Swarm-Organized Topographic Mapping

D i s s e r t a t i o n

zur Erlangung des Doktorgrades
der Naturwissenschaften
(Dr. rer. nat.)

dem Fachbereich Mathematik und Informatik
der Philipps-Universität Marburg
vorgelegt

von
Lutz Herrmann
aus Gladenbach

Marburg, 2011

Dem Fachbereich Mathematik und Informatik der Philipps-Universität Marburg als Dissertation vorgelegt von: Diplom-Informatiker Lutz Herrmann, geboren am 01. Dezember 1975 in Marburg/Lahn

Dem Fachbereich Mathematik und Informatik der Philipps-Universität Marburg als Dissertation vorgelegt am: 14. April 2010
Vom Fachbereich Mathematik und Informatik der Philipps-Universität Marburg als Dissertation angenommen am: 19. August 2010

Erstgutachter: Prof. Dr. Alfred Ultsch
Zweitgutachter: Prof. Dr. Eyke Hüllermeier
Tag der mündlichen Prüfung: 01. November 2010

Hochschulnummer: 1180

Zusammenfassung

Die vorliegende Arbeit befasst sich mit visueller, explorativer Datenanalyse auf Basis des Berechenbarkeitsparadigmas der sogenannten Schwarmintelligenz. Der Begriff Schwarmintelligenz bezeichnet die Menge aller Informationsverarbeitungstechniken, die das Verhaltung von sozialen, schwarmbildenden Tieren imitieren. In Analogie zur Natur können dadurch effektive, parallelisierbare und dezentrale Problemlöser realisiert werden, die beispielsweise das Handlungsreisendenproblem mittels simulierter stochastischer Ameisen lösen.

Topographierhaltende Abbildungen konstruieren eine Abbildung von hochdimensionalen oder komplexen Daten auf einen niederdimensionalen Ausgaberaum und versuchen, die Topographie der Daten hinreichend gut zu erhalten. In dieser Arbeit wird ein neuartiger Algorithmus zur topographieerhaltenden Abbildung von vektoriellen und metrischen Daten vorgestellt. Die Schwarm-Organisierte Projektion (SOP) wurde durch verschiedene Techniken aus dem Gebiet der Schwarmintelligenz inspiriert. Das bedeutet, dass das Lernverfahren der SOP Verhaltensmuster von sozialen Tieren imitiert, z.B. Vogelschwärme oder stigmergische Kommunikation von Ameisen.

Die vorliegende Arbeit führt in die Thematik topographieerhaltender Abbildungen ein und behandelt die hervorstechendsten Eigenschaften der wichtigsten Verfahren. Die Qualität topographieerhaltender Abbildungen hängt entscheidend vom gewählten Konzept der Nachbarschaft ab. Fokussierende Verfahren basieren auf einem absinkenden Nachbarschaftsparameter, um globale Eigenschaften der Daten zuerst festzuhalten und lokale Eigenschaften später. Sinkt der Nachbarschaftsparameter zu schnell oder zu langsam ab, führt das in der Regel zu Falschdarstellungen der inhärenten Datenstrukturen.

Die bekanntesten fokussierenden Algorithmen sind die Selbstorganisierende Merkmalskarte (Batch-SOM), sowie die Curvilinear Component Analysis (CCA). Nicht-fokussierende Algorithmen verwenden ein festgefügtes Konzept von Nachbarschaft, das sich während des algorithmischen Lernvorgangs nicht ändert. Auch hier gilt, dass eine falsch gewählte Parametrisierung des Nachbarschaftskonzepts zu Falschdarstellungen der hochdimensionalen Daten führt. Besonders erwähnenswerte nicht-fokussierende Verfahren sind Ant-Based Clustering (ABC) und Stochastic Neighbour Embedding (SNE). Eine formale Analyse des Ant-Based Clustering Verfahrens offenbart eine Verbindung zur Selbstorganisierenden Merkmalskarte. Dies kann anhand von charakteristischen Zielfunktionen in beiden Verfahren gezeigt werden. Die Unfähigkeit von ABC, brauchbare topographieerhaltende Abbildungen zu erzeugen, kann dadurch erklärt werden, dass die korrekte

Version der Zielfunkionen für die Selbstorganisierende Merkmalskarte bekannt ist. Es stellt sich heraus, dass die Zielfunktion im ABC Algorithmus Störterme enthält.

Die Schwarm-Organisierte Projektion (SOP) wurde eingeführt, um das Parametrisierungsproblem fokussierender Algorithmen zu lösen. Dabei soll für eine gegebene Datenmenge ein Absinken des Nachbarschaftsradius automatisch abgeleitet werden. SOP adaptiert das Absinken des Radius selbsttätig, wobei sich das Absinken nach dem Verhalten des Schwarms richtet. Anhand einiger ausgewählter Datensätze kann gezeigt werden, wie sich SOP im Vergleich mit anderen Verfahren verhält, die auf einer fest vorgegebenen Parametrisierung basieren. SOP wird mit Selbstorganisierenden Merkmalskarten (Batch-SOM), Curvilinear Component Analysis, Ant-Based Clustering und Stochastic Neighbour Embedding in mehreren Testreihen verglichen. SOP erzeugt hinreichend topographieerhaltende Abbildungen, während die Vergleichsalgorithmen bei einigen Datensätzen ihre Pathologien offenbaren. Die Ergebnisse von SOP weichen geringfügig von den besten Ergebnissen der Vergleichsverfahren ab, bzw. übertreffen deren Qualität.

Die Verwendbarkeit von SOP wurde durch zwei praktische Anwendungen demonstriert. Die Schwarm-Organisierte Quantisierung (SOQ) ist eine Abwandlung der SOP, die auf vektoriellen Daten arbeitet und eine geringere Rechenkomplexität aufweist. SOQ projiziert Fundamentaldaten börsen-notierter, amerikanischer Unternehmen auf einen niederdimensionalen Ausgaberaum zwecks visueller Analyse der gegebenenfalls vorhandenen Klassen von Unternehmen. Die gefundenen Klassen lassen statistisch signifikante Rückschlüsse auf den Börsenverlauf der betreffenden Aktien zu.

Die zweite Anwendung beschäftigt sich mit microRNAs, einem neuen Forschungsgebiet der molekularen Biologie. MicroRNAs sind kleine Fragmente von RNA, die Gene regulieren. Die Menge der von microRNAs regulierten Gene wurde mit SOP untersucht und in homogene Klassen gleichartiger Gene segmentiert. Damit soll die Frage beantwortet werden, welche Arten von Genen überhaupt durch microRNAs reguliert werden. Die so erzeugten Genklassen wurden auf ihre Signifikanz hin mittels einer externen Validierungsmethode hin untersucht, der sogenannten Overrepresentation Analysis. Die mittels SOP erzeugten Klassen sind signifikanter als die mit Ward's Linkage und k-Means hergeleiteten Genklassen. Ein externer Experte konnte zudem den Nutzen und die Neuartigkeit des so gewonnenen Wissens bestätigen.

Die Seite 5 enthält persönliche Daten. Sie ist deshalb nicht Bestandteil der Online-Veröffentlichung.

Erklärung

Ich versichere, dass ich meine Dissertation

"Swarm-Organized Topographic Mapping"

selbständig, ohne unerlaubte Hilfe angefertigt und mich dabei keiner anderen als der von mir ausdrücklich bezeichneten Quellen und Hilfen bedient habe. Die Dissertation wurde in der jetzigen oder einer ähnlichen Form noch bei keiner anderen Hochschule eingereicht und hat noch keinen sonstigen Prüfungszwecken gedient.

Contents

Acknowledgements

First of all I'd like to thank my wife Anja for her loving support and encouragement. I would like to express my sincere gratitude to my advisor Prof. Alfred Ultsch for the continuous support of my research, for his guidance and immense knowledge. His ideals and philosophical concepts helped me in all the time of writing this thesis. Besides my advisor, I would like to thank my colleague Florian Meyer for lots of fruitful discussions. Last but not least, praise and thanks goes to my savior Jesus Christ.

Chapter 1

Introduction

1.1 Objective

In the last decades, the information technology has produced a rapidly increasing amount of data, as well as more complex types of data. A particularly interesting and valuable field for research is "Knowledge discovery in databases", i.e. the automated retrieval of interesting patterns in large or complex data sets. Companies want to learn more about their customers to better target advertisement or to prevent fraud. The financial sector wants to gain insight into the developments of the stock market. Scientists from many domains search for interesting patterns in data observed in experiments, i.e. generating hypotheses about underlying processes. In medicine, genetic and other data from patients is searched to find the cause of diseases and develop treatments. Complex data structures arise in many life sciences, e.g. protein data for pharmaceutical drug design or the understanding of biochemical systems. Such data collections increase the understanding of molecules' behaviour and enable the generation of novel research hypotheses in molecular biology for instance.

All these information technologies have led to vast amounts of non-trivial data whose precise and reliable analysis requires a considerable amount of human intervention and expertise and, therefore, leads to a cost factor of substantial economic relevance. From a formal computer science perspective, the question arises how knowledge might be extracted from data. A central aspect is to unveil hidden regularities in given complex or high-dimensional data. Thus, raw data is hopefully transformed into representations conceivable to the human mind. Several terms relating to this challenge have been coined in the last decades, such as pattern recognition, data mining, knowledge discovery, or exploratory data analysis.

For effective knowledge discovery, it is important to include the human in the data exploration process and combine the flexibility, creativity and general knowledge of the human with the arithmetic power of today's computers. Thus, visual data exploration is a promising approach to the knowledge discovery challenge. The basic principle of visual data exploration is to present the data in some visual form, allowing the human expert to get insight into the data, draw conclusions and come up with novel hypotheses about the available data. Visual data exploration

is especially useful when little is known about the data in beforehand, and the exploration goals are vague.

A particularly interesting approach to visual data exploration is called *topographic mapping*, i.e. the creation of low-dimensional images from high-dimensional or complex data. Topographic mapping methods typically depict data as low-dimensional point clouds. Dense piles of low-dimensional points supposedly refer to hidden regularities in the original raw data. Topographic mapping algorithms may be realized by using principles from statistics, artificial neural networks and swarm intelligence. Such projections are, for example, Principal Component Analysis, Sammon's Mapping, Self-Organizing Maps and Curvilinear Component Analysis. These methods aim to preserve the proximity structure of the data, such that similar data objects are visualized as nearby low-dimensional points. Dissimilar data objects are supposed to become faraway points in the low-dimensional output space.

Most topographic mapping algorithms rely on a pre-defined concept that indicates which proximity relations are to be preserved in the output space, and which are not. Such a concept might be given by means of neighbourhood graphs, neighbourhood radiuses or geometrical relations. For a given data set, a matching concept for preservation of important structures is, however, an unknown quantity in beforehand. Thus, mismatching parametrization of topographic mapping algorithms leads to misrepresentations of the patterns hidden in the raw data. In this paper we present a novel approach for topographic mapping, which adapts itself to the topographical structures of a data set. It is based on swarm intelligence, i.e. the algorithm is guided by the flocking behaviour of numerous independent but cooperating agents in a swarm. No critical parametrization for the construction of meaningful topographic mappings is necessary.

1.2 Contributions

The main objective of this work is to provide a convenient answer to the question how the swarm intelligence paradigm may contribute to unsupervised visual data exploration. The following aspects are considered in this thesis.

Unifying Framework

Swarm intelligence has been successfully applied for supervised search and optimization tasks, e.g. Ant Colony Optimization meta heuristics [Dor92] for traveling salesman problems. However, swarm intelligence methods have shown weak performances when applied to unsupervised data analysis and clustering tasks. For example, the widely adapted Ant-Based Clustering meta heuristics [DAGP89] [LF94] was found to be not competitive in comparison with other techniques [HKD05]. This thesis offers an explanation for the weak performance of Ant-Based Clustering algorithms. A unifying framework for several dimension reduction methods is derived. This framework allows to explain the behaviour of Ant-Based Clustering by means of other well-established algorithms. The connection between swarm intelligence and artificial neural networks is specified. Due to this, naive but effective improvements of swarm intelligence methods were derived.

Swarm-driven Guidance

A promising approach for unsupervised analysis of large and high-dimensional data is visualization, i.e. the creation of a low-dimensional image that reflects the data-inherent structures of interest. This is referred to as topographic mapping. Most visual data inspection methods depend on a crucial, pre-defined parametrization that determines which properties of the given data are used for creation of low-dimensional images, and which are not. The self-adjustment of such parameters is a main aspect of this work. We introduce heuristics that adjusts its focus on proximities by means of swarm entities' behaviour, instead of pre-defined and mismatching proximity concepts.

Furthermore, the behaviour of swarm entities indicates which regions in search space are more promising and which are not. A novel method is proposed which is based on swarm-driven search routines, in order to significantly decrease the computational complexity when creating low-dimensional proximity images.

Evaluation

In the field of machine learning, the benefits of novel techniques are usually demonstrated by means of test series and statistical evaluation of results. Appropriate statistical tests are selected in order to assess the distributions of obtained performance measures. This indicates whether the proposed topographic mapping method leads to superior projection quality than conventional methods, or not. In this work, the usefulness of arbitrary topographic mappings for visualization with U-Matrix methods is investigated by means of geometry. A novel geometric approach for quality assessment of topographic mappings will be derived from these insights.

Real-World Applications

The usefulness of the proposed methods is demonstrated on real-world data. An application from molecular biology illustrates how the use of topographic mappings leads to the discovery of novel knowledge concerning the cellular regulation mechanisms in genetics, i.e. which types of genes are actually regulated by micro RNA. Another application illustrates the use of our proposed method on the field of financial fundamental analysis.

1.3 Outline of the Thesis

This doctoral thesis is structured as follows. Chapter 2 gives a brief insight into the broad field of knowledge discovery with machine learning algorithms. Furthermore, the need for robust visual data inspection methods is outlined. Chapter 3 gives an overview on the most important topographic mapping algorithms. Strengths and weaknesses of these approaches are briefly illustrated. The main focus is on nature-inspired methods like Self-Organizing Maps (SOM) and Ant-Based Clustering (ABC). Additionally, popular conventional statistical methods are described. Chapter 4 introduces the Swarm-Organized Projection method, a novel topographic

mapping algorithm based on the swarm intelligence paradigm. Furthermore, the Swarm-Organized Quantization is proposed as a fast method for vectorial spaces. Chapter 5 analyzes and assesses the relevant topographic mapping algorithms. The behaviour of Ant-Based Clustering finally becomes explainable. Chapter 6 evaluates the abilities of Swarm-Organized Projection by means of cardinal cluster problems. In Chapter 7 the proposed methods are applied on biological data that is concerned with regulation of cellular activity of genes by micro RNA. Swarm-Organized Projection retrieves several meaningful classes of functionally similar genes regulated by micro RNA. Chapter 8 contains an application from a real-valued domain for the proposed methods. Fundamental analysis of stock market data is performed by means of Swarm-Organized Quantization. Finally, the results are discussed and summarized in Chapters 9 and 10.

Chapter 2

Basic Principles

In this chapter, all topics that are essential for the understanding of this thesis are presented in a brief yet formal manner.

