., Chen, H., & Sui, X.**: A clustering approach using cooperative artificial bee colony algorithm, *Discrete Dynamics in Nature and Society, Vol. 2010*, Article ID 459796, 16 p., doi: 10.1155/2010/459796, **2010.**

[Zrehen, 1993] **Zrehen, S.**: Analyzing Kohonen maps with geometry, Proc. of the International Conference on Artificial Neural Networks Amsterdam (*ICANN'93)*, pp. 609-612, Springer, London, **1993**.

Appendices

The following section are additions to the various chapters. Supplement A evaluates various QMs on the examples of the Hepta and Chainlink data sets. Supllement B illustatres an high-dimensional example of a bimodal distribution of distances explained in chapter 3 (see Fig. 3.1). Supplement C to D show all visualizations of ESOM, SOP and Pswarm of various data sets introduced in chapter 9. Most importantly it is illustrated that Pswarm does not find any structure if such a structure does not exist in a data set (supplemen D). Supplement G shows additions 3D prints of Pswarm visualizations. Supplement F, H and I complement results of this work with further (mostly statistical) comparisons and testings.

Supplement A: Evaluation of Common QMs

The following section unravels the pitfalls of quality measures based on two different examples: Hepta and Chainlink. They will demonstrate that no quality measure is generalizable because every quality measure (QM) assumes the underlying structure of the data set. If this were not the case the minimizing of a QM would lead to the best possible projection of every data set.
Both data sets are defined by discontinuities: Hepta is a data set with compact structures whereas Chainlink is a data set with connected structures.

First Example: Hepta

For example, three projections methods for the Hepta data set are chosen: PCA, CCA and t-SNE. Overall, four projections are evaluated denoting the two projections of t-SNE with *t-SNE (1)* and *t-SNE (2).* Visually the results are depicted in chapter 5, Figure 5.2, where the seven class labels refer to the colors of the points.
PCA has the highest structure preservation. With default parameters CCA adds gaps of around 3 points. In t-SNE (1) projection the density of the data is overestimated and wide gaps are also added between two points and their cluster, if the default parameter setting is used. By changing one parameter of t-SNE, the t-SNE (2) projection is not able to preserve the structures of data, because many gaps are randomly added.
In Figure A.1 curves of Trustworthiness and Continuity (T&D) are drawn for the four projections of the Hepta data set. The best quality of structure preservation was achieved by PCA (see supplementary), however the curves tend to prefer CCA over PCA. If one plotted only the first 25 k nearest neighbors, t-SNE (1) would reach the best results. Out of the four cases, the T&D is finally able to distinguish the worst case of a low structure preservation of t-SNE (2).
In Table A.1 Topological Index (Spearman's error) and Cpath fail to distinguish the four cases. Topological Correlation (TC) is able to distinguish t-SNE (2) from the other three cases. Cwiring is able to distinguish the four cases, but the difference in values between CCA and PCA is very small. Additionally, without a normalization scheme different data sets would be incomparable. The Classification error with knn=5 is able to rank the PCA projections as the best one and t-SNE (2) as the worst, but prefers t-SNE (1) over the CCA projection.
Calculating AUC in accordance with [Lee et al., 2014] does not yield proper results either because CCA is rated as the best projection by far, and the other three are rated very similar. The RAAR (Figure A.2) curves do not lead to correct interpretations. Zrehen's measure evaluates

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9

t-SNE (1) as a better projection than PCA or CCA, and is only able to depict t-SNE (2) as the worst one.

The precision and recall measures validate that t-SNE minimizes the recall. The measures clearly separate CCA and PCA projections from t-SNE's but cannot distinguish between PCA and CCA projections (see Figure A.3).

On the other hand, the four Shepard Diagrams make it possible to clearly distinguish all four cases. Accordingly, the scatter plot of PCA is distinctly correlated, CCA has some errors on the right corner, t-SNE (1) has problems with density and in t-SNE (2) the distances are randomly distributed. The results of the Shepard Diagram seem to be captured quite well by Kendalls τ (Table A.1).

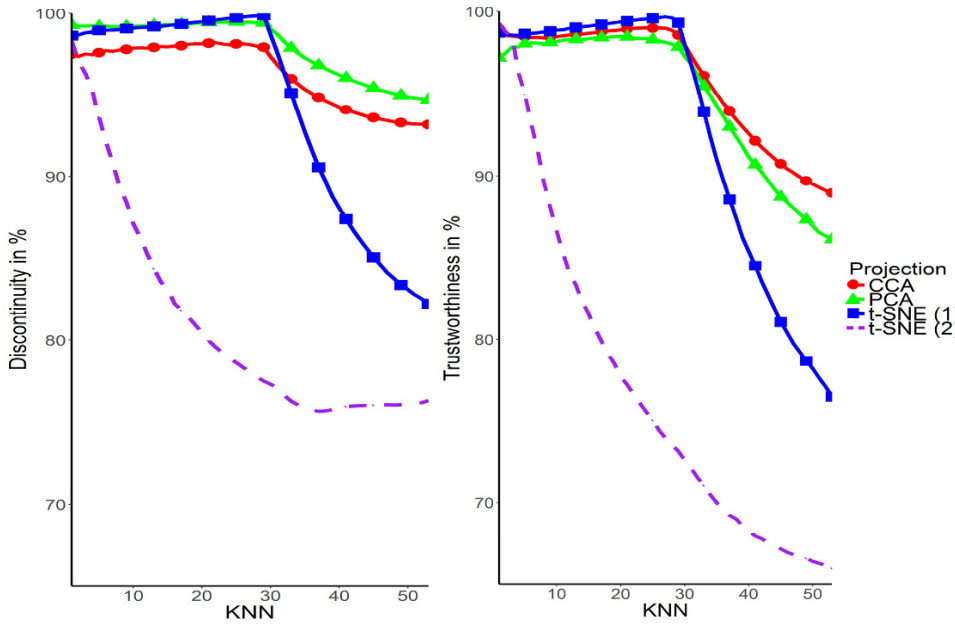

Figure A.1: Trustworthiness and Continuity [Kaski et al., 2003] of the four projections for the first 50 k nearest neighbors. T-SNE (1) instead of PCA has the best values for the first 30 knn, but t-SNE (1) projection does not represent the density of the data set and adds some gaps (see supplementary). From 30 to 50 knn it is unclear if one should prefer CCA or the PCA projection, but CCA disrupts one cluster (see supplementary) by adding additional gaps. The worst projection, t-SNE (2), can be clearly distinguished. The curves do not change their ranks for figures above 50 knn.

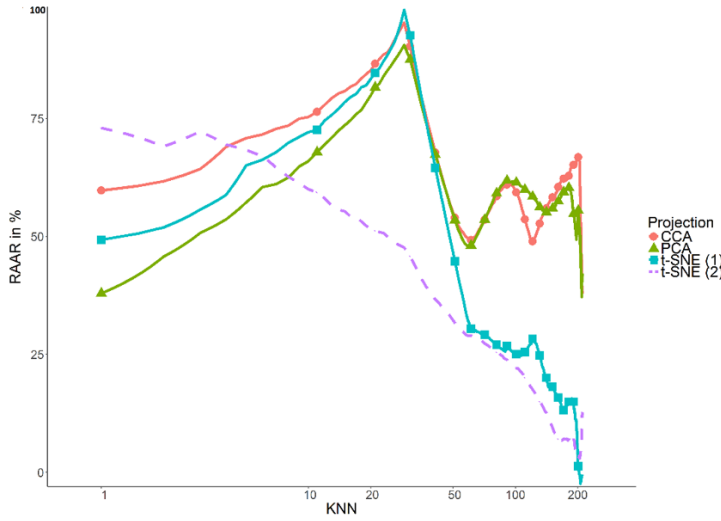

Figure A.2: Rescaled Average Agreement Rate (RAAR) [Lee et al., 2014]. The x-axis is in log scale. CCA is performing slightly better than PCA, and the difference between CCA, PCA and t-SNE to the right of the chart is only visible for knn>50.

Table A.1: Seven quality measures, which produce values of four projections of the Hepta dataset are displayed. The projections are listed in order from best to worst structure preservation. Higher AUC or Correlation values denote better quality of a projection, however for Zrehen and the C values, high values imply a bad quality. TI=Topological Index, TC=Topological Correlation

Projection	**Cpath**	**Cwiring**	**AUC**	**TC**	**TI's ρ**	**Kendall's τ**	**Zrehen**	**Classification-Error**
PCA	52.9	22.9	57.6	0.666	0.808	0.656	4.94	0.0
CCA	28.6	70.5	66.9	0.670	0.809	0.645	4.57	0.014
tSNE (1), „right“	34.2	278	54.6	0.455	0.512	0.365	3.60	0.0047
tSNE (2), „wrong“	38.3	1174	51.9	0.185	0.332	0.233	12.7	0.024

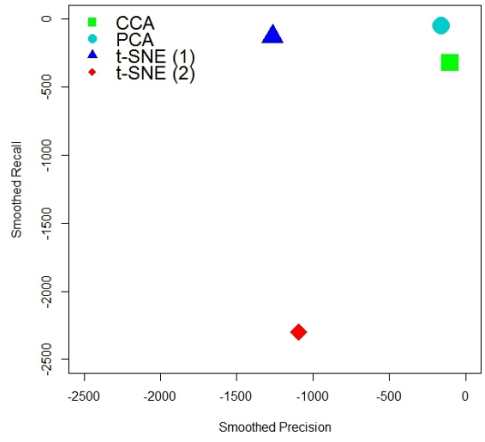

Figure A.3: For the Smoothed Precision and recall of Hepta one could prefer either the CCA or PCA projection. The quality measure shows that t-SNE maximizes the recall. One may also choose the best projection depending on the preference for recall over precision, or vice versa.

Second Example: Chainlink

In this instance the projections of PCA and two different trials of CCA which yield different results are evaluated. The projections are shown in Fig 4. Both CCA projections were computed using the same set of parameters, but the outcome is not deterministic. Instead, the quality of the projection depends on the trial. The PCA projection completely fails to preserve the structures, and the reason is that the PCA only rotates the data set and the discontinuities are not linearly separable. The first CCA (1) projection shows good quality structure preservation but the second CCA (2) projection cuts one cluster in half and projects it in the middle of the second cluster, thus disrupting discontinuities in the input space by letting intruding points in-between. This example illustrates, that for high structure preservation it is sometimes necessary to make higher BPE/FPE errors. A smaller BPE/FPE in CCA (2) does not yield to higher structure preservation, because CCA (2) projections results in additional gaps (Figure A.4).

The evaluation of QMs is restricted to the Sheppard Density Plot with Kendall's $\boldsymbol{\tau}$, the Cwiring measure, precision and recall (Figure A.5), and Trustworthiness and Discontinuity (T&D in Figure A.6) which were the best approaches in the first example. In terms of the CCA and PCA projection of Hepta, the results of precision and recall, as well as of Classification error, were ambiguous. Thus, they are added for the projections of the Chainlink dataset. One could argue that T&D alone cannot distinguish gaps of lower relevance (some points are in the wrong neighborhood) and data density. Hence, results are shown in Fig 6 for the Chainlink data set.

The Sheppard Density Plot and Kendall's $\boldsymbol{\tau}$ are not able to measure structure preservation. This is because the structures of the data sets are not based on compact structures; each ring is closer to some points of the other class than to points of its own class. Cwiring also fails completely. The difference in the T&D measure is very small (<3%). Discontinuity ranks PCA as the best projection, for Trustworthiness CCA (2) ranks highest for the first 50 knn, and thereafter CCA (1). For the PCA projection, recall is clearly much better than for both CCA projections. For the CCA (1) projection, precision is a slightly better than for the CCA (2) projection. However, the best projection may be chosen according to the preference for recall over precision or vice versa.

The classification error is exact zero for both CCA projections. They cannot be distinguished. The PCA projection has a slightly above zero error of 0.3% although the structure preservation is very low.

Table A.2: Cwiring results in three projections of the dataset whereby Chainlink is sorted from the worst to the best structure preservation. The CCA projection is ranked worse than PCA projection. However one CCA projection preserves structures significantly better than the PCA projection. For Kendall's $\boldsymbol{\tau}$ the PCA projection is ranked as the best.

Projection	**Kendalls τ**	**Classification error**	**Cwiring**
PCA	0.792	0.0037	14.3
CCA (2) „wrong“	0.757	0	20.0
CCA (1) „right”	0.748	0	18