2.1 Data

Vectorial Data

In the context of this thesis, it is assumed that an unknown stochastic process is underlying a certain domain, e.g. stock market prices or genetic expressions. The data $\mathbb{X} = \{x_1, ..., x_n\} \subset \mathbb{D}$ is a finite set of real-valued, D-dimensional observations of this process, with $D \in \mathbb{N}$. The set of all possible observations is called data space $\mathbb{D} \subset \mathbb{R}^D$ with the following operators:

- vector addition $+ : \mathbb{D} \times \mathbb{D} \to \mathbb{D}$
- scalar multiplication $\cdot : \mathbb{R} \times \mathbb{D} \to \mathbb{D}$
- norm $\| \cdot \| : \mathbb{D} \to \mathbb{R}_0^+$

Each element $x_i \in \mathbb{X}$ is called a data vector and contains observations from $D \in \mathbb{N}$ (random) variables. The data $\mathbb{X}$ is organized as matrix at which rows correspond to data vectors, and columns contain variables. See Figure 2.1 for illustration. Usually the Euclidean norm $\|.\|$ is used, e.g. for induction of distances:

$$\|z\| = \sqrt{\sum_{i=1}^{D} z(i)^2}$$

It is assumed that the data is not equally distributed in data space, but assembles in or near lower-dimensional manifolds embedded in data space. A manifold is a mathematical subspace that on a small scale (locally) resembles the Euclidean space of a specific dimension, called the dimension of the manifold. However the global structure of a manifold may be more complicated, e.g. the surface of a sphere or a torus.

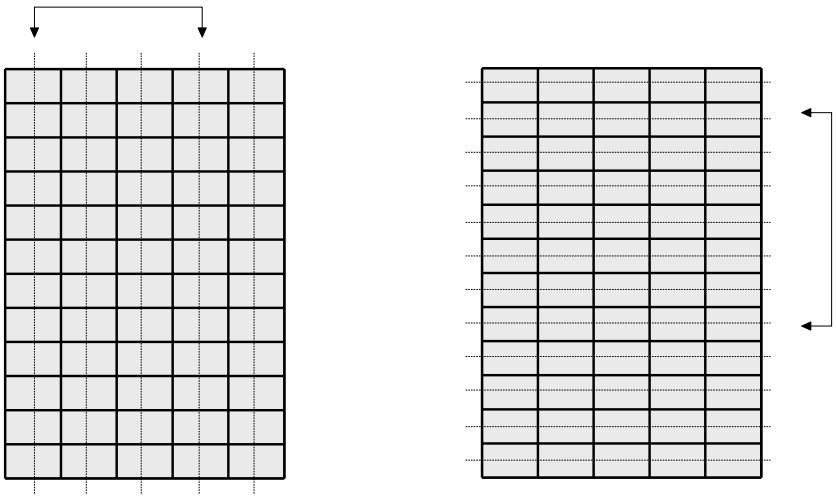


Figure 2.1: Real-valued data can be investigated both in terms of data vectors (right) and in terms of variables (left). Typically, the number of vectors is much higher than the number of variables.

Dissimilarity Data

Besides numerical vectors, more formalisms exist for representation of data. For example:

- Strings represent sequences of symbols, i.e. identifiers of domain-specific entities. For example, proteins may be characterized by strings of amino acid identifiers [Alt97] [PKM06].
- Graphs and, as a special case, trees are popular representations of non-vectorial data. Often molecules' structure is represented as graphs in order to model binding activities by graph alignment approaches. See [WHKK07] for example.
- Sets of symbols are suited for representation of unordered data, i.e. where no hierarchical or successive relation is defined among elements. Sets (and multisets) are widely used, for example, in text mining for representing terms' occurances in a document [Ber03].

Such data structures usually does not meet the requirements of normed vector spaces. Scaling and addition are not meaningfully defined on strings and graphs for instance. However, meaningful measures of (dis)similarity can easily be defined on such data. For example, the dissimilarity of string data is usually quantified by means of an edit distance, e.g. the Levenshtein distance [Lev66]. Measures of (dis)similarity have been defined on sets, e.g. Jaccard similarity [Jac01] and Dice coefficient [Dic45].

A similarity measure is a symmetric function $s : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ that assumes its maximum values on identical elements. A dissimilarity measure is a function $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ that fulfills the first two following conditions.

1. Identity: $d(x_i, x_i) = 0$ for all x_i
2. Symmetry: $d(x_i, x_j) = d(x_j, x_i)$ for all x_i, x_j
3. Triangle inequality: $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k)$ for all x_i, x_j, x_k

A dissimilarity measure is called (metric) distance function if it fulfills the third condition. Metrics are more demanding than (dis)similarity functions. Many similarity functions are convertible into dissimilarity functions, and vice versa. For example, certain similarity functions $s : \mathbb{X} \times \mathbb{X} \to [0, 1]$ are easily transformed into dissimilarities $d = 1 - s$. In case of $s : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ the transformation $d = \frac{1}{1+s}$ is more appropriate. See [OLH08] for details. However, (dis)similarities usually are not convertible into metrics because of the missing triangle inequality.

Dissimilarity data refers to a set $\mathbb{X} = \{x_1, ..., x_n\}$ of data objects where pairwise dissimilarities $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ are available. Vectorial addition and scalar multiplication may not be defined on dissimilarity data. Obviously, vectorial data is also dissimilarity data. See Figure 2.2 for illustration of pairwise dissimilarities.

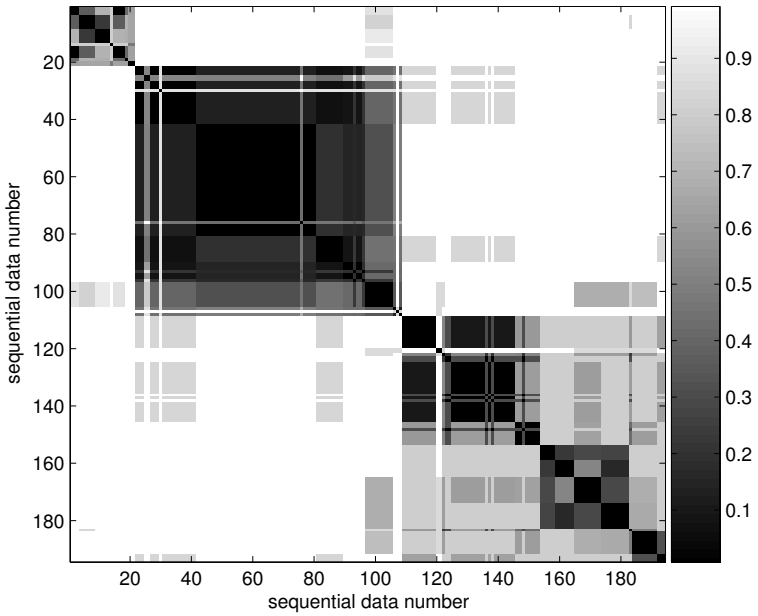


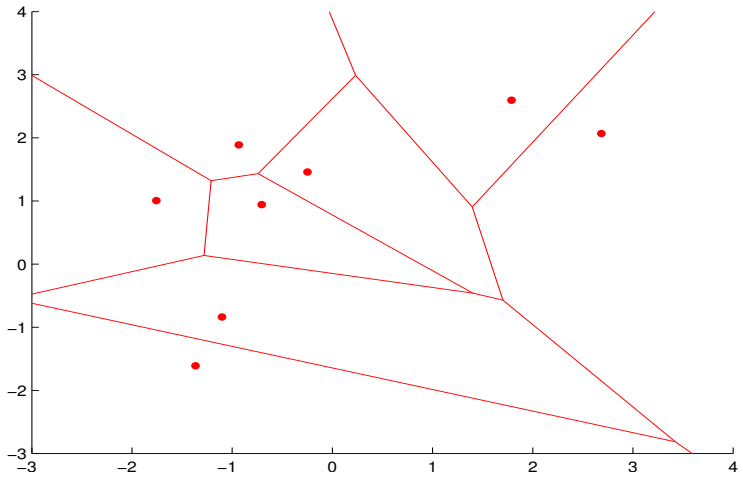
Figure 2.2: Matrix of pairwise dissimilarities among 194 proteins depicted as shades of gray values [PKM06]. Three distinct classes are shown.

Classes

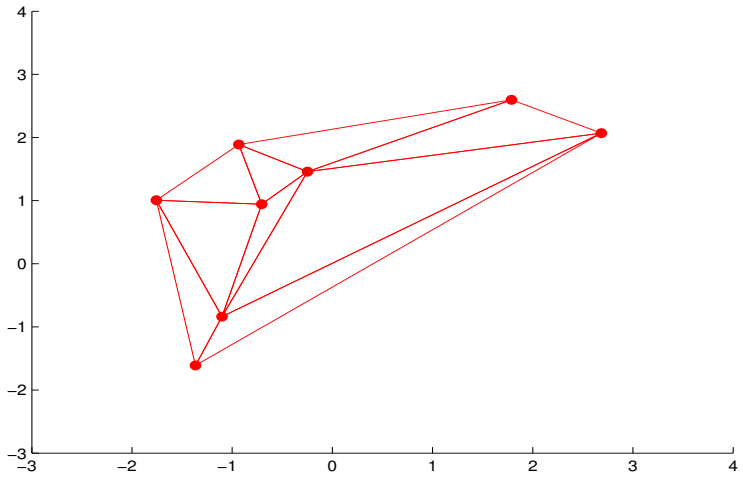
In the context of this thesis, it is assumed that the set of data objects $\mathbb{X}$ consists of disjoint classes $C_1, ..., C_k$ with $\mathbb{X} = \bigcup_{i=1}^{k} C_i$. Each class contains similar data objects according to a meaningful notion of similarity. Objects of different classes are supposedly dissimilar (according to the same concept of similarity). However, the classes are usually unknown in beforehand when trying to analyze a data set. A function $c : \mathbb{X} \to \{C_1, ...C_k\}$ that assigns class labels (or classes) to data objects is called classification.

2.2 Graphs

A particularly useful topic is concerned with graphs. Formally, a graph is a pair of sets (V, E). The set $V \subset \mathbb{X}$ of nodes (vertices) represents the underlying data objects. The set of edges $E \subset V \times V$ represents a similarity or dissimilarity relation among nodes. A graph is undirected iff the set of nodes is symmetric, i.e. for all $(x_i, x_j) \in E$ exists $(x_j, x_i) \in E$. Otherwise a graph is directed. Graphs are represented graphically by drawing a dot for every node, and drawing an arc between two nodes if they are connected by an edge. If the graph is directed, the direction is indicated by drawing an arrow. A graph in which any two nodes are connected by exactly one path is referred to as tree, i.e. any connected graph without cycles is a tree.



(a)



(b)

Figure 2.3: (a) Assigning elements of data space to nearest points $x_1, ..., x_n$ defines Voronoi cells. Borders shown as lines. (b) Delaunay graph represents adjacency of Voronoi cells.

Delaunay Graph

A particularly interesting topic is concerned with Voronoi tessellations [Vor07] and its dual, the Delaunay graph. Let $\mathbb{X} \subset \mathbb{D}$ denote a finite set of points from a normed vector space $\mathbb{D}$. For each $x_i \in \mathbb{X}$ the corresponding Voronoi cell $V(x_i) = \{x \in \mathbb{D} \text{ with } \forall x_j \in \mathbb{X} : \|x - x_i\| \leq \|x - x_j\|\}$ comprises the elements that are closer or equally close to x_i than to any other $x_j \in \mathbb{X}$. In the corresponding Delaunay graph $(\mathbb{X}, E)$ any nodes x_i, x_j are connected with an edge iff the Voronoi cells are adjacent, i.e. $V(x_i) \cap V(x_j) \neq \emptyset$. The Delaunay graph is particularly useful for representing the proximity structure of a data set. See Figure 2.3 for illustration.

Spanning Trees

Given a connected undirected graph (V, E), a spanning tree of that graph is a subgraph which is a tree and connects all the nodes. A weighting function $w : E \to \mathbb{R}_0^+$ with $\forall (x_i, x_j) \in E \ : \ w(x_i, x_j) = w(x_j, x_i)$ might assign a weight to edges in order to express the (un)favorability with respect to a given problem. The weight of a spanning tree then follows as the sum of the weights of its edges. A minimum spanning tree (MST) is a spanning tree whose weight is less than or equal to the weight of every other spanning tree. The MST to a given graph is obtained, for example, by Prim's algorithm [Pri57].

2.3 Knowledge Discovery

KDD is a stepwise process with the aim of turning raw data into knowledge. Ultsch [Ult00b] and Fayyad [FPSS96] agree with the concept of knowledge as formal abstraction of "data in a certain language that is understandable for humans as well as usable by symbol-processing machines". To facilitate the KDD process, it was formalized by breaking it into a number of sequential steps:

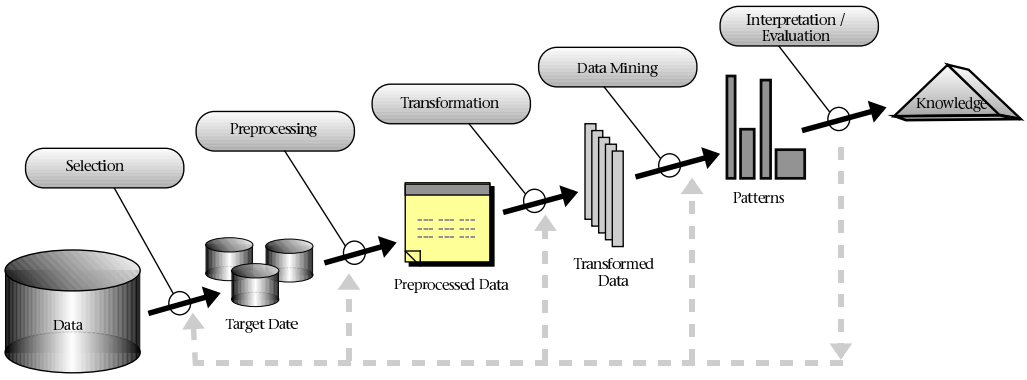


Figure 2.4: Knowledge discovery is a stepwise process including the data mining task [FPSS96]. The KDD process can involve significant iteration and can contain loops between any steps.

Appraisal and Preparation

Appraisal refers to the task of finding out what kind of data is available and how it can be processed for further analysis. The aim is to provide a documentation of both meta-knowledge - such as origin, physical source, and access methods and protocols for the data - and rough properties of the data, e.g. descriptive statistics.

Data preparation refers to the task of correcting errors and generating new variables from available raw data in order to facilitate the discovery and extraction of patterns. Preparation techniques are used for data standardization and treatment of missing values for instance. General guidelines can be found in [Pyl99].

Visual Data Exploration

Visual data exploration is the process of presenting the data in a visual form in order to apply humans' perceptual abilities to the large data sets. For example, projections are used to adjust high-dimensional numerical data for low-dimensional scatter plots, e.g. by means of Multidimensional Scaling [Tor52] [Kru64]. Visualization of data allows the human to get insight into the data and come up with a hypothesis about the underlying structure. The hypotheses can be verified automatic methods from the fields of statistics and machine learning. See [Kei02] for a brief survey on visual exploration methods.