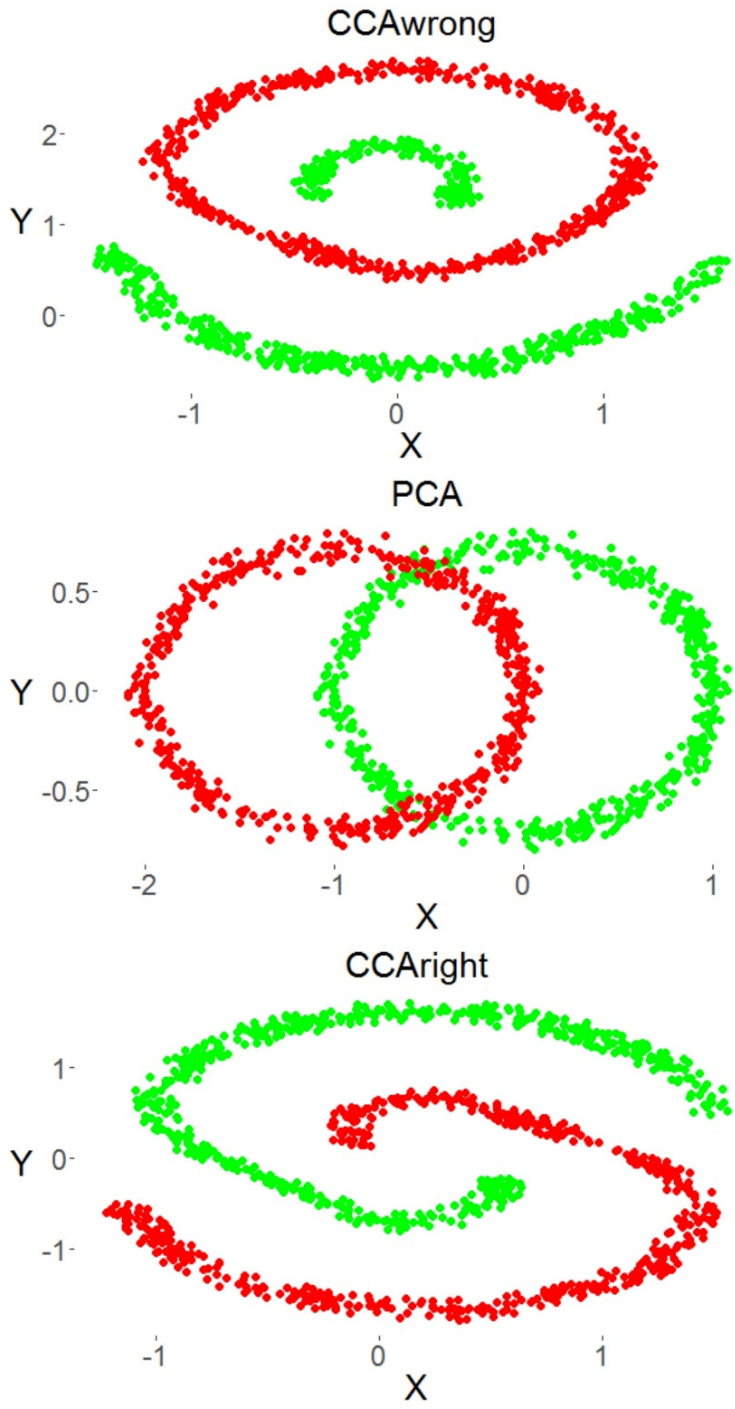

Figure A.4: Chainlink Projection by the PCA and CCA methods. The PCA projection overlaps the clusters, as CCA shows three clearly separated clusters in the first trial (CCA wrong), and preserves the cluster structure in the second trial (CCA correct).

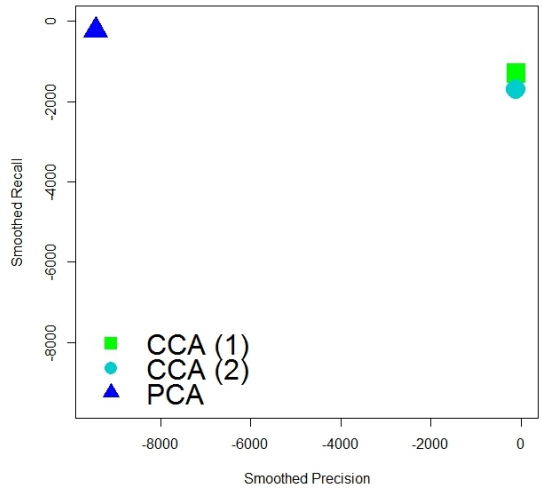

Figure A.5: Smoothed Precision and Recall of Chainlink. It is unclear which projection is structure preserving, but the projections of CCA can be distinguished from each other.

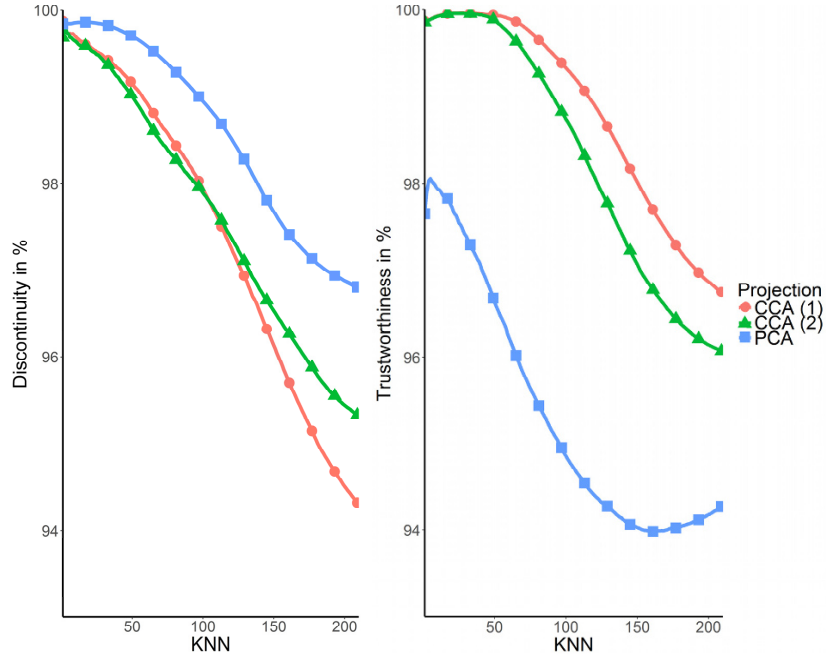

Figure A.6: T&D for the Chainlink data set. For Discontinuity PCA is clearly regarded as the best projection, while the CCA (2) projection is most ideal for Trustworthiness up to the first 50 knn and after that the CCA (1) projection is most suitable. Compared to Figure A.2 of the supplementary, the CCA (1) projection is clearly the best one. Note, that the difference between the three projections is only around 3 percent, but the visual differences in Figure A.2 are clear.

Supplement B: Wine Dataset Distance Distribution

Only Euclidean distances (Figure B.7) were used for SOP, consistent with the settings defined by [Herrmann, 2011, p. 98] and the restrictions of the source code. For Pswarm the squared Euclidean distances were used, because they are slightly more bimodal (Figure B.8) indicating a better distinction between inter and intracluster distances, for further details see chapter 3, Figure 3.1. Distance distributions was generated using the AdaptGauss CRAN package [Thrun/Ultsch, 2015; Ultsch et al., 2015].

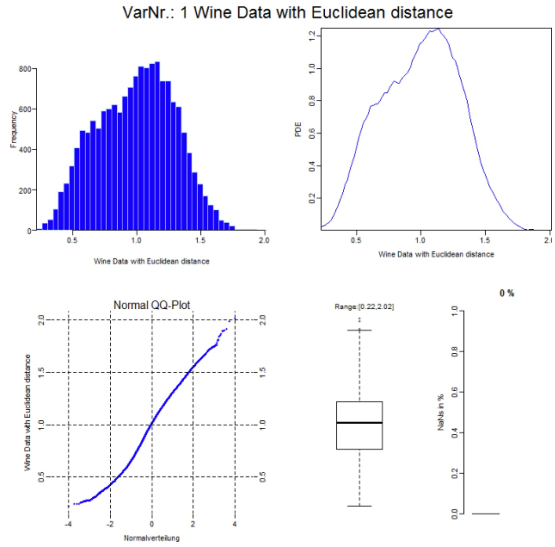

Figure B.7: Distribution of Euclidean distances visualized by histogram, PDEplot, QQplot, Boxplot and the amount of NaNs: The distribution is in the first approximation unimodal.

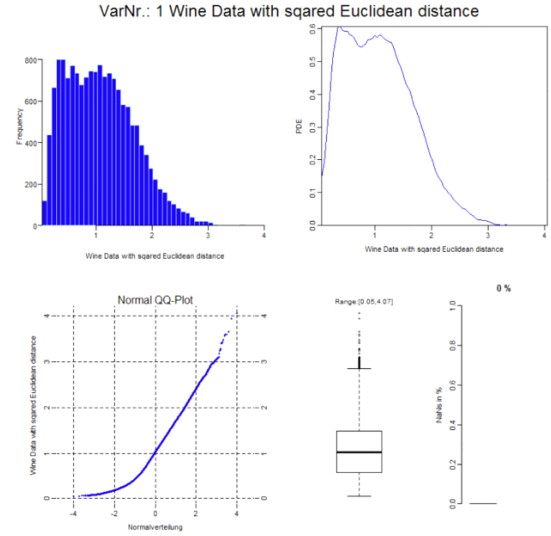

Figure B.8: Distribution of squared Euclidean distances visualized by histogram, PDEplot, QQplot, Boxplot and the amount of NaNs: The distribution is in the first approximation bimodal distinguishing intra- and inter-cluster distances.