Cluster Analysis

Cluster analysis is the task of finding intrinsic groups, so-called *clusters*, in a data set. Clusters are often interpreted as point clouds in a high-dimensional space whereas each point represents a single data object. See [JMF99] for a comprehensive survey on clustering methods. Cluster analysis can be realized by means of visual data exploration, if high-dimensional cluster structures are depicted and (manually) classified in two- or three-dimensional spaces.

Knowledge Conversion & Validation

Knowledge conversion is the task of putting found patterns down on a symbolic formalism that is understandable to humans and machines as well [UGKL93]. Popular formalisms are rules, frames and trees. Cluster analysis and knowledge conversion techniques are to be combined in order to extract and express patterns for both humans and machine-driven interpreters, e.g. knowledge-based systems using logic-based programming. A knowledge conversion method derives, for instance, typical patterns of the underlying clusters and formulates these point cloud patterns as decision rules or trees.

Finally, validation refers to the task that evaluates the validity, quality and accuracy of the obtained knowledge. A machine-driven interpreter evaluates the knowledge according to the amount of patterns that can be retrieved from the original input data.

2.4 Statistics

Density Estimation

A probability density function $f : \mathbb{D} \to [0, 1]$ of a random variable is a function which describes the aggregation of probability at each point. For the sake of simplicity, the probability density function will be referred to as density. The probability of a random variable falling within a given set is given by the integral of its density over the set. Densities are usually associated with continuous univariate distributions. However, the definition of density comprises discrete and multivariate spaces as well. Densities describe how univariate or multivariate random variables are distributed in data space.

Let $\mathbb{X} = \{x_1, ..., x_n\} \subset \mathbb{D}$ denote a finite set of data vectors, i.e. observed data. Density estimation means the construction of an estimate of an unobservable underlying density function, based on observations $x_1, ..., x_n$. An infinitely large population is distributed according to the underlying density function, at which data $x_1, ..., x_n$ is thought of as a random sample from that population.

Kernel density estimation is a popular method for graphical investigation of variables' distributions. It associates to each data point $x_1, ..., x_n$ a so-called kernel function K, where $K : \mathbb{D} \to [0, 1]$ is usually chosen as a symmetric probability density function satisfying the condition $\int_{-\infty}^{\infty} K(x)dx = 1$ in case of univariate, real-valued data space $\mathbb{D} = \mathbb{R}$. In this context, kernels are assumed to be of gaussian shape:

$$K(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}x^2}$$

The density estimate is the properly normalized sum of these functions. For the univariate case, the density estimate f_h follows as Formula 2.1. From the definition of K follows that f_h itself is a density function and, furthermore, inherits all the continuity and differentiability properties of K. The amount of structure is determined by the bandwidth $h \in \mathbb{R}^+$, in this case the standard deviation of the Gaussian. The bandwidth significantly affects the roughness or smoothness of the density estimate. See Figure 2.5 for illustration.

Formula 2.1

$$f_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Bayesian Probabilities

Bayes' theorem [Bay63] shows the relation between a conditional probability $p(C|x)$ and its inverse $p(x|C)$. Let $p(C|x)$ denote the probability for classification C given the observation x such that

$$p(C|x) = \frac{p(x|C) \cdot p(C)}{p(x)}$$

at which $p(C)$ denotes the prior probability of classification C to occur. The conditional probability $p(x|C)$ for observation x to occur in class C is based on a model

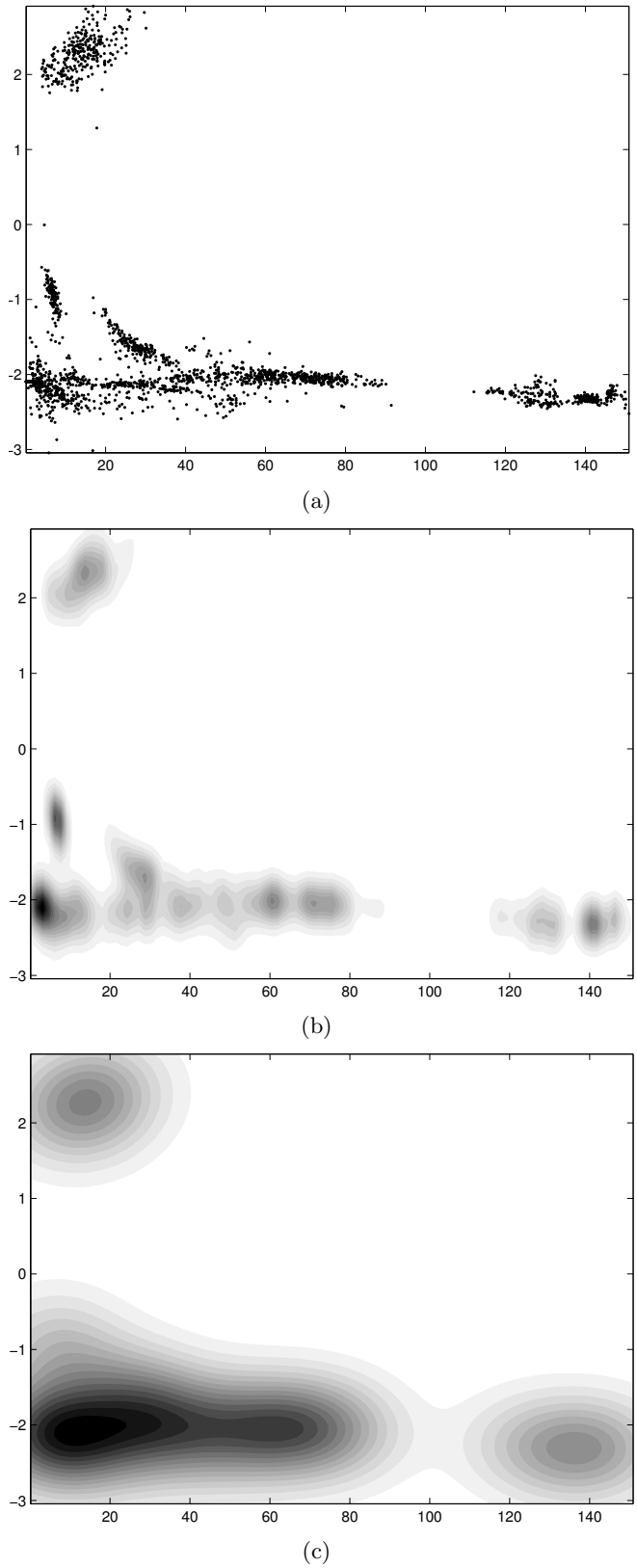


Figure 2.5: Density Estimation performed on random data. (a) Data in $\mathbb{R}^2$. (b) Kernel density estimate with small bandwidth. (c) Kernel density estimate with large bandwidth.

that states how the observations are distributed in class C. The prior probability $p(x)$ states how all observations are distributed, and acts as normalization term. Bayes' theorem is useful for classification according to observations.

$$c(x) = \arg\max_{C} p(C|x)$$

Statistical Hypothesis Testing

A hypothesis test is a method that attempts to refute a specific claim about a domain parameter based on the experimental data, e.g. two available sets of samples (experimentally derived data objects) were drawn from a single underlying domain. To reject a hypothesis is to conclude that it is false. However, to accept a hypothesis does not mean that it is true, only that there is not enough evidence to conclude otherwise.

A common form is the null-hypothesis, i.e. the test is usually stated in terms of both a condition that is doubted (null hypothesis) and a condition that is believed (alternative hypothesis). Statistical tests are based on test statistics, i.e. numerical summaries of data that reduces the data to one or a small number of values. So-called p-values are derived on the basis of test statistics. A p-value is the probability of obtaining a test statistic by chance that is at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. The null hypothesis is rejected if the p-value is less than the significance level $\alpha \in [0, 1]$ which defines the sensitivity of the test. The significance level indicates how often the null hypothesis is approximately rejected when it is in fact true. See [JHK98] [Leh98] for an overview on statistical tests.

The two-sample Student's t-test [Stu08] is a parametric test that relies on normally distributed samples M, M'. The null hypothesis assumes that the underlying means of samples M, M' are actually equal. If the null hypothesis is true, the test statistic t follows Student's t distribution [Stu08], at which s, s' denote the samples' standard deviations with cardinalities n, n'.

$$t = \frac{\bar{M} - \bar{M}'}{\sqrt{\frac{s^2}{n} + \frac{s'^2}{n'}}}$$

The two-sample Mann-Whitney U-test [MW47] is a non-parametric test that is to be preferred over Student's t-test in case of equally shaped, non-normal distributions of approximately equal variance. The null hypothesis in the U-test is that the two populations $M = \{m_1, ..., m_n\}, M' = \{m'_1, ..., m'_n\}$ are drawn from a single underlying domain and, therefore, that their probability distributions are equal. Test statistics U approximates a normal distribution.

$$U = \sum_{i=1}^{n} \sum_{j=1}^{n'} \begin{cases} 1 & : \quad m_i < m'_j \\ 0 & : \quad \text{else} \end{cases}$$

The Kolmogorov-Smirnov test (KS test, [Smi48]) is a nonparametric test of equality of one-dimensional probability distributions used to compare a sample with a reference probability distribution (one-sample KS test) or, in this context, to compare two samples (two-sample KS test). The KS statistic quantifies a distance between the empirical distribution functions of two sample sets M, M'. The null distribution of this statistic is calculated under the null hypothesis that the samples are drawn from the same domain and, therefore, the same distribution. The distributions considered under the null hypothesis are continuous distributions but are otherwise unrestricted. The KS statistic for two given cumulative distribution functions F, F' is given as

$$D_{n,n'} = \sup_x |F_n(x) - F_{n'}(x)|$$

and the null hypothesis is rejected at level α if holds:

$$\sqrt{\frac{nn'}{n+n'}} D_{n,n'} > K_\alpha$$

at which K_α is found from $Pr(K < K_\alpha) = 1 - \alpha$ by means of Kolmogorov distribution K [Smi48].

Standardization

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 $x \leftarrow \frac{x-\mu}{\sigma}$ with mean μ and standard deviation σ. For non-normal distributed variables, a meaningful variance σ^2 may be hard to estimate. Instead, a (robust) min/max-standardization transforms a variable's values $x \leftarrow \frac{x-\nu_{min}}{\nu_{max}-\nu_{min}}$ with robust estimates ν_{min}, ν_{max} for minimum and maximum values. There is empirical evidence by Milligan and Cooper [MC88] that min/max-standardization is to be preferred over Z-score, especially if variances of underlying distributions is hard to estimate.

2.5 Machine Learning

(Un)supervised Learning

There are two major settings in which a classification function c is to be learned. In case of continuous-valued $c : \mathbb{D} \to \mathbb{R}$, supervised learning is referred to as regression. Prediction of class labels with $c : \mathbb{D} \to \{C_1, ..., C_k\}$ is called classification. In supervised learning the values of c are known for a finite set of samples $\{x_1, ..., x_n\}$, called the training set. It is widely assumed that if a hypothesis c' can be found that closely agrees with c on the training set, then c' will be a good guess on the entire domain $\mathbb{D}$, especially on large training sets. For details on supervised learning see [Mit97].

In contrast to that, unsupervised learning constructs a hypothesis c' without any given function values of c. Unsupervised learning seeks to determine the structural patterns of the training set, and expresses found patterns as hypothesis. One particularly interesting form of unsupervised learning is cluster analysis (clustering), i.e. finding subsets of similar data objects. For details on unsupervised learning see [Mit97].

(Un)supervised Learning is often accomplished by means of an objective function. An objective function E evaluates the usability of a hypothesis and, furthermore, yields the construction of c' by means of optimization of E. See [JMF99] [Mit97] for details. The Expectation-Maximization algorithm [DLR77] is a popular method for estimation of models' parameters, e.g. when empirical data is represented as mixture of normal distributions.

Hierarchical Clustering

Cluster analysis (clustering) is an unsupervised learning task that is concerned with the construction of a classification function $c : \mathbb{X} \to \{C_1, ..., C_k\}$ in order to divide the data into homogeneous classes. In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, the data is iteratively partitioned which may run from a single cluster containing all objects to n clusters each containing a single object. See [JMF99] for a review of methods. Hierarchical Clustering may be classified into agglomerative and divisive methods, at which agglomerative are more commonly used. Agglomeration refers to a series of fusions of single objects $\{x_1, ..., x_n\}$ into larger clusters, until a single cluster is obtained. At each stage, the two clusters $C, C' \subset \{x_1, ..., x_n\}$ which are closest together are merged, i.e. with minimum $d(C, C')$. Dendrograms represent the merging made at each successive stage by a two dimensional diagram. For an example see Figure 2.6. There are several agglomerative methods that differ in the way how distance between clusters is defined.

Ward's clustering is one of most popular methods of hierarchical clustering, which is known to produce reliable results in real-world applications. Ward [War63] proposed a clustering procedure that seeks to form clusters which minimize the "information loss" associated with each merging, in terms of an error sum of squares criterion:

$$d(C, C') = ess(C \cup C') - ess(C) - ess(C')$$

$$ess(X) = \sum_{x_i \in X} \left(x_i - \bar{X}\right)^2$$

Quantization

Quantization is the process of approximating a large range of values by a relatively small, finite set of values called codebooks. Formally, a (vector) quantization $q : \mathbb{D} \to W$ assigns elements from the data space to a small set of codebook vectors $W = \{w_1, ..., w_k\} \subset \mathbb{D}$ by means of:

$$q(x) = \arg\min_{w \in W} \|x - w\|$$

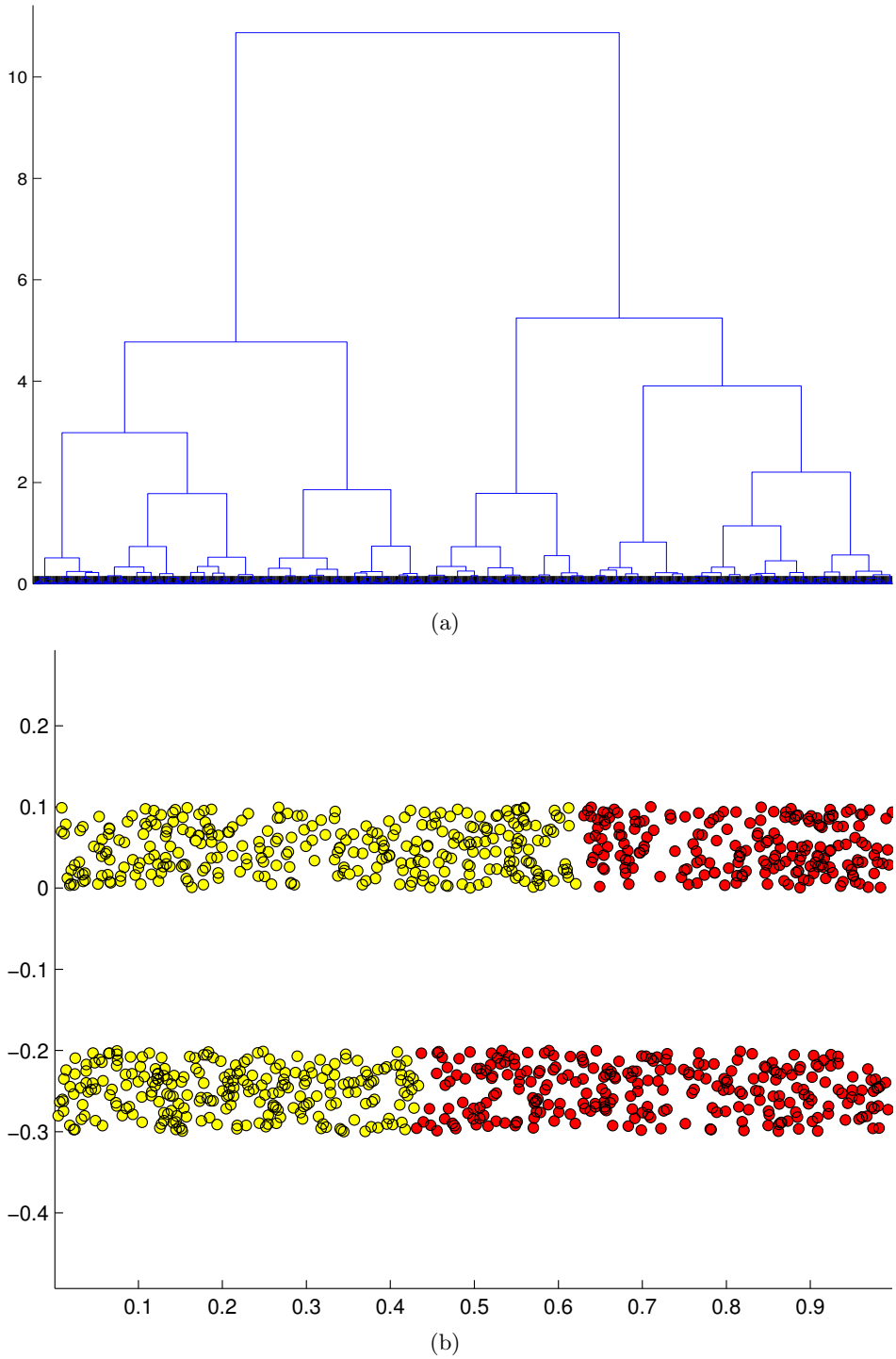


Figure 2.6: Ward's hierarchical clustering. (a) Dendrogram depicts two clusters. (b) Ward's objective function does not match the given cluster structure, which leads to misclassifications of well separated clusters.

Codebook vectors are supposed to follow the density underlying the data space, i.e. more codebooks are located where density is high. Quantization is often used for unsupervised classification (clustering) of vectorial data, by taking the content of each codebook's Voronoi cell as a single cluster. See Figure 2.7 for illustration. The k-means method [Mac67] is the most popular algorithm for construction of vector quantizations. Starting from an initial solution, the codebooks are iteratively updated in order to decrease a certain quantization error term, for example $E = \sum_{x \in \mathbb{X}} \|x - q(x)\|^2$. See Algorithm 2.1 for details.

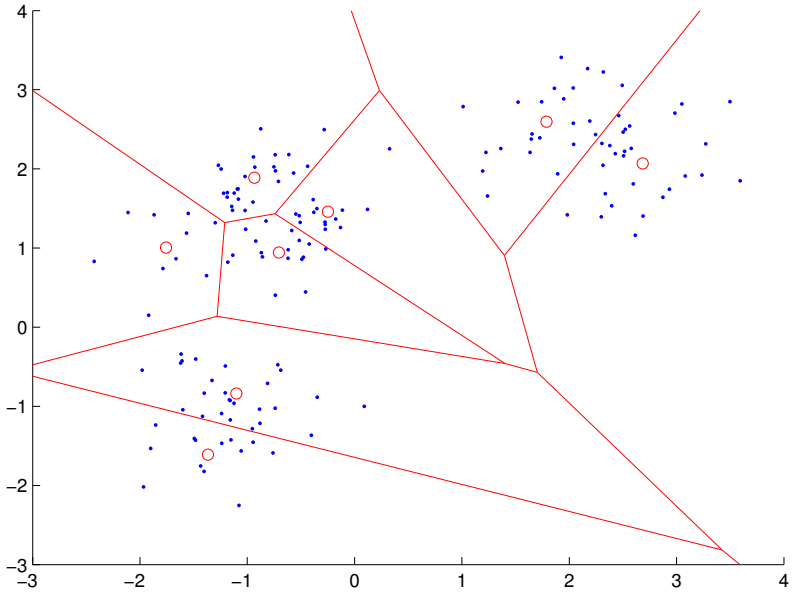


Figure 2.7: Many (blue) data vectors are represented by few (red) codebooks. Quantization induces Voronoi cells.

Algorithm 2.1 k-means algorithm

function KMEANS($k, \{x_1, ..., x_n\}$)
  randomize $W = \{w_1, ..., w_k\}$
  while $\neg$ converged **do**
    for $i \leftarrow 1, ..., n$ **do**
      $q(x_i) \leftarrow \arg\min_{w_j \in W} \|w_j - x_i\|_{\mathbb{D}}$
    end for
    for $j \leftarrow 1, ..., k$ **do**
      $w_j \leftarrow \frac{q^{-1}(w_j)}{|q^{-1}(w_j)|}$
    end for
  end while
end function

Cluster Shapes

In literature it has been widely noticed that many cluster algorithms rely on geometric models with respect to the shape of obtainable clusters. See [DHS01] [ELL01] for an overview. This means that the algorithm imposes a certain geometrical shape, spheres for instance, on the data in order to retrieve clusters. Mismatch of geometrical shapes and clusters in data may lead to unpredicted results, despite a well-separated configuration of clusters in data. See Figure 2.6 for illustration. The obtained results can, therefore, not be trusted until further evaluation is performed, for instance by means of visual depiction of high-dimensional cluster structures in low-dimensional spaces.

Method	Criterion	Preferred shape
Single Linkage	nearest neighbour	unbalanced clusters, "chaining"
Complete Linkage	furthest neighbour	compact clusters of equal diameter
Ward	minimum increase in sum of squares	spherical, ellipsoidal clusters of same size
k-means	within-cluster sum of squares	spherical, convex decision boundaries

Table 2.1: Few algorithms and preferred cluster shapes according to [DHS01] [ELL01].

2.6 Nature-inspired Computing

Many machine learning methods are inspired by principles found in nature. For example, darwinian evolution theory [Dar59] yields an optimization paradigm powerful enough to explain the creation and adaption of complex life forms. Another example is the decentralized organization of tasks in large colonies of social insects by means of stigmergic (indirect) communication. All these biological paradigms have inspired many algorithmic solutions to mathematical problems.

Self-Organization

The term "self-organizing" was first introduced by the psychiatrist and engineer W. Ross Ashby [Ash47]. Self-Organization means the ability of systems to adapt their internal structures automatically to sensory stimuli without external guidance. Here, some kind of mathematical model acts as system. Adaptation of structures means adaption of the underlying mathematical model. The adaption is a function of the system's experience and its environment. In this context, the system is some kind of mathematical model for the purpose of abstraction of spatial cluster structures, i.e. clustering. Stimuli are presented as (numerical) data objects.

Self-organization has first been studied and described in physical and chemical research in order to explain the occurrence of macroscopic patterns based on

processes on the microscopic level. This has been expanded by Bonabeau et al. [BDT99] [CDF+01] in order to explain the behaviour of social insects. They explained that complex group behaviour can emerge from local interactions between simple individuals. Self-organizing systems are neither in need for complex behaviour of individuals, nor do they rely on high order control mechanism.

Stigmergy

Biologist Grassè explained the complex behaviour of social insects by introducing the concept of stigmergy [Gra59]. Several emergent group behaviours of social insects could be explained my means of stigmergy, e.g. task coordination and regulation in the context of nest reconstruction in termites. It could be shown that the building activities of these insects do not depend on the workers but on the current state of the nest itself. Any local state of nest construction triggers a response in a worker which in turn modifies the local state to the next step. This continues until a final or stable state of the nest is achieved. Termites use pheromones to build their complex nests by following a simple decentralized rule set. Each termite scoops up material from its environment, invests it with pheromones, and deposits it on the ground. Termites are attracted to their nestmates' pheromones and are therefore more likely to drop their own material, e.g. mud balls. Over time this leads to the construction of pillars, arches, tunnels and chambers. See Figure 2.8 for illustration.

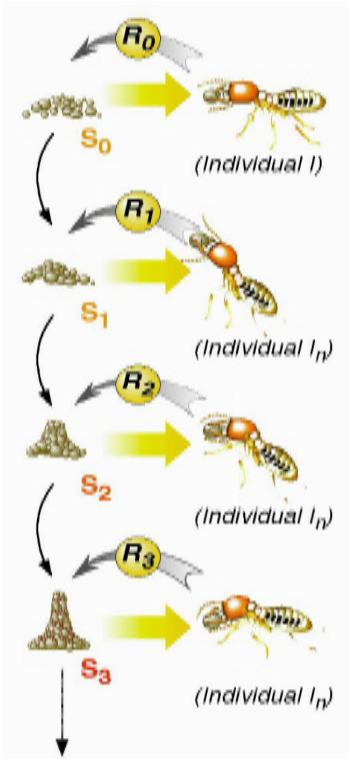


Figure 2.8: Stigmergic communication of termites [Dor01].

Grassè derived the term stigmergy from the greek words stigma (mark, sign) and ergon (work, action). It captures that an insect's actions leave signs in the environment. Other insects sense these signs and that determines their subsequent actions. Formally, stigmergy is a mechanism of spontaneous, indirect coordination between individuals or actions. A trace left in the environment by an action stimulates the performance of a subsequent action, by the same or a different individual. Two types of behaviour patterns can be distinguished. First, individuals can be seen as some kind of state automaton with an internal state and transformations executed mainly by environmental factors. The second type does not need an internal state in the individuals. Actions can be achieved using only external stimuli. The environment acts as an external memory.

Self-organization often requires interactions among individuals. Interactions and communication, respectively, can be direct or indirect. A direct interaction would be contact between two individuals. Indirect interactions are more latent, however, as they occur when one individual modifies the environment which another individual responds to at another time. This is referred to as stigmergic communication.

Swarm Intelligence

Swarm intelligence (SI) refers to natural or artificial systems that exhibit collective behavior by means of decentralized self-organized entities. SI can be found in natural phenomena such as ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. SI systems typically consist of a population of simple individuals, interacting locally with one another and with their environment. Individuals are supposed to follow very simple rules and, furthermore, there is no centralized control that determines how individuals should behave. Social insects work without supervision, i.e. their teamwork is mostly self-organized. Coordination of individuals has its source from interactions among individuals and with their environment. For example, an ant merely follows the trail left by another. Taken together these interactions allow to retrieve the shortest route to a food source among myriad possible paths. Local interactions between individuals and their environment lead to the emergence of complex global behavior. See Figure 2.9 for illustration. The term swarm intelligence was introduced by Beni and Wang [BW89] in the context of cellular robotic systems. For an overview on SI methods see [BDT99] [BM01] [RFR06].

From the definition, a SI system provides an environment, with lots of unsophisticated, primitive entities that interact locally with the environment and other entities and, finally, causes the emergence of coherent functional global (behavioural) patterns. SI provides a basis for algorithms that aim at solving distributed problems without centralized control or the provision of a global model. Many SI methods are concerned with optimization tasks and have been successfully applied for solving traveling salesman type of problems, e.g. Ant Colony Optimization [Dor92] and Particle Swarm Optimization [KE95]. For unsupervised machine learning and data visualization the algorithms Ant-Based Clustering [LF94], Databots [Ult00a] and Schelling's Segregation Model [Sch69] are of interest.

Figure 2.9: Swarm intelligence: ants clustering corpses onto piles during several hours [BDT99].

Agents

According to literature [Goe06] [CW09] an agent is a software entity that fulfills tasks delegated by the user, can communicate with the user or other agents, has actors to modify its environment, has sensors to gain information about its environment, tries to satisfy internal goals, acts on its own, based on the sensor input and its internal goals. There is no control mechanism of higher order to coordinate the agents' activities. Agents are often applied for distributed problem solving. The initial problem is split into several subproblems which are solved by agents. This is often accomplished using multi agent systems (MAS). MAS consist of numerous agents, and an environment providing communication exchange among agents.

Agents are classified according to their abilities. Reflex agents can only react on sensorial input. Goal-based agents act according to a goal known due to some sort of formalism (e.g. energy function). These agents pick the action with the outcome that most likely brings the agent closest to the goal. This is comparable with greedy hill-climbing algorithms. Utility-based agents are more sophisticated. They can create a plan of action and create sub-goals to achieve their personal goal. One of the first Multi Agent Systems beyond cellular automata was Schelling's Segregation model [Sch69], at which simple ethnic agents could move on a low-dimensional grid in order to locate themselves in an ethnically pure neighbourhood. Chli and DeWilde [WNL99] [CDWG+03] have investigated conditions for stable organization of simulated agents that are producing, consuming and trading resources.

Artificial Neural Networks

The term artificial neural networks (ANN) refers to algorithms that mimic the structure and functional aspects of biological neural networks in order to solve a machine learning problem. ANN are a connectionist approach to computation, i.e. they consist of interconnected artificial neurons that process information in a distributed, parallel way. Each neuron is an elementary signal processing unit. ANN have a learning phase at which the internal structure is adapted based on external stimuli, i.e. numerical input data objects. ANN are used to model complex

relationships between inputs and outputs or to find patterns in data. Popular unsupervised learning ANN are Self-Organizing Maps [Koh89].

Chapter 3

Topographic Mapping

3.1 Concepts

In geographical sciences for instance, topographic maps represent any real-world object on a plain (paper or computer monitor) by means of two-dimensional location and contour lines to depict elevation change on the surface of the earth. For details see [Hat08]. In the broader context of data mining, topographic mapping refers to creation of a low-dimensional image of the topography of a high-dimensional data set. Cluster analysis aims at identifying groups of similar data objects in a given data set (cf. Section 2.3). By this, a notion of cohesiveness is established between the data objects of a cluster. Topographic mapping methods add another analytical aspect. Topographic mapping aims at establishing a relation between clusters and between data objects as well, i.e. capturing the overall similarity structure of the data and putting it into a more abstract yet understandable formalism.

Definition 3.1 *The* topography *of a given finite set* $\mathbb{X} \subset \mathbb{D}$ *is the set of all pairwise dissimilarities among elements of* $\mathbb{X}$*.*

A topographic mapping $m : \mathbb{X} \to \mathbb{O}$ aims to preserve the topography of $\mathbb{X}$ in the low-dimensional output space $\mathbb{O}$ such that $m(x_i), m(x_j)$ are (dis)similar iff x_i, x_j are (dis)similar for all pairs of elements. For the sake of simplicity, the following notation is used:

- $\mathbb{X} = \{x_1, ..., x_n\}$ denotes the set of data objects.
- The data objects $\mathbb{X}$ are mapped onto $\{m(x_1), ..., m(x_n)\} = \{y_1, ..., y_n\} \subset \mathbb{O}$.
- $d_{\mathbb{D}} : \mathbb{D} \times \mathbb{D} \to \mathbb{R}_0^+$ denotes the dissimilarity function of the data space.
- $d_{\mathbb{O}} : \mathbb{O} \times \mathbb{O} \to \mathbb{R}_0^+$ denotes the distance function of the output space.