Supplement C: Generalized Umatrix of Pswarm and SOP

Supplement C compares the visualizations of DBS through the projection method of Pswarm with the Generalized U-Matrix of SOP for all data sets introduced in chapter 9 which were not shown in this work up until now.

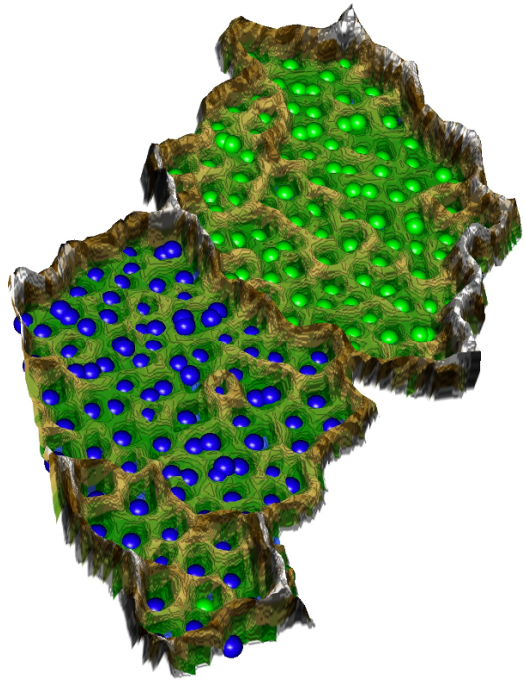

Figure C.9: Topographic map of the Swiss Banknotes data set projected using SOP with the default parameters: The hills of the generalized U-matrix indicate 3 clusters, and one green point is misplaced in the small cluster.

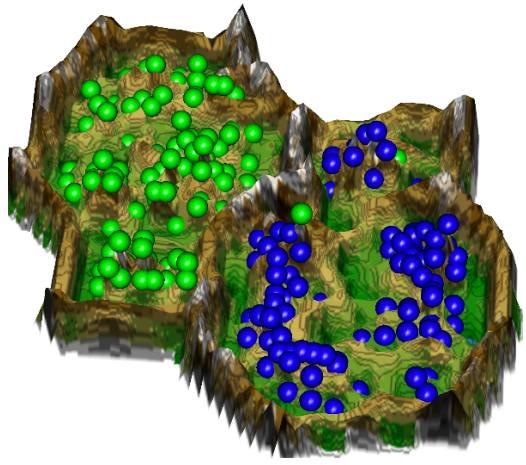

Figure C.10: Topographic map of the Swiss Banknotes data set projected using DBS (36x40) with an automatically chosen lattice size: Two clusters are clearly visible, with two misplaced points. The clustering accuracy of the DBS projection is 99%.

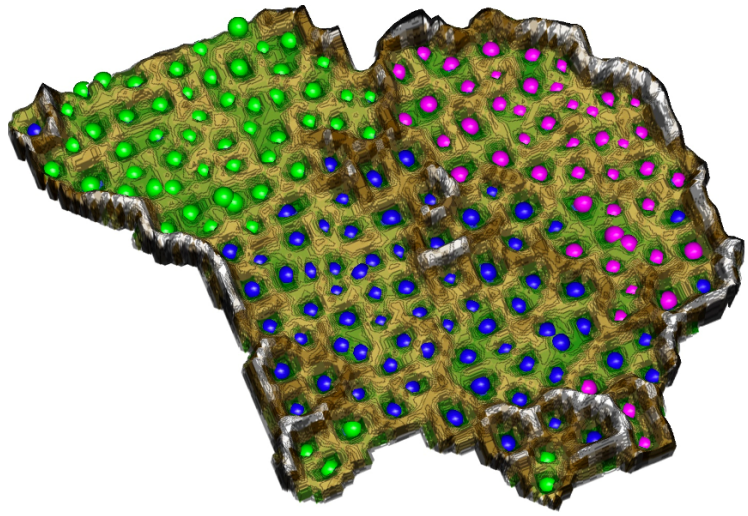

Figure C.11: Topographic map of the Wine data set projected using SOP with the default parameters: The cluster structure is intertwined. Without the colored labels, the clusters could not be identified.

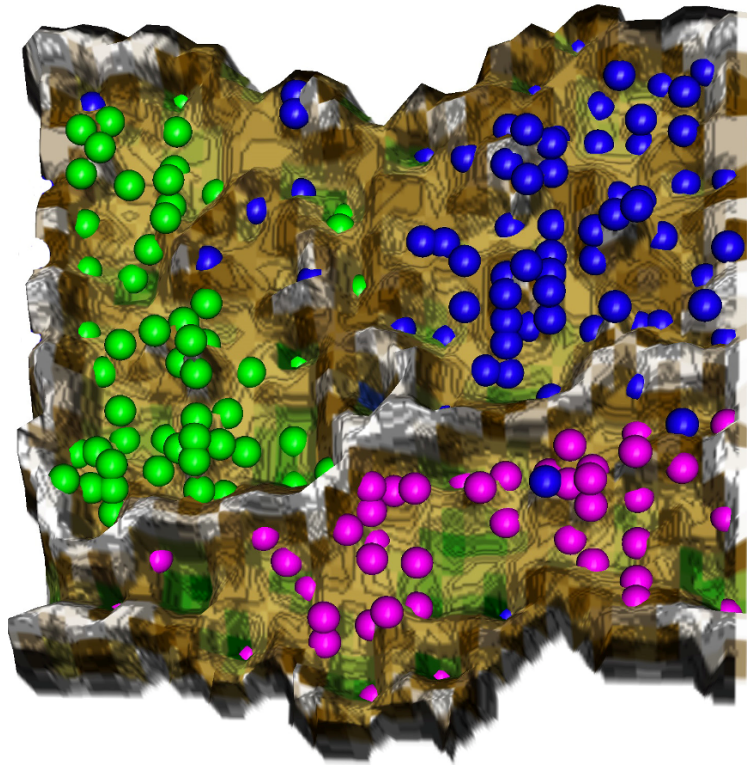

Figure C.12: Topographic map of the Wine data set projected using DBS (28x32) with an automatically chosen lattice size and squared Euclidean distances: The first cluster (green, right) is rectangular in form, the second cluster (blue, left) is square, and the third (pink, bottom) is triangular. The DBS projection yields a clustering accuracy of 92%.

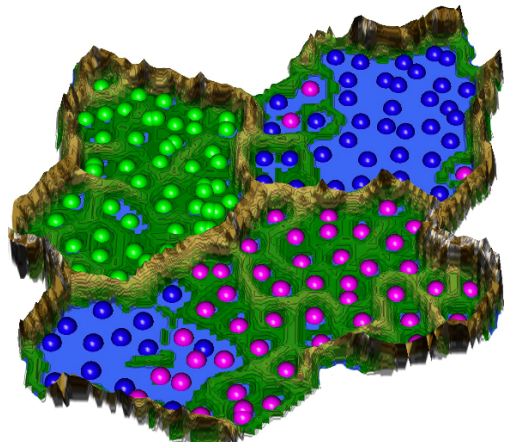

Figure C.13: Topographic map of the Iris data set projected using SOP with the default parameters: One cluster (green) is clearly visible, but the other two clusters (pink and blue) are not correctly reproduced because too many points (11%) are misplaced. The radius of the P-matrix was automatically chosen to be 1.38.

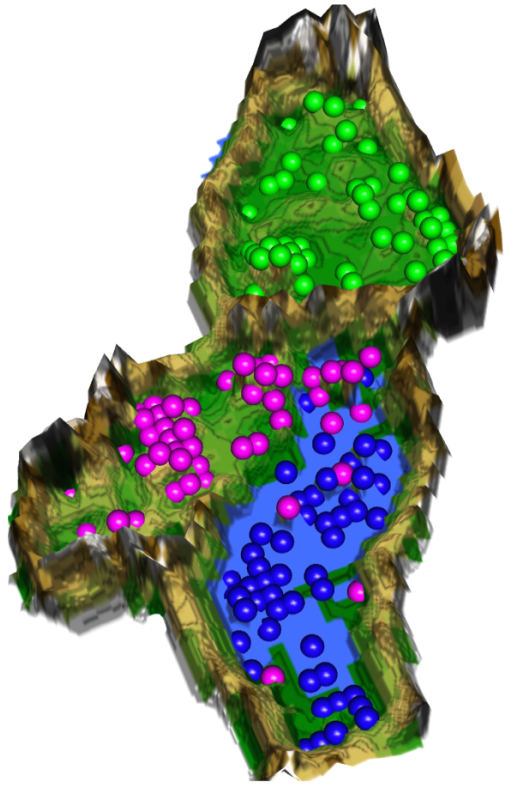

Figure C.14: Topographic map of the Iris data set projected using DBS (26x28) with an automatically chosen lattice size: Three clusters are clearly visible, but with five misplaced points. The points in the first cluster (green) are clearly separated, and the second cluster (blue) has a much higher density than the third cluster (pink). The clustering accuracy of the DBS projection is 99%.

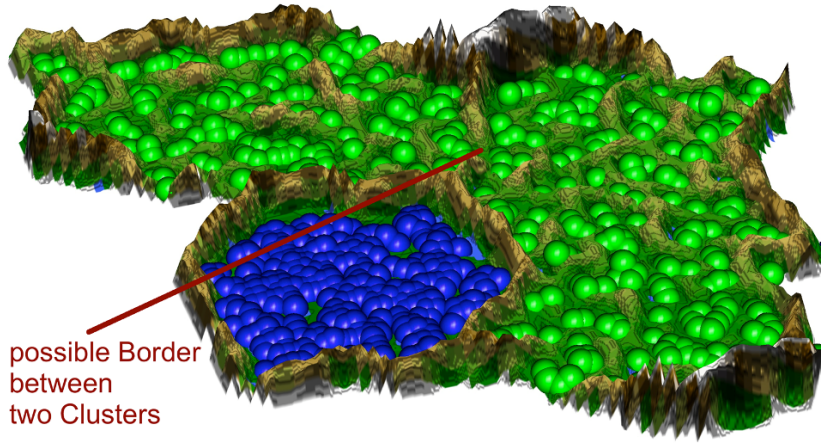

Figure C.15: Topographic map of the Atom data set projected using SOP with the default parameters: The projection shows hills separating parts of the green-labeled cluster. Without the labels corresponding to the prior classification, three clusters would be seen.

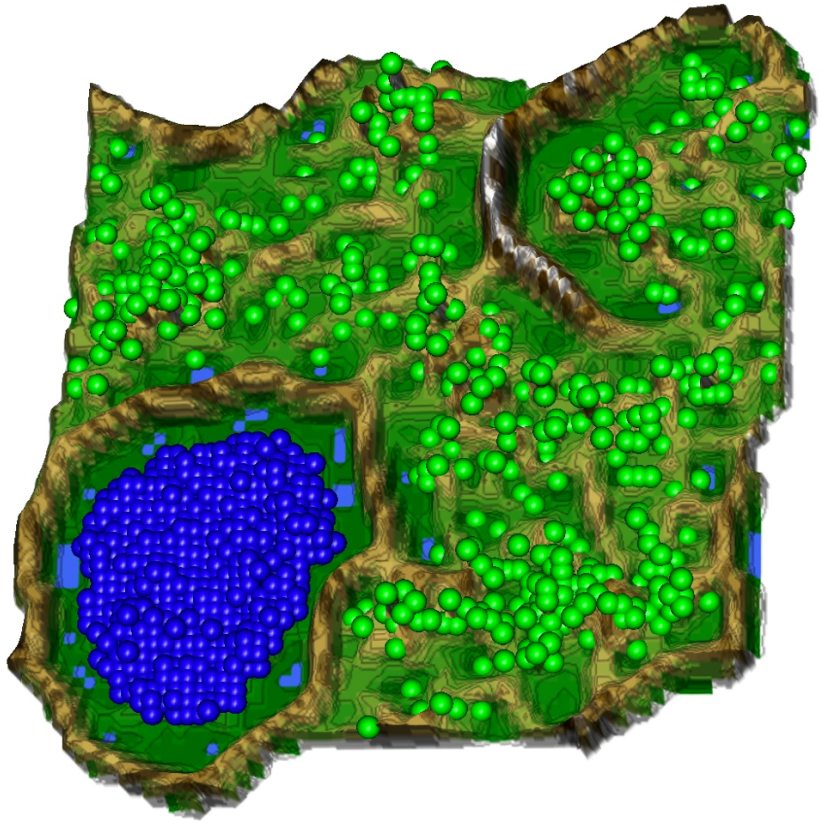

Figure C.16: Topographic map of the Atom data set projected using DBS (58x60) with an automatically chosen lattice size: Two clusters are visible, without any substructures. The clustering accuracy of the DBS projection is 100%.

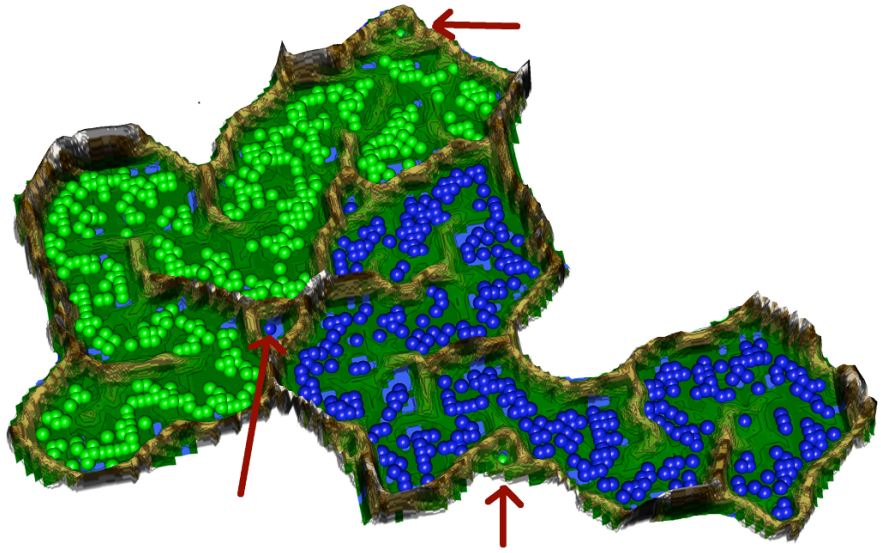

Figure C.17: Topographic map of the Chainlink data set projected using SOP with the default parameters: Two clusters are visible, with two points that could be misinterpreted as outlier points (the green point is shown twice here). The projection is not smooth, as seen from the hilly substructures evident in the clusters.