According to [BHV99], topography may be captured by means of several principles:

- pairwise dissimilarities,
- ranks of dissimilarities or
- (geometrical) neighbourhood relations, which is refered to as topology.

Definition 3.2 *A* topology *of a set* $\mathbb{X} = \{x_1, ..., x_n\}$ *of data objects the set of all pairwise neighbourhood relations on* $\mathbb{X}$*.*

Especially interesting are geometrical neighbourhood relations that are invariant to basic operations such as rotation, stretching, scaling or translation for instance. For the sake of simplicity, we will restrict our considerations to adjacency of Voronoi cells. Thus topological mappings refer to a special case of topographic mappings that aim at preservation of topology. In case of topological mappings, neighbourhood relations among $\{x_1, ..., x_n\}$ are to be deduced from its image $\{y_1, ..., y_n\}$ but scale is largely disregarded. In general it is not possible to preserve topographies correctly when mapping to lower-dimensional spaces [Kir78] [Dry78] [Sch80]. Topologies can only be preserved if the effective dimension of data space and output space match [BHV99]. Effective dimension is the dimension of a submanifold of $\mathbb{D}$ which contains all the data. Thus, misrepresentations of topological structures are to be expected when trying to preserve data in effectively lower-dimensional output spaces.

Learning Methods

In literature the topographic mapping subject has been addressed by several projection techniques, i.e. methods that are used for reducing the dimensionality of the data items. For an overview see [Kas97]. In this thesis, the term *mapping* refers to all dimensionality reducing projections based on linear and non-linear, orthogonal and non-orthogonal projections. Many topographic mappings are iteratively constructed, i.e. the mapping's image $\{y_1, ..., y_n\}$ is updated by a hopefully improved image $\{y'_1, ..., y'_n\}$. This is usually referred to as *learning*. Learning methods are based on few noteworthy principles.

Linear methods, such as Principal Component Analysis (PCA, [Pea01]), try to find a basis for construction of a linear transformation that maps high-dimensional vectors onto an output space of lower dimensionality. However, these (orthogonal) projections cannot deal with data that is arranged on non-linear manifolds in data space. See [UM06] for illustration. Non-linear embeddings assume that the data objects $\mathbb{X}$ are arranged on a low-dimensional (embedded) manifold in the data space. See Figure 3.1 for illustration. If the manifold has a low-dimensional effective dimension, then a set of points is to be found in the visualizable low-dimensional output space that reproduces the topography of $\mathbb{X}$. Multidimensional Scaling [Tor52] [Kru64] aims at preservation of all pairwise dissimilarities. In contrast to that, LLE [RS00] tries to preserve a selected subset of pairwise dissimilarities. Non-linear embeddings are, in principle, more powerful than linear methods.

However, these methods frequently misrepresent intricate but well-separated cluster structures. Nature-inspired methods, such as Self-Organizing Maps [Koh89] and Ant-Based Clustering [LF94], are based on a large number of interacting models that behave according to simple rules in order to mimic biological entities. It is however unknown which principle yields the construction of the most meaningful mappings.

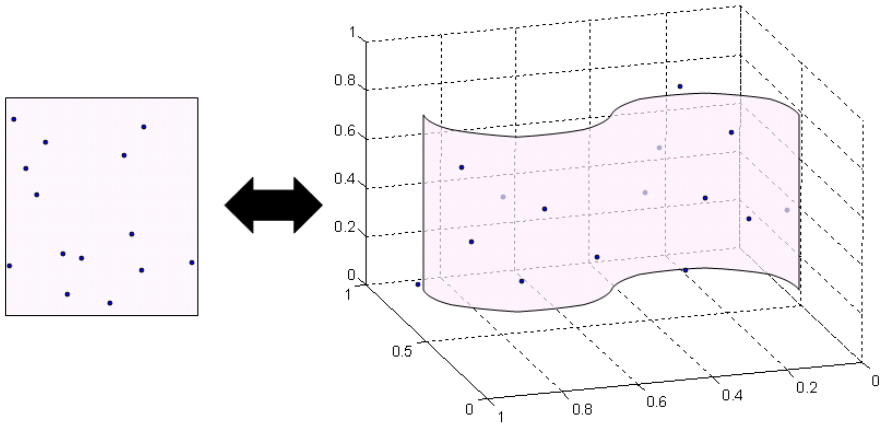


Figure 3.1: Topographic mapping: (right) data points are located nearby a low-dimensional submanifold embedded in data space, (left) the unfolded manifold is visualized in $\mathbb{R}^2$.

Definition 3.3 *In the context of this thesis, learning is referred to as* focusing *if the learning algorithm excludes more and more pairwise dissimilarities during each update.*

Usually, focusing algorithms first capture global (inter-cluster) structures, then more local (intra-cluster) data structures are captured. A focus may be realized by means of shrinking neighbourhoods (i.e. shrinking radius $\sigma \in \mathbb{R}_0^+$) in data space or output space. Typically, a monotonically decreasing focus function $F_\sigma : \mathbb{R}_0^+ \to [0,1]$ is composed with a dissimilarity function in order to exclude pairwise dissimilarity relations from the learning process. Popular functions are F_{bubble} and F_{bell} (see [DH97] [Koh97]):

$$F_{\sigma,bubble}(x) = \begin{cases} 1 & : \quad x \leq \sigma \\ 0 & : \quad \text{else} \end{cases}$$

$$F_{\sigma,bell}(x) = e^{-\frac{x^2}{2\sigma^2}}$$

The canonical benchmark example of contradicting global and local structures was proposed as the so called "Chainlink" dataset [UV94]. See Figure 3.2 for illustration. For each ring there are some points closer to the center of the other ring than its own. Non-focusing learning might lead to misrepresentations, whereas some focusing methods perform correctly. See [UM06] for details. It is, however, unknown whether focusing algorithms are in principle more powerful than non-focusing algorithms.

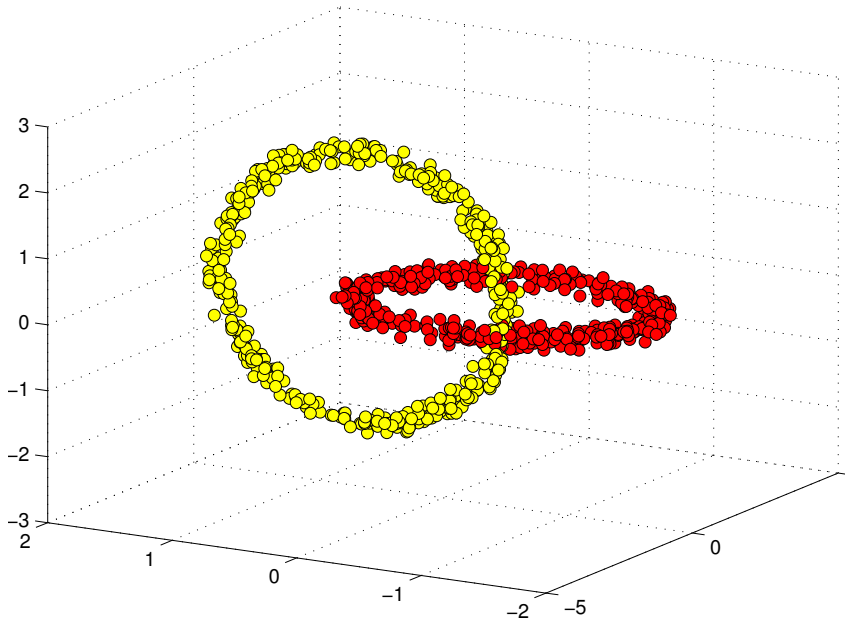


Figure 3.2: Chainlink benchmark data [UV94]: some points are closer to the center of the other chain than its own.

Output Spaces

The appearance of topographic mappings $m : \mathbb{X} \to \mathbb{O}$ depends on the structure of the low-dimensional output space $\mathbb{O}$. Two kinds of output spaces can be found in literature. Mathematical spaces in the sense of $\mathbb{R}^2$ or $\mathbb{R}^3$ are unbounded and continuous vector spaces with norm $\|.\|_{\mathbb{O}}$. In contrast to that, regular grids are often used to realized topographic mappings with nature-inspired algorithms (cf. [DAGP89] [Koh89]). Grids are finite, regular, low-dimensional graphs with a distance $d_{\mathbb{O}} : \mathbb{O} \times \mathbb{O} \to \mathbb{R}_0^+$. To avoid borders, periodic boundaries may be introduced such that the output space mimics the surface of a torus. Such output spaces may be depicted as adjacent tiles. The elements of the output space are denoted as $o_i \in \mathbb{O}$.

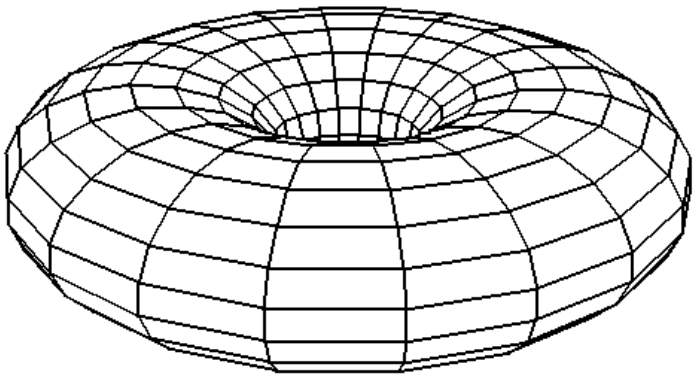


Figure 3.3: Toroidal output space.

3.2 Self-Organizing Batch Maps

The Self-Organizing Batch Map (Batch-SOM) by Kohonen [Koh89] [Koh97] is a popular, widely adapted artificial neural network used for topographic mapping. The SOM was loosely inspired by the observation of topologically arranged sensory maps in the human cortex. Neighbouring neurons tend to respond to neighbouring regions of, for instance, the retina and body surface. For details see [RMS92] [Koh89]. The Batch-SOM constructs a topographic mapping from a finite set $\mathbb{X} = \{x_1, ..., x_n\}$ originating from a normed vector space $\mathbb{D}$. The mapping's image $\{y_1, ..., y_n\}$ is arranged on a finite, fixed, regular grid $\mathbb{O} \subset \mathbb{N}^2$. Each grid node $o_i \in \mathbb{O}$ has got a codebook vector $w_i \in \mathbb{D}$ for quantization purposes. In literature, a node and its codebook vector are likewise referred to as *neuron*. The learning algorithm of the SOM modifies codebook vectors in order to approximate density and proximity structure of $\mathbb{X}$. The codebook vectors represent the underlying data manifold on $\mathbb{O}$ for analytical purposes. See Figure 3.4 for illustration..

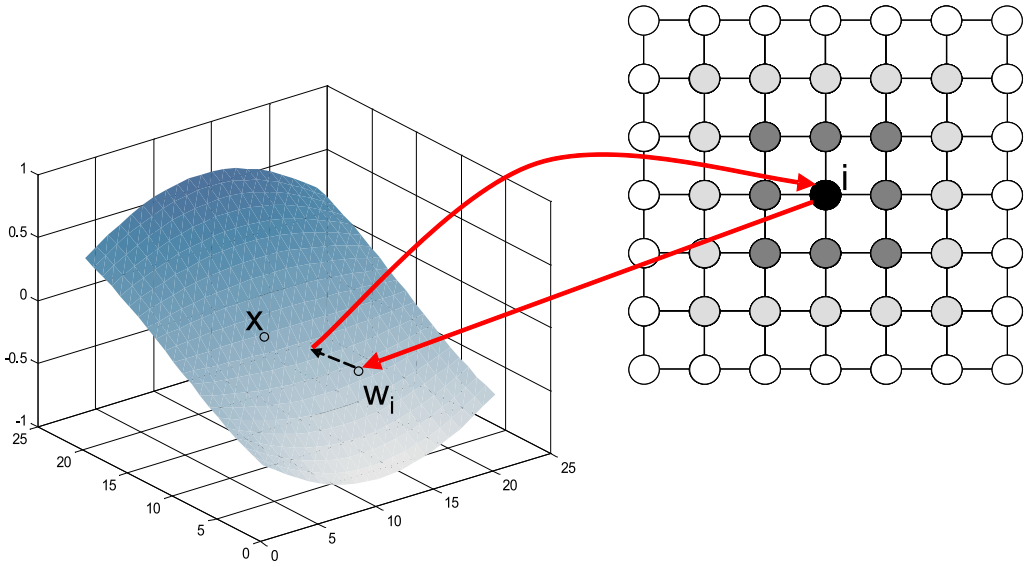


Figure 3.4: Batch-SOM: vector x_j from the blue data manifold triggers best matching node y_j and neighbours to shift their codebooks towards x_j. The magnitude of the shift is indicated by shades of gray. The shift of codebook vectors is only depicted, though, for node i.

The learning algorithm of the Batch-SOM is iterative. For each learning step $t = 1, ..., t_{max}$ the mapping's images $\{y_1, ..., y_n\}$ are simultaneously updated (cf. Formula 3.4). This is referred to as *bestmatch search*. Then, all codebook vectors w_i are updated according to Formula 3.6. After each step, the neighbourhood radius $\sigma \in \mathbb{R}^+$ is decreased according to a pre-defined annealing scheme. See Algorithm A.1 for details. The Batch-SOM is an approximation of the Online-SOM [Koh89] which shifts codebook vectors towards input data x_i such that $w_j \leftarrow w_j + \alpha \cdot F_\sigma(d_{\mathbb{O}}(y_i, o_j)) \cdot (x_i - w_j)$ with learning rate $\alpha \in [0, 1]$.