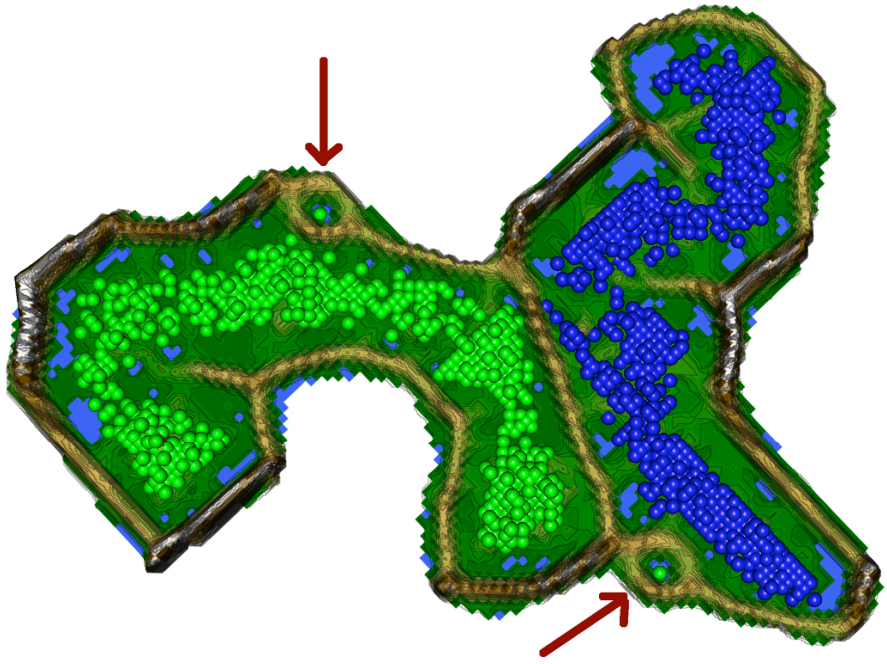

Figure C.18: Topographic map of the Chainlink data set projected using DBS (64x64) with an automatically chosen lattice size: Two clusters are clearly visible, but there is one point that could be misinterpreted as an outlier point (shown twice here). The projection is smoother than that of SOP, as seen from the fact that no hills are visible within the clusters. The clustering accuracy of the DBS projection is 100%.

Supplement D: DBS Visualizations of S-shape and uniform Cuboid

In Figure D.19 it is verified that DBS does not visualize any structures in a data set if the data set does not contain structures.

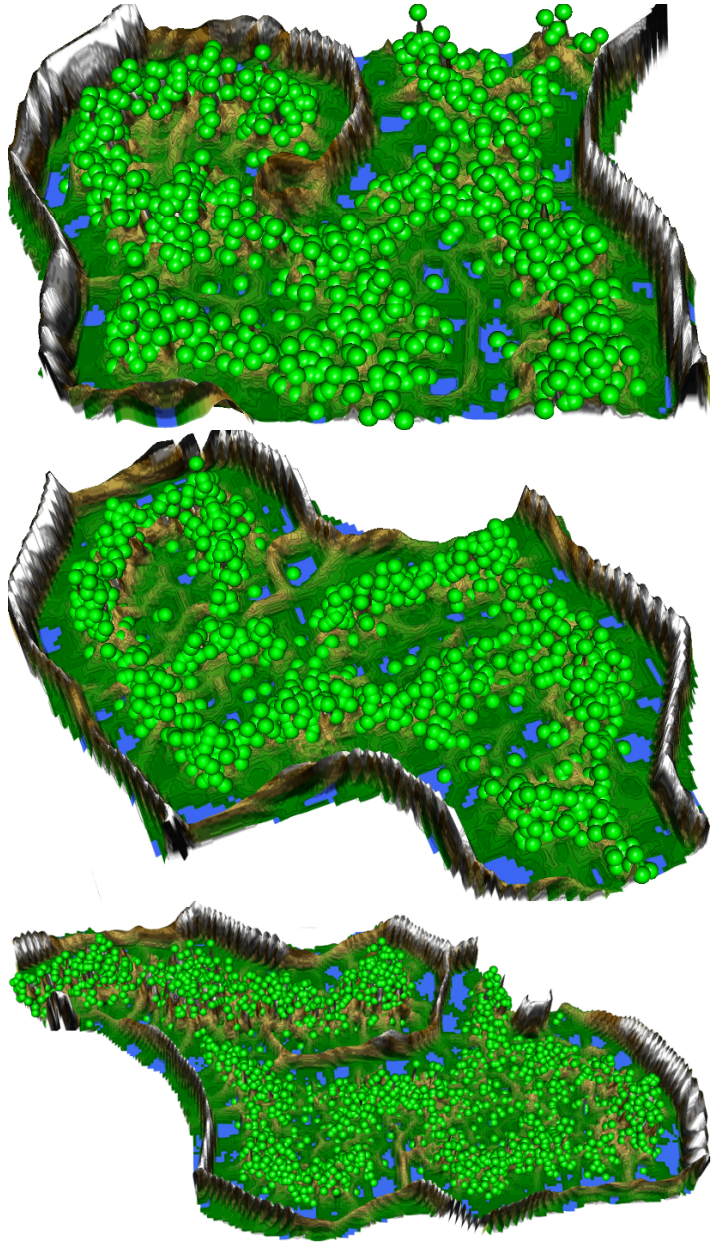

Figure D.19: Topographic maps of three data sets by DBS which do not contain any natural cluster structure. The visualizations show that a cluster structure cannot be seen. Top: cuboid with uniform distributed points; Middle: cuboid with Gaussian distributed points; Down: S-share data sets (see chapter 9 for data set descriptions).

Supplement E: U-Matrix Visualizations of ESOM Projections

All source code was executed in R 3.2.3 [R project, , 2008] on a Windows 7, 64bit system. The ESOM parameterization was chosen for a 50x82 sized toroidal lattice with Gaussian neighborhood function. Further parameterization for the annealing scheme were: 20 epochs, the global neighborhood (learning) radius Rmax=24 and Rmin=1, and the learning rate started at 0.5 and ended at 0.1. The visualization of Fig E.120 E.21, E.22, E.23 are compared in chapter 10.3 to the DBS visualizations.

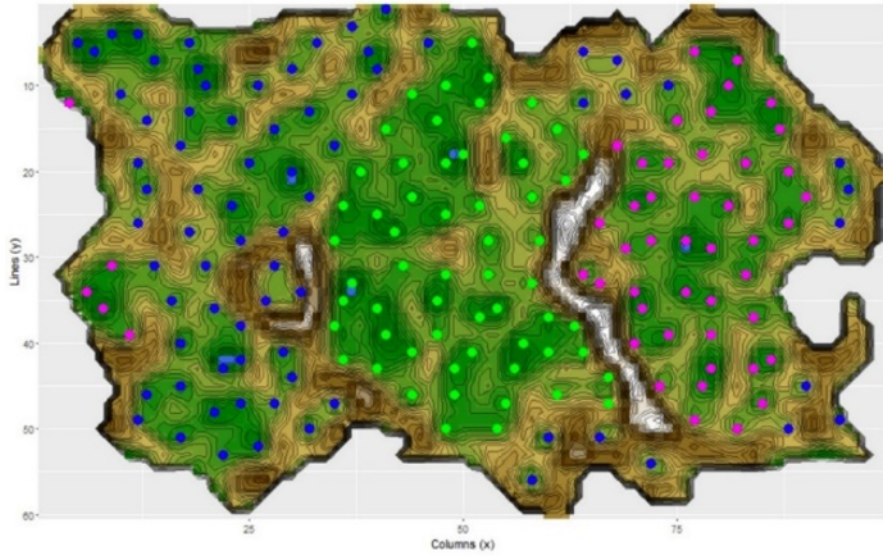

Figure E.20: ESOM projection and U-matrix visualization on Wine data set. The clusters are difficult to separate without the colored labels. Many points are misplaced.

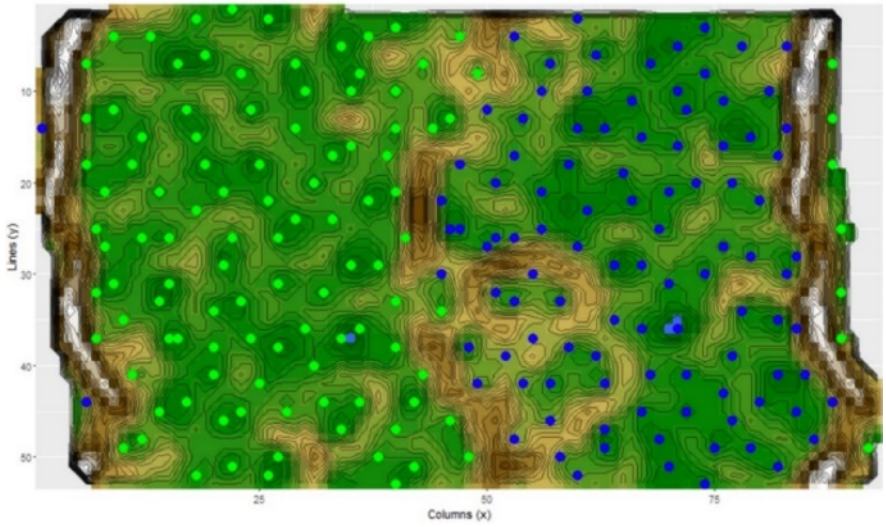

Figure E.21: ESOM projection and U-matrix visualization on Swiss banknotes data set. One best matching unit is misplaced. The cluster with blue best matching units could be interpreted as a small and a big cluster because of the high hills in-between.

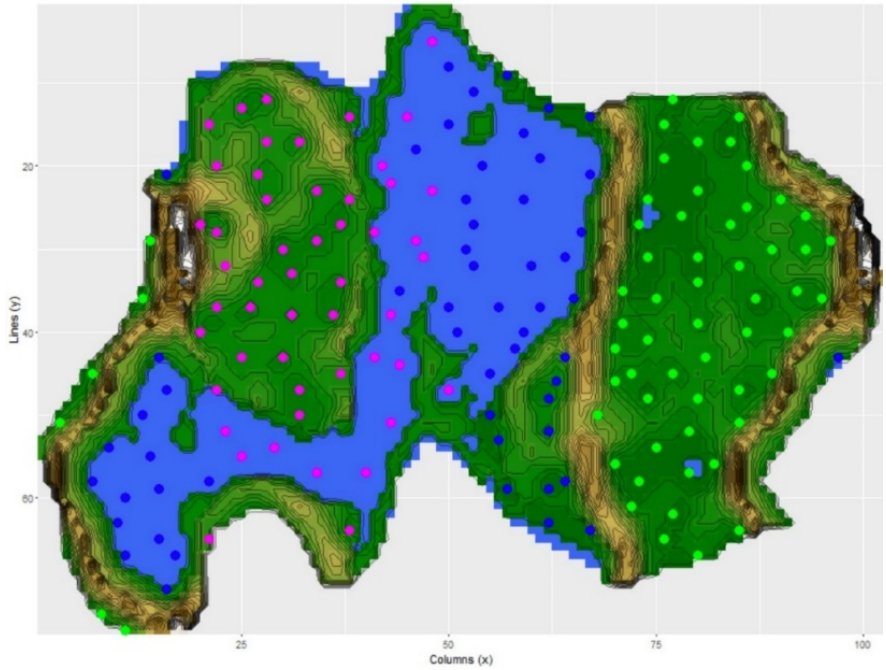

Figure E.22: ESOM projection and U*-matrix visualization of Iris data set. With default parameters the clusters with blue and pink best matching unit cannot be separated.

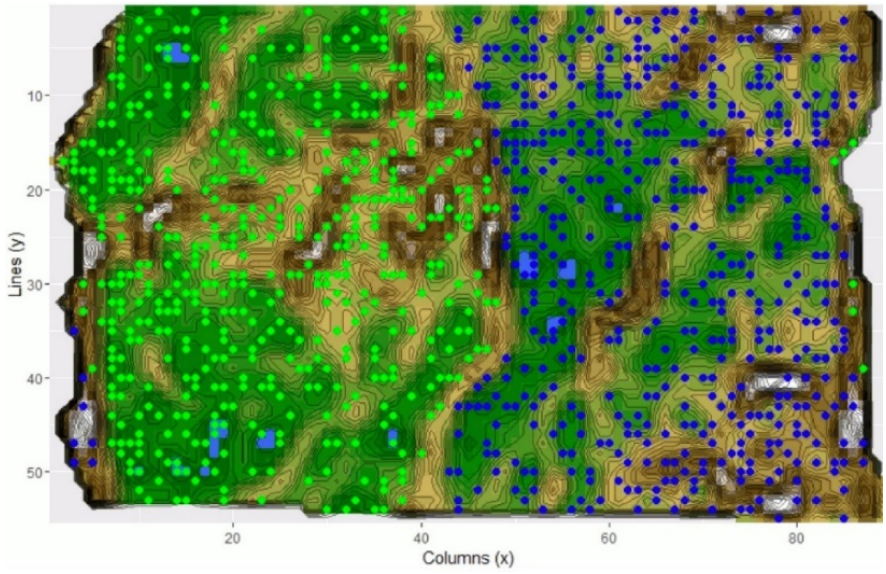

Figure E.23: ESOM projection and U-matrix visualization of Wing Nut data set. If the default parametrization of ESOM is not changed from toroid to planar, the structures of the clusters are very difficult to see.

Supplement F: Statistical Tests in Hydrology

Tab F.3 and F.4 compare the clustering achieved in chapter 12.1 for conductivity and for nitrate. The clusters should contain samples of different natures and based on different processes. Given this assumption, it is valid to statistically test whether the N&C distributions significantly differ between clusters. The Kolmogorov–Smirnov test (KS test) is a nonparametric two-sample test of the null hypothesis that two variables are drawn from the same continuous distribution [Conover, 1971, pp. 309-314. All N&C distributions significantly differ between clusters, with the exception of cluster 4 compared with 5.

Table F.3: KS-test with test statistics *D* and p-value *p* for conductivity. The null hypothesis for cluster 4 and 5 could not be disproved.