Formula 3.4

$$y_j \leftarrow \arg\min_{o_i \in \mathbb{O}} \Psi(x_j, o_i)$$

Formula 3.5

$$\Psi(x_j, o_i) = d_{\mathbb{D}}(x_j, w_i)$$

Formula 3.6

$$w_i \leftarrow \frac{\sum_{x_j \in \mathbb{X}} F_\sigma(d_{\mathbb{O}}(y_j, o_i)) \cdot x_j}{\sum_{x_j \in \mathbb{X}} F_\sigma(d_{\mathbb{O}}(y_j, o_i))}$$

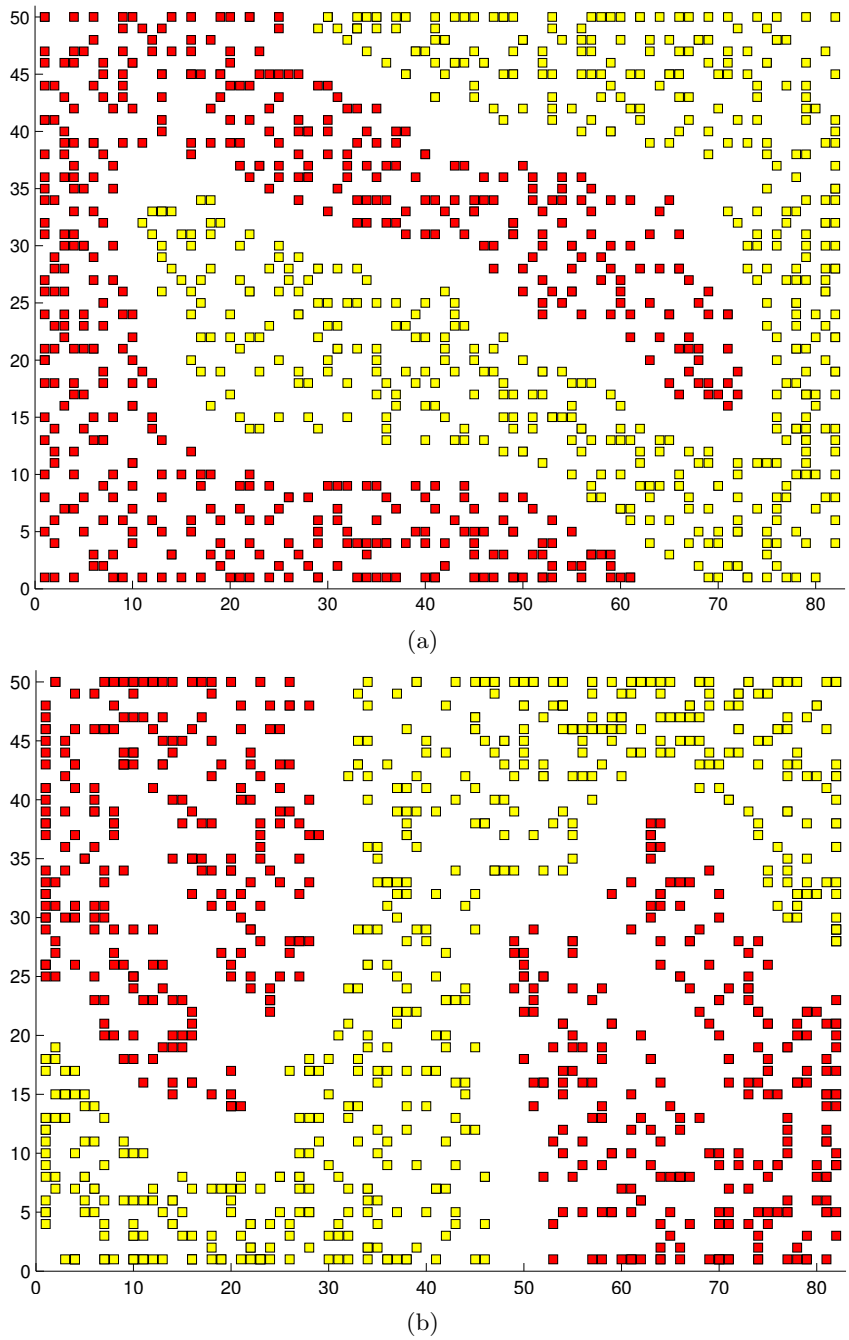
In literature, Self-Organizing Maps (SOM) with few grid nodes can be found merely, cf. [Koh89] [Ves02]. In these SOM, the number of neurons corresponds to the number of clusters assumed in the input data. Usually, this number is very small (≈ 20). In contrast to that, SOM may be used as tools for visualization of structural features of the data space. A characteristic of this paradigm is the large number of neurons, usually several thousands (≈ 4000) of neurons. These SOM allow the emergence of intrinsic structural features of the data space on the map. They are called Emergent Self-Organizing Maps (ESOM, [Ult99]).

Several derivatives of the Batch-SOM have been proposed, e.g. in order to cope with dissimilarity data. The Dissimilarity-SOM, was proposed by Kohonen and Somervuo [KS02] as a counterpart for dissimilarity data. For a given data set $\mathbb{X} = \{x_1, ..., x_n\}$ with pairwise dissimilarities $d_{\mathbb{D}} : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}_0^+$ the *generalized median* follows as:

$$\bar{X} = \arg\min_{x_i \in X} \sum_{x_j \in X} d_{\mathbb{D}}(x_i, x_j)$$

Each codebook vector is updated according to $w_i \leftarrow \bar{X}_i$ with $X_i = \{x_j \in \mathbb{X} : d_{\mathbb{O}}(y_j, o_i) \leq \sigma\}$ being the set of objects mapped within the actual neighbourhood radius σ. There is no proof of convergence for the Dissimilarity-SOM, yet. The Dissimilarity-SOM is prone to produce *model collisions*, i.e. identical codebooks on different nodes due to the discrete nature of the generalized median. A complex branch and bound approach was proposed in order to solve this so-called *model collision* problem [Ros07].

In literature it has been extensively notices that the Batch-SOM greatly depends on an appropriate annealing of radius σ that matches the proximity structure of the input data $\mathbb{X}$. For details see [Koh97] [NVK07] [GBO98]. For given input data, however, an appropriate annealing scheme this is an unknown quality in advance. For illustration of effects see Figure 3.5.



(a)

(b)

Figure 3.5: Batch-SOM maps Chainlink data onto 50×82 sized planar grid. (a) Matching annealing scheme. (b) Too fast annealing leads to misrepresentations, three clusters instead of two.

3.3 Curvilinear Component Analysis

Algorithm

Curvilinear Component Analysis [DH97] is a non-linear embedding method inspired by Self-Organizing Maps [Koh89] and Multidimensional Scaling [Kru64]. Curvilinear Component Analysis (CCA) was proposed by Demartines and Hérault [DH97] in order to overcome the problems that occur when trying to reproduce distances $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ in effectively lower-dimensional spaces. When mapping data set $\mathbb{X} \subset \mathbb{D}$ from a nonlinear data manifold to low-dimensional output space $\mathbb{O}$ not all pairwise distances can be preserved [Dry78] [Kir78]. In fact, reproduction of distances usually cannot be achieved, so that large distances in data space may be preserved to the disadvantage of local proximities. CCA excludes less important relations from the learning process by means of a focus function F_σ. Formally, this reasoning leads to the following error function:

Formula 3.7

$$E = \sum_{x_i, x_j \in \mathbb{X}} E_{ij} \quad with \quad E_{ij} = \left(d_{\mathbb{D}}(x_i, x_j) - d_{\mathbb{O}}(y_i, y_j)\right)^2 \cdot F_\sigma\left(d_{\mathbb{O}}(y_i, y_j)\right)$$

Thus, Kruskal's raw stress [Kru64] is enhanced by means of focus F_σ in order to mimic the learning of Batch-SOM. The error function rapidly vanishes to zero when output distances are large. Radius $\sigma \in \mathbb{R}_0^+$ is decreased over time according to a predefined annealing scheme. See Algorithm A.2 for details. In CCA a randomly chosen projection point $y_i = m(x_i)$ is temporarily fixed, and all other y_j move around in order to adjust the pairwise distances. The update rule [DH97] for CCA follows as gradient descend $\Delta y_j = -\alpha \cdot \frac{\partial E_{ij}}{\partial y_j}$. See Formula 3.8 with $\alpha \in (0, 1)$ being the learning rate that is also decreased down to 0.

Formula 3.8

$$\Delta y_j = \alpha \cdot F_\sigma(d_{\mathbb{O}}(y_i, y_j)) \cdot (d_{\mathbb{D}}(x_i, x_j) - d_{\mathbb{O}}(y_i, y_j)) \cdot \frac{y_j - y_i}{d_{\mathbb{O}}(y_j, y_i)}$$

Properties

CCA aims at topographical mappings of input data such that pairwise distances in output space reflect distances' scale in data space. Mismatching annealing of radius σ usually leads to severe misrepresentation of available proximities in data. See Figure 3.6 for illustration. In addition to SOM-like annealing, Demartines and Hérault [DH97] propose a user-controlled scheme for CCA, allowing an interactive selection of σ at which the *unfolding* of the data onto the low-dimensional output space takes place. This approach produces mappings of unknown quality and is of limited reproducibility .

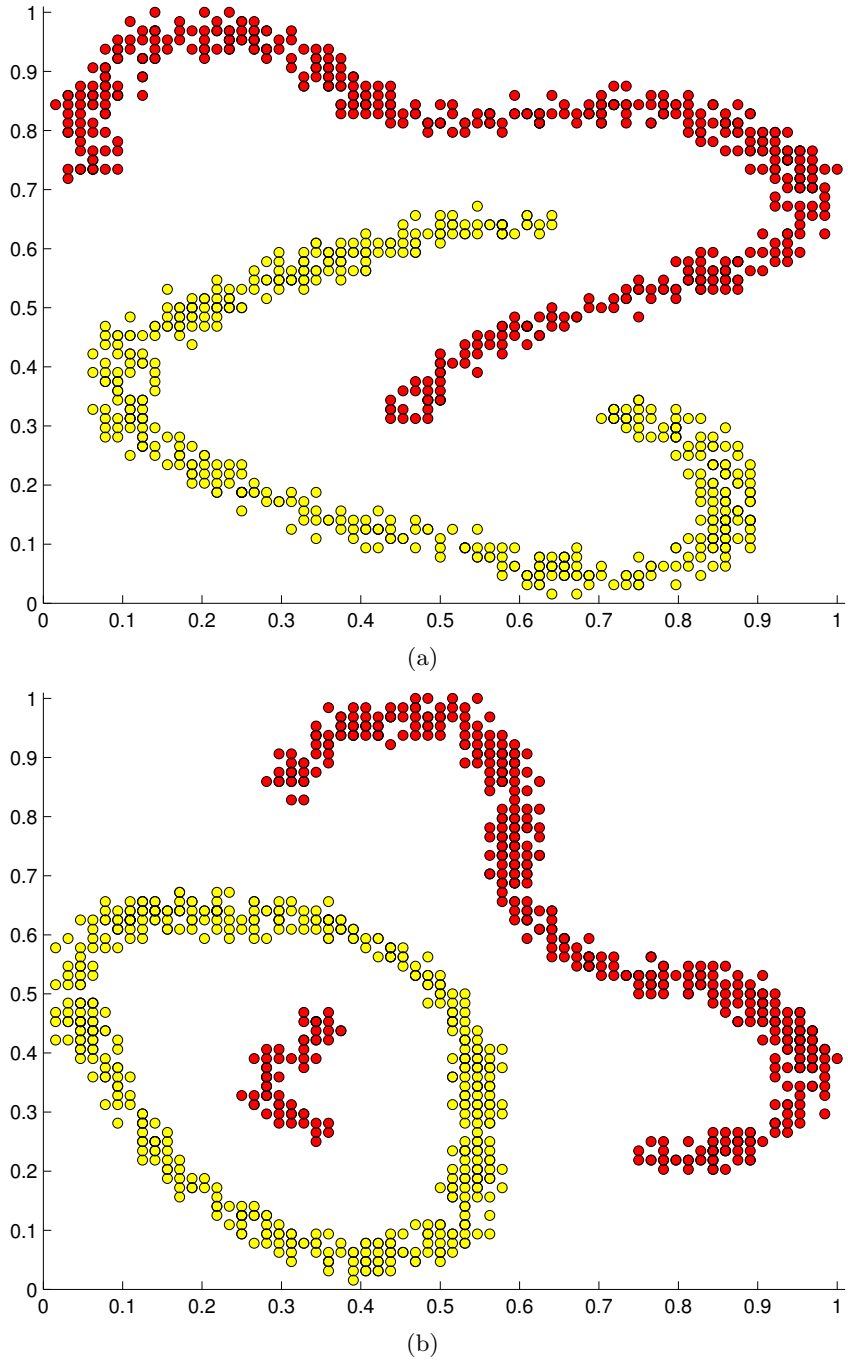


Figure 3.6: Chainlink data mapped with CCA. (a) Matching annealing scheme preserves clusters' cohesion. (b) Mismatching annealing falsely depicts three clusters.

3.4 Ant-Based Clustering

Algorithm

Ant-based clustering (ABC) is a nature-inspired heuristic first introduced as a model for explaining emergent behavior observed in real ant colonies [DGF+90]. More recently, it has been applied in a data-mining context to perform both clustering and topographic mapping [LF94] [HKD05].

A data set $\mathbb{X} = \{x_1, ..., x_n\}$ with pairwise dissimilarities $d_{\mathbb{D}} : \mathbb{X} \times \mathbb{X} \to [0, 1]$ is mapped onto a regular low-dimensional grid $\mathbb{O} \subset \mathbb{N}^2$. Simulated stochastic agents, called ants, are supposed to modify the topographic mapping by changing the set $\{y_1, ..., y_n\} \subset \mathbb{O}$ of mapped objects. No more than a single data object is mapped on each grid node, i.e. the mapping is injective. Let $\{a_1, ..., a_{n'}\} \subset \mathbb{O}$ denote the current locations of the ants. Ants perform random walks. When facing an occupied grid node, ant ant might pick up the data object. When facing an empty grid node, an ant might drop its carried data object. The probabilities for picking or dropping $x \in \mathbb{X}$ on node $a \in \mathbb{O}$ is denoted with p_{pick} and p_{drop}, respectively. Threshold constants $k_p, k_d \in \mathbb{R}^+$ calibrate these dynamics [DGF+90].

$$p_{pick}(x, a) = \left(\frac{k_p}{k_p + \phi(x, a)} \right)^2$$

$$p_{drop}(x, a) = \left(\frac{\phi(x, a)}{k_d + \phi(x, a)} \right)^2$$

An ant located at $a \in \mathbb{O}$ perceives the surrounding $\sigma^2 \in \{9, 25\}$ quadratically arranged nodes. See Figure 3.4 for illustration. The set of objects mapped onto this *perceptive neighbourhood* is denoted with $N(x_i, a) = \{x_j \in \mathbb{X} : j \neq i,\ y_i \text{ neighbouring } a\}$. Thus, $\phi : \mathbb{X} \times \mathbb{O} \to \mathbb{R}_0^+$ determines the "attractiveness" to map objects on certain nodes. See Algorithm for A.3 details.

Formula 3.9

$$\phi(x_i, a) = \frac{1}{\sigma^2} \sum_{x_j \in N(x_i, a)} \left(1 - \frac{d_{\mathbb{D}}(x_i, x_j)}{\alpha} \right)$$

Properties

ABC methods lead to a local sorting of objects in terms of similarities. Ants gather scattered input samples into dense piles. In literature, it has been noticed that ABC derivatives are prone to produce too many and too small clusters [RA04] [HKD05] [AI06]. For illustration see Figure 3.4. The raw ABC algorithm does never converge into a meaningful fixed point, because the probability for an ant to remove an object from its associated cluster never approximates zero.

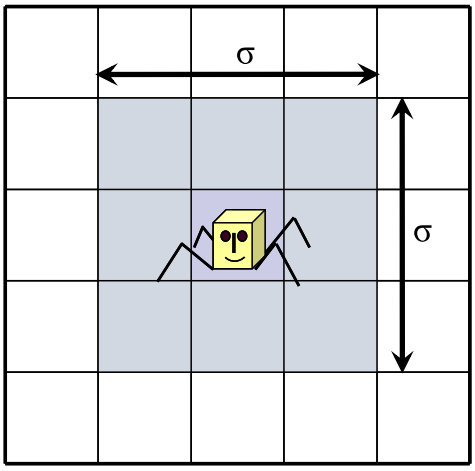


Figure 3.7: Ant-Based Clustering. Perceptive area of an ant.

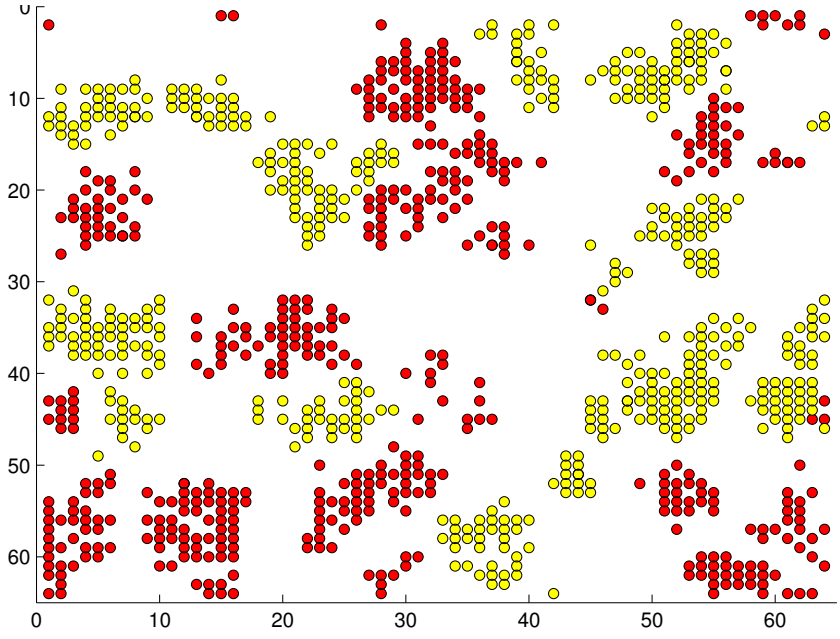


Figure 3.8: Ant-Based Clustering maps Chainlink data onto a 64×64 toroidal grid. Too many and too small clusters emerge.

3.5 Stochastic Neighbour Embedding

Algorithm

Stochastic Neighbour Embedding (SNE) relies on a probabilistic formulation of topography [HR02]. The probability for x_i, x_j being neighbours in data space is denoted as $p_{ij} \in [0, 1]$. The probability for the mapping's images y_i, y_j being neighbours in output space is denoted as $q_{ij} \in [0, 1]$.

Formula 3.10

$$p_{ij} = \frac{e^{-\frac{d^2_{\mathbb{D}}(x_i, x_j)}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{d^2_{\mathbb{D}}(x_i, x_k)}{2\sigma_i^2}}}$$

Formula 3.11

$$q_{ij} = \frac{e^{-d^2_{\mathbb{O}}(y_i, y_j)}}{\sum_{k \neq i} e^{-d^2_{\mathbb{O}}(y_i, y_k)}}$$

A mapping is obtained by minimizing the the difference between the original distribution (p_{ij}) and the distribution (q_{ij}) of the mapped objects. This is realized by minimization of the Kullback-Leibler [KL51] divergence $E = \sum_{i,j} p_{ij} \cdot \log \frac{p_{ij}}{q_{ij}}$ by means of gradient descent methods, i.e. the gradient $\frac{\partial E}{\partial y_i}$ is scaled with a learning rate α in order to modify the images y_i. See Algorithm A.4 for details.

Properties

The neigbourhood radius $\sigma_i \in \mathbb{R}^+$ determines how the proximity structure is perceived by the SNE algorithm. Radius σ_i is found by a binary search for the value that makes the entropy of the distribution over neighbors equal to $logk$. Here, $k \in \mathbb{N}$ is the effective number of local neighbors ("perplexity") and is chosen by hand. A small perplexity disregards global structure, whereas large perplexity ignores local structures. See Figure 3.9 for illustration. The t-SNE method [vdMH08] is a derivative of the original SNE. It uses Student-t distributions instead of gaussians in order to avoid the crowding of points in the center of the output space.

3.6 Other Methods

Linear Transformations

Linear methods, such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA), try to find a basis $B \in \mathbb{R}^{k' \times k}$ for construction of a linear function $y_i = Bx_i$ that maps k-dimensional data points onto an output space of dimension $k' < k$ which is spanned by the first k' principal components. For example, the PCA [Pea01] finds a new orthonormal basis in order to re-express the data set as linear combinations of its basis vectors. Hopefully this new basis

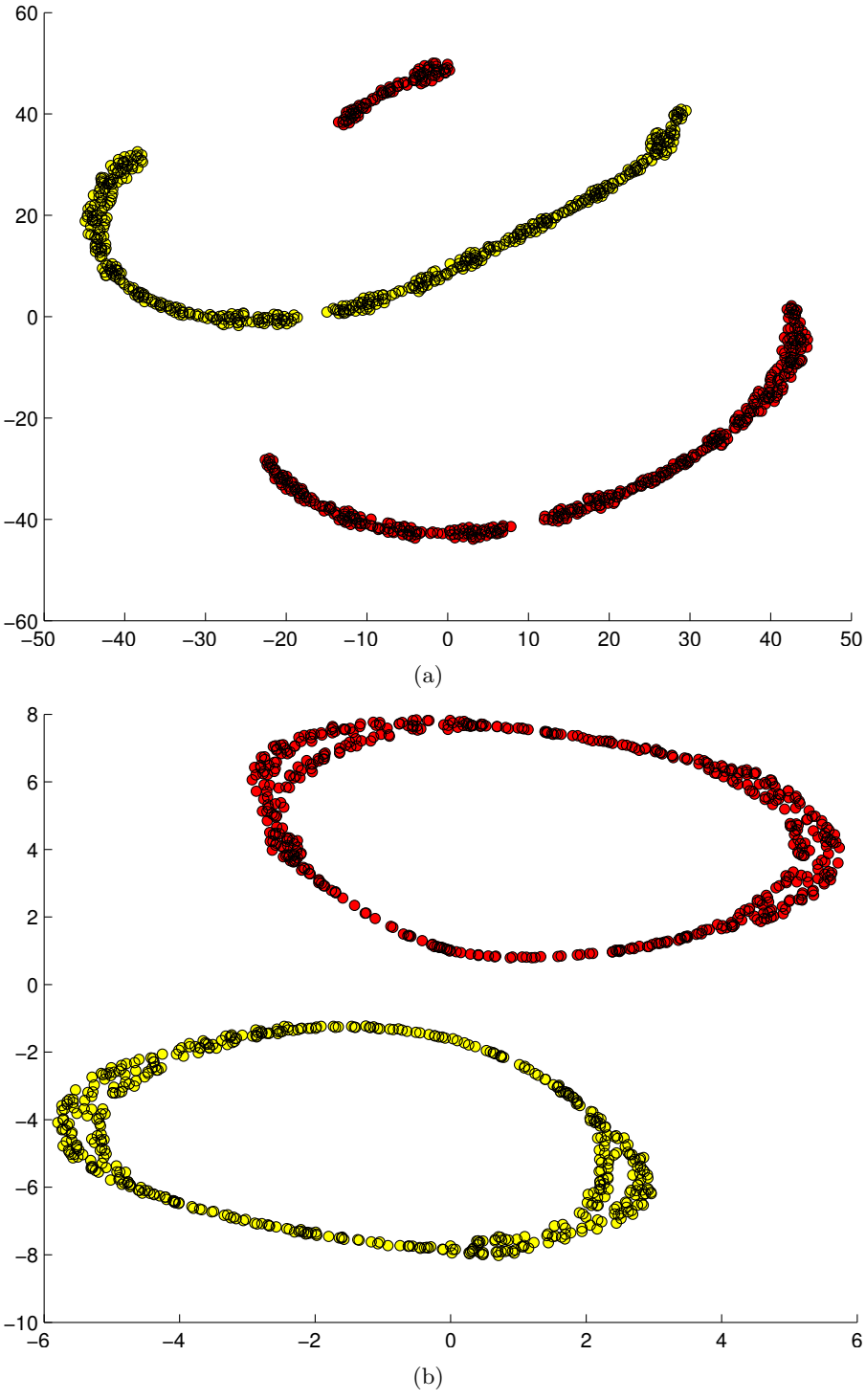


Figure 3.9: t-SNE [HR02] maps Chainlink data onto $\mathbb{R}^2$. (a) Disrupted clusters with default perplexity $k = 30$. (b) Correct mapping obtained by large perplexity $k = 200$.

will filter out the noise and reveal hidden structure. Based on the assumption that large variances have important structure, variance $b^\top Cb$ is to be maximized whereas basis vector b is a unit vector with $b^\top b = 1$.From the Lagrange multiplier follows $Cb = \lambda b$. So the PCA finds eigenvectors of the covariance matrix C of centralized vectorial data. The largest eigenvalues indicate the basis vectors of biggest variance.

However, the use of linear methods for construction of topographic mappings has several flaws. Linear mappings are restricted to real-valued data only. Linear mappings only capture the structure of linear manifolds within the data. Linear mappings might fail to reveal actual cluster structures, because they are not concerned with local structures of the data. See Figure 3.10 for illustration.

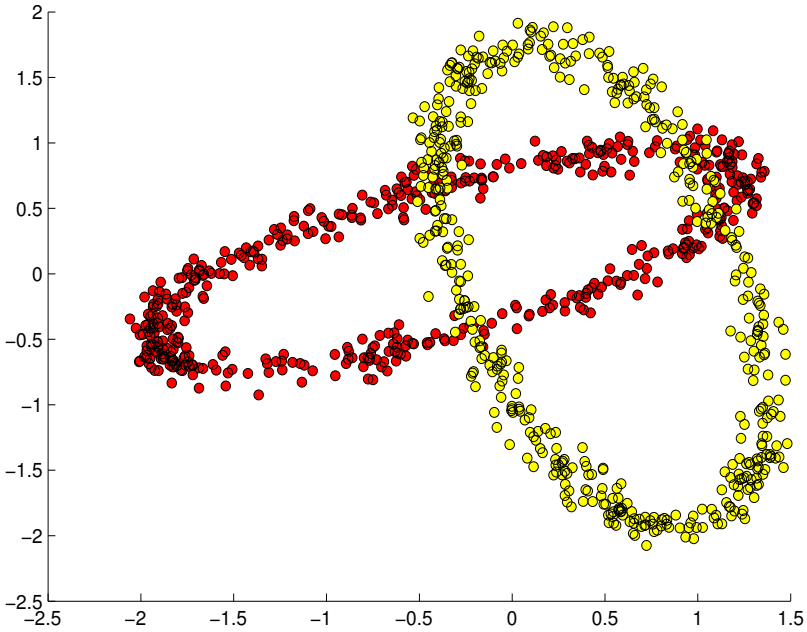


Figure 3.10: First two principal components of Chainlink data. Linear transformations, such as PCA and ICA, cannot unfold non-linear structures.

Kernel Methods

Methods based on the kernel trick have become increasingly popular during the last decade due to their unfolding abilities. Kernel trick [ABR64] means that any continuous, symmetric, positive semi-definite kernel function k can be expressed as inner product in a higher-dimensional space, such that $k(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ at which φ denotes the mapping into the higher-dimensional space. Using kernels, the function φ is never explicitly computed. A set of points cannot in general be linearly separated in few dimensions. In higher-dimensional spaces, especially with more dimensions than points, points can almost always be linearly separated. The kernel trick enables the compution of inner products in such spaces.

The Kernel PCA [SSM98] for instance is an extension to the original Principal Component Analysis. The covariance matrix in the high-dimensional space follows as $C = \frac{1}{n}\sum_{j=1}^{n}\varphi(x_j)\varphi(x_j)^\top$. Eigenvectors and Eigenvalues of C are to be found by means of kernel function k because φ is only needed in inner products. Popular functions are gaussian kernels $k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$ and polynomial kernels $k(x_i, x_j) = (x_i^\top x_j + 1)^d$. A major drawback of kernel methods is, however, the dependency from the chosen kernel, such that carefully parametrized gaussian kernels were observed to outperform linear kernels (cf. [FSSZ06] [GKCS08]).

Laplacian eigenmaps [BN03] are a dimensionality reduction technique based on nearest neighbour relations of a data set $\{x_1, ..., x_n\}$. Edge weights are derived from the distances between the corresponding data objects. The so-called Laplacian L is a symmetric, positive semidefinite matrix which can be thought of as an operator on functions defined on nodes of the proximity graph or on the matrix of pairwise distances, respectively The eigenfunctions of the Laplacian provide a natural basis for functions on the manifold. The eigenvectors of L are used for mapping the data in low-dimensional euclidean space. However, the resulting mapping highly depends on the chosen kernel function and neighbourhood relation and size.

Generative Topographic Mapping

The Generative Topographic Mapping (GTM) was proposed by Svensen [Sve98] as a probabilistic counterpart of Self-Organizing Maps. The GTM approach relies on a generative model that defines a relationship between data space and output space, such that

$$x = \gamma(o, W) + e$$

where e denotes some noisy error term, and $\gamma : \mathbb{O} \to \mathbb{D}$ is a product of basis function and weight vector for each data vector $\{x_1, ..., x_n\}$. The data vector x_i is assigned according to its posterior probability:

$$p(o_j|x_i, W) = \frac{p(x_i|o_j, W, \sigma) \cdot p(o_j)}{p(x_i|W)}$$

The conditional probability $p(x_i|o_j, W)$ usually is chosen as a Gaussian centered on $\gamma(o_j, W, \sigma)$ with variance σ. The prior probability $p(o)$ describes the (grid) structure of the output space, e.g. as sum of delta functions (cf. [Sve98]). To obtain W and σ the log likelihood

$$\mathcal{L}(W, \sigma) = \sum_{x_i \in \mathbb{X}} \ln\left(\frac{1}{|\mathbb{O}|}\sum_{o_j \in \mathbb{O}} p(x_j|o_j, W, \sigma)\right)$$

is maximized, e.g. using the EM-Algorithm [DLR77]. There is no critical annealing of a neighbourhood radius in the output space. However, parameter estimation of gaussian mixtures by means of EM algorithm [DLR77] is prone to produce suboptimal results. See Figure 3.11 for illustration.

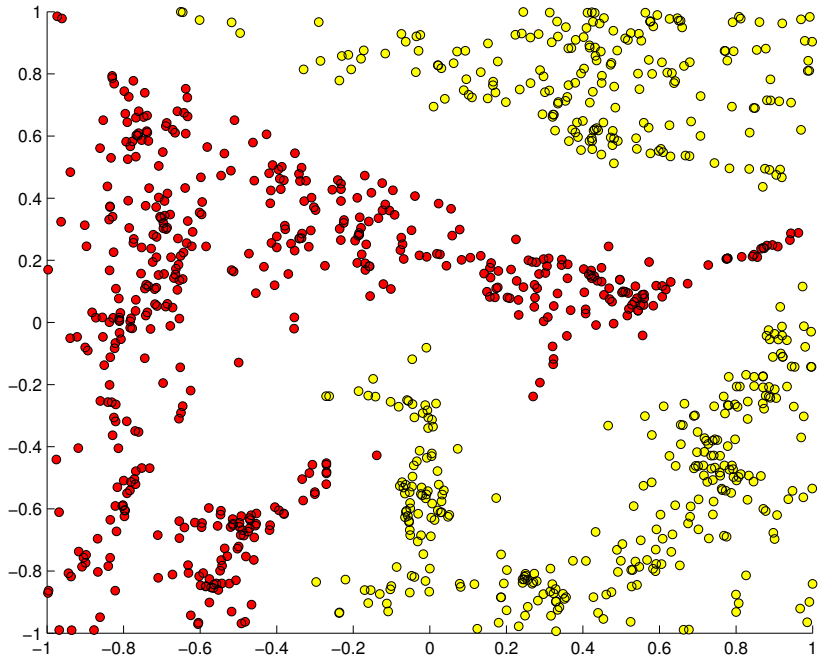


Figure 3.11: Chainlink data mapped onto $\mathbb{R}^2$ with GTM. The yellow class is disrupted.

Locally Linear Embedding

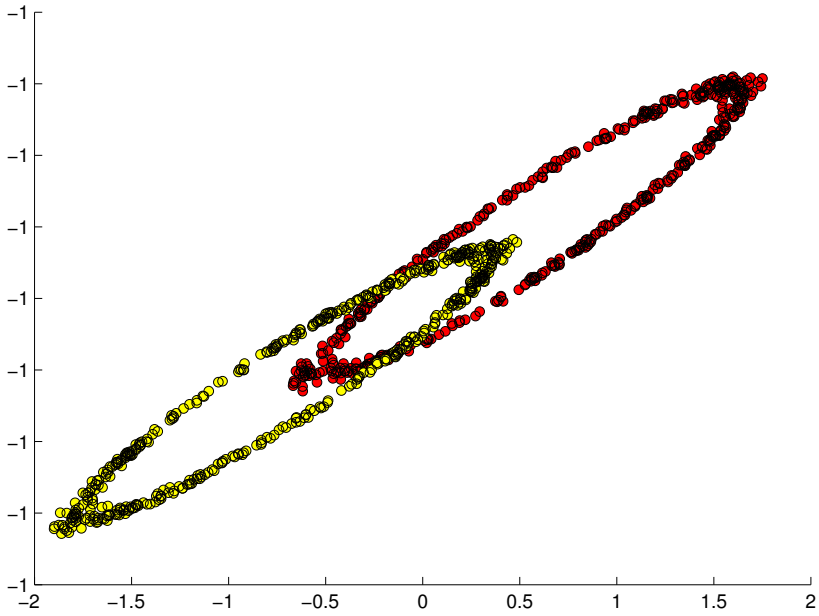
Locally Linear Embedding (LLE, [RS00]) assumes that each data vector $x_i \in \mathbb{X}$ and its neighbours lie on or close to a locally linear patch of a single manifold, such that x_i follows as linear combination $\sum_{j \neq i} w_{ij} \cdot x_j$ from its neighbours. Minimization of reconstruction error $\sum_i \left\| x_i - \sum_{j \neq i} w_{ij} \cdot x_j \right\|^2$ yields computation of coefficients w_{ij}. The mapping of $\mathbb{X} = \{x_1, ..., x_n\}$ into the low-dimensional output space is accomplished by choosing $\{y_1, ..., y_n\} \subset \mathbb{O}$ that minimizes $\sum_i \left\| y_i - \sum_{j \neq i} w_{ij} \cdot y_j \right\|^2$ using fixed weights w_{ij} as determined in data space.

LLE greatly emphasizes local (intra-cluster) structures and disregards global (intra-cluster) proximities. See Figure 3.12(a) for illustration. Furthermore it has been widely noticed that the obtained mapping heavily relies on the chosen neighbourhood size.

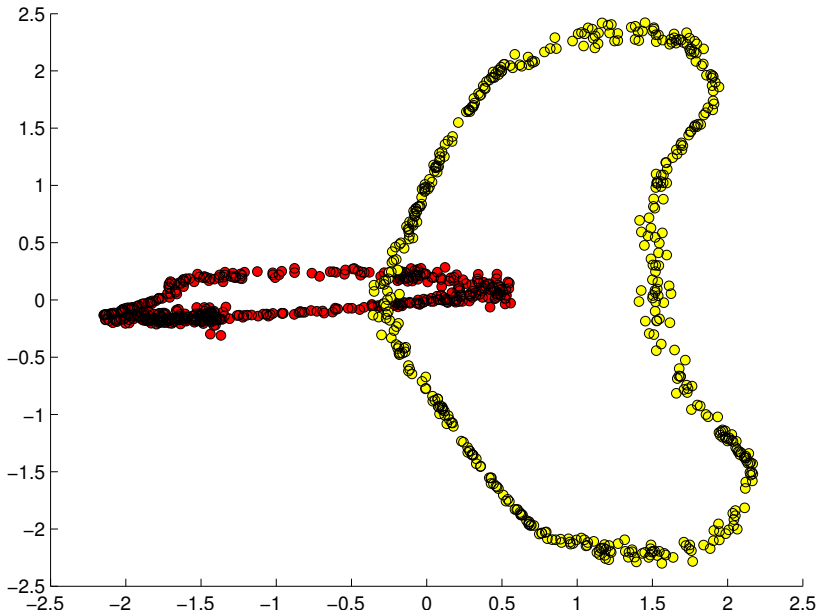
Isomap

Isomap [TdSL00] first estimates the pairwise geodesic distances between data objects $\{x_1, ..., x_n\}$, i.e. the shortest distances along the supposed data manifold. Then classical Multidimensional Scaling [Tor52] is applied to constructing an embedding $\{y_1, ..., y_n\}$ of the data objects in low-dimensional euclidean space that best preserves the estimated geodesic distances. Despite its unfolding capabilities, Isomap relies on a nearest-neighbour graph used for estimation of geodesic distances inside the assumed manifold. If such an graph is not chosen properly (or

cannot be found because of several submanifolds) the obtained mapping usually misrepresents the proximity of the data. See Figure 3.12(b) for illustration.



(a)



(b)

Figure 3.12: Chainlink data mapped onto $\mathbb{R}^2$. (a) Locally Linear Embedding [RS00] with 120 nearest neighbours. Clusters are falsely depicted as overlapping. (b) Isomap [TdSL00] with 120 nearest neighbours. Clusters are falsely depicted as overlapping.

Exploratory Morphogenesis

The Exploratory Morphogenesis (XOM,[Wis08]) is a particularly interesting learning method that inverts the data processing as known from Self-Organizing Maps, i.e. Online-SOM algorithm [Koh89]. Instead of vectors from data space, the XOM method relies on uniformly drawn samples $o \in \mathbb{O} \subset \mathbb{R}^2$ from the output space. The mapped data objects are moved in output space by means of an inversed Online-SOM update rule

$$y_i \leftarrow y_i + \alpha \cdot F_\sigma(d_{\mathbb{D}}(x_i, x(o)) \cdot (o - m(x))$$

at which $x(o) = \arg\min_{x_j \in \mathbb{X}} d_{\mathbb{O}}(o, y_j)$ denote the nearest neighbour in output space. The XOM method produces topographic mappings, i.e. the distance structure of data set $\mathbb{X}$ is to be retrieved directly from the mapping. However, mappings obtained from XOM suffer from the impossibility of distance preservation. Global structures are preferred over local ones. See Figure 3.13 for illustration.

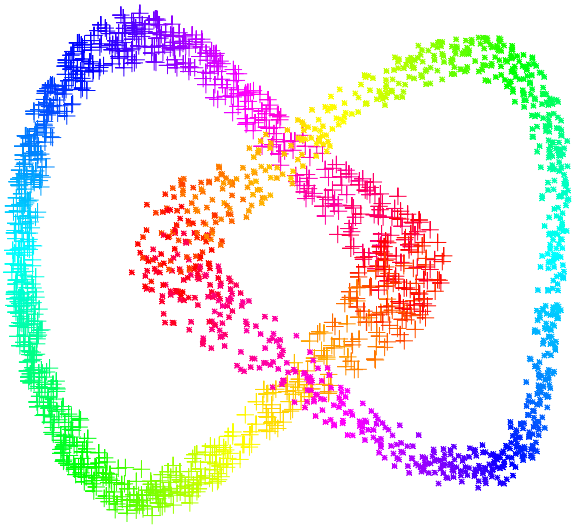


Figure 3.13: XOM misleadingly shows overlapping rings. Picture from [Wis08].

Schelling's Segregation Model

Schelling's segregation model [Sch69] is one of the first swarm intelligence algorithms that aims at the preservation of a simple topology. Two types of agents reside on a two dimensional grid. The agents have a limited tolerance for living next to agents of the other type. An agent with too much stress, i.e. too many opposite-type neighbours, is allowed to jump randomly into a free grid space. Schelling's model led to a segregation process of agents, even when individual agents had only a moderate bias against living near agents of the opposite type. Originally the model was intended to explain of how racialized city ghettos might emerge from individual choices, given even slight racial biases. Some important constraints on effective segregation have been described by Vinkovic and Kirman [VK06]. Segregation is greatly increased if agents are allowed to jump to any node that yields less

stress, instead of neighbouring nodes only. This model suggests that fixed point iteration leads to separation of two (or more) clusters.

Vinkovic and Kirman [VK06] have explained the behaviour of Schelling's model on basis of physical surface tension forces. According to this, agents of the same class correspond to coalescing liquid particles. Increased mobility of agents facilitates class-wise segregation. For illustration see Figure 3.14.

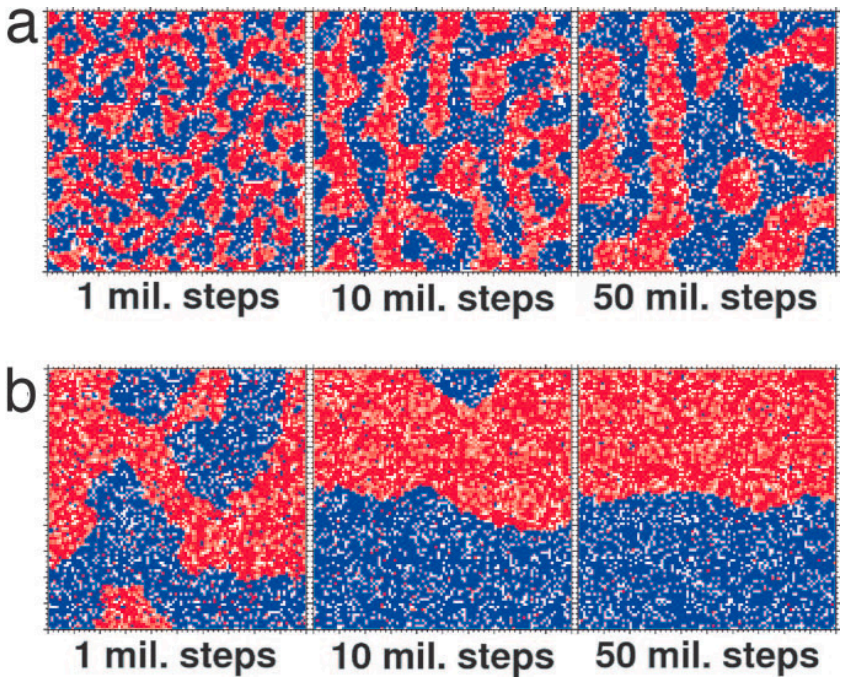


Figure 3.14: Schelling's model with blue and red agents on a planar grid [VK06]. Empty nodes are depicted as white. (a) Only jumps to the nearest acceptable location are allowed. Agents diffuse very slowly. (b) Agents can jump to any location. Diffusion is increased because empty locations are not required for movement.

DataBots

Databots have been proposed in 1999 [Ult00a]. In contrast to the pick and drop of ABC where agents and data are considered different, Databots are identified with single data objects. The Databots are able to move on a two-dimensional discrete grid which is finite but unbound, i.e. toroid [Ult03]. The movement program of the Databots is controlled by a hierarchy of programs for walking. These include, for example, random walk, directional inertia and attractive and repulsive forces. The forces are proportional to the (dis-)similarities of neighbouring Databots and, respectively, data objects. To our experience, cluster formation in this model depends critically on the formulation of these forces. The formation of a topographic mapping depends on the annealing scheme for his threshold. Map formation turned out to be more stable and correct if a substantial amount of random walk is included in the movement programs [Ult00a].

Projection Pursuit

The Projection Pursuit [FT74] is a statistical method that aims at reduction of high-dimensional data in order to uncover "interesting" structures hidden in the data. To achieve this, a hyperplane for instance is sought by which to project the data onto, i.e. by linear transformations. A so-called *index function* measures the degree to which a desired property is revealed. For example, the *mean distance between nearest neighbours* is a common test statistic for clustering in two dimensions. Then the choice for the linear transformation becomes an optimization problem for an optimal index value. The obtained projection is determined by the chosen index function.

3.7 Cohesive Mappings for Visualization

Scatter Plots

Scatter plots are diagrams that depict the mapping's image $\{y_1, ..., y_n\}$ as point clouds in the output space. There is no further information visualized, such as neighbourhood relations or distances. Cluster structures and neighbourhood relations are to be retrieved from the drawn points only. Therefore, naive plots may be topographic or topological according to the nature of the underlying mapping $m : \mathbb{X} \to \mathbb{O}$. See Figure 3.15 for illustration.

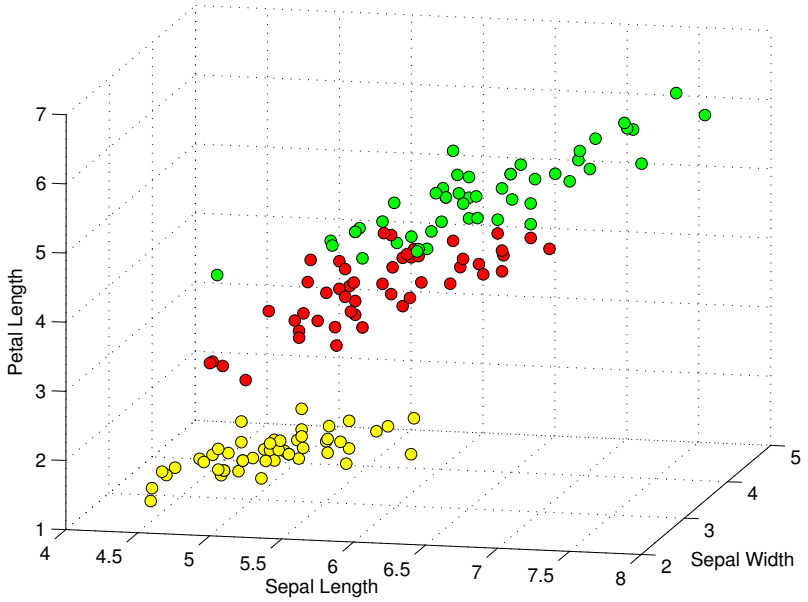


Figure 3.15: Scatterplot of Fisher's Iris data [Fis36] in output space $\mathbb{R}^3$.

Distance Maps

Self-Organizing Maps (SOM) are commonly visualized using distance maps. Distance maps $u : \mathbb{O} \to \mathbb{R}_0^+$ depict distances of the data space on top of the output space $\mathbb{O}$. The most popular method is the U-Matrix (unified distance matrix) proposed by Ultsch [US90]. The U-Matrix method visualizes the average distances between codebook vector w_i and its immediate neighbours $N(i) = \{w_j \mid o_j \text{ neighbouring