Cluster No. (Sample Size)	C1 (223)	C2 (87)	C3 (21)	C4 (7)	C5 (5)
C1(223)		D=0.29, p<0.001	D=0.87, p<0.001	D=1, p<0.001	D=1, p<0.001
C2 (87)	D=0.29, p<0.001		D=0.84, p<0.001	D=1, p<0.001	D=1, p<0.001
C3 (21)	D=0.87, p<0.001	D=0.84, p<0.001		D=1, p<0.001	D=1, p<0.001
C4 (7)	D=1, p<0.001	D=1, p<0.001	D=1, p<0.001		D=0.31, p=0.84

Table F.4: KS-test test with test statistics *D* and p-value *p* for nitrate. The null hypothesis for cluster 4 and 5 could not be disproved.

Cluster No. (Sample Size)	C1 (223)	C2 (87)	C3 (21)	C4 (7)	C5 (5)
C1(223)		D=0.19, p=0.02	D=0.91, p<0.001	D=0.96, p<0.001	D=0.96, p<0.001
C2 (87)	D=0.19, p=0.02		D=0.79, p<0.001	D=0.99, p<0.001	D=0.99, p<0.001
C3 (21)	D=0.91, p<0.001	D=0.79, p<0.001		D=1, p<0.001	D=1, p<0.001
C4 (7)	D=0.96, p<0.001	D=0.99, p<0.001	D=1, p<0.001		D=0.26, p=0.96

Supplement G: 3D Prints of Generalized Umatrix Visualizations of DBS

In Fig. G.1 and G.2 the 3D prints of the visualizations of chapter 12 are shown . [Thrun et al., 2016a].

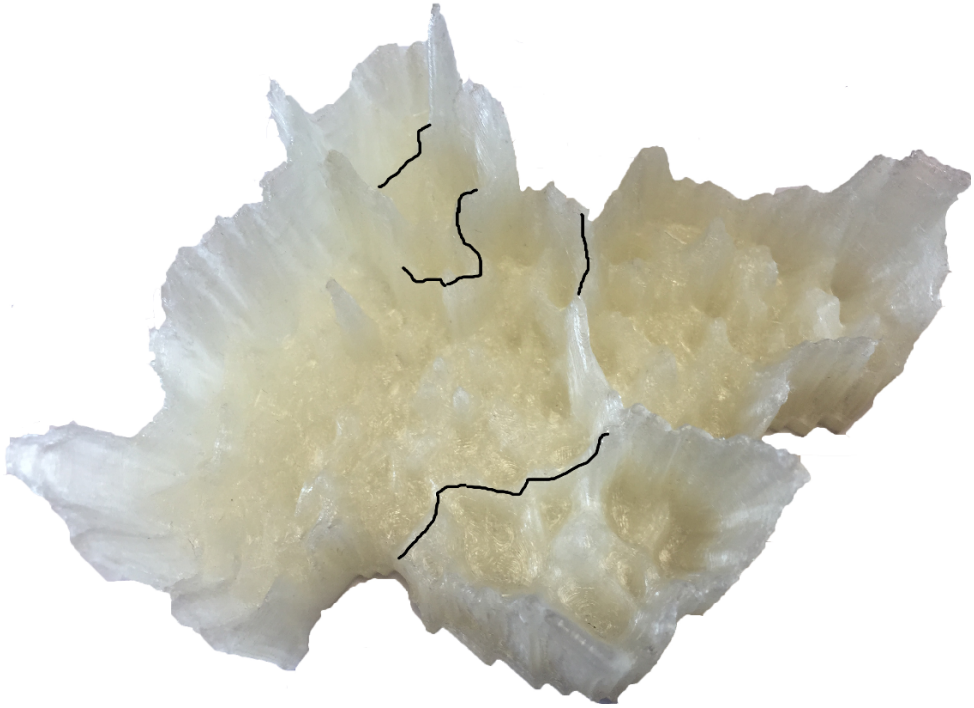

Figure G.24: 3D print of the topographic map of DBS the Hydrology data set of chapter 12, Figure 12.4 (cf. [Thrun et al., 2016a]), colors are not available yet due to technical limitations.

Figure G.25: 3D print of the topographic map DBS of pain genes of chapter 12, Figure 12.9 (cf. [Thrun et al., 2016a]), colors are not available yet due to technical limitations.

Supplement H: Contingency Table for Tetragonula Bees Clustering

Chapter 11.3 introduces the Databionic swarm clustering of the Tetragonula Bees data set and evaluates it with the unsupervised indices of the heatmap and the Silhouette plot. In addition Tab H.5 evaluates the clustering by comparing it to the clustering of [Hennig 2014] by using a contingency table. Besides cluster 6 both clusterings are similar to each other.

Table H.15: DBS clustering in rows versus H2014 ([Hennig 2014]) average linkage clustering in columns. Seven clusters can be reproduced. Total accuracy of DBS clustering in comparison to H2014 is 93%. Abbreviations: R∑ −Rowsum, R% - Rowpercentage, C∑ −Columnsum, C% - Columnpercentage,

H2014/ DBS	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**R∑**	**R%**
1	**63**	0	0	0	0	0	0	0	0	0	63	26,7
2	0	**48**	0	0	0	0	0	0	0	0	48	20,3
3	0	0	**35**	0	0	0	0	0	0	0	35	14,8
4	0	0	0	**23**	**1**	0	0	0	0	0	24	10,2
5	0	0	0	0	**17**	0	0	0	0	0	17	7,2
6	0	**15**	0	0	0	0	0	0	**1**	0	16	6,78
7	0	0	0	0	0	**13**	0	0	0	0	13	5,51
8	0	0	0	0	0	0	**11**	0	0	0	11	4,66
97	0	0	0	0	0	0	0	**4**	0	0	4	1,69
98	0	0	0	0	0	0	0	0	0	**2**	2	0,85
99	0	0	0	0	0	0	0	0	**3**	0	3	1,27
C∑	63	63	35	23	18	13	11	4	4	2	236	0
C%	26,7	26,7	14,8	9,75	7,63	5,51	4,66	1,69	1,69	0,85	0	100

Supplement I: Statistical Tests for FCPS clustering compared to DBS

In Tab I.6 the p-values of the Bonferroni adjusted Wilcoxon rank sum test of the results in chapter 10 Figure 10.1 are presented. If the p-value is lower than 0.05, then DBS outperforms the other clustering method significantly.

Table I.6: Wilcoxon rank sum test for Fig. 10.1 in chapter 10. Abbreviations: single linkage (SL), Linde-Buzo-Gray algorithm (LBG-kMeans), partitioning around medoids (PAM), mixtures-of-Guassians clustering (MoG) also known as model based clustering

DataSet/ Method	**Spectral**	**kMeans**	**PAM**	**Ward**	**SL**	**MoG**
Atom	1	p<0.001	p<0.001	p<0.001	1	p<0.001
Chainlink	1	p<0.001	p<0.001	p<0.001	1	p<0.001
EngyTime	1	1	1	p<0.001	p<0.001	1
Hepta	p<0.001	p<0.001	1	1	1	1
Lsund3D	p<0.001	p<0.001	p<0.001	p<0.001	p<0.001	p<0.001
Target	p<0.001	p<0.001	p<0.001	p<0.001	1	p<0.001
Tetra	1	1	1	1	p<0.001	1
Two Diamonds	p=0.02	1	1	1	p<0.001	1
Wing Nut	1	1	p<0.001	p<0.001	1	1

Index

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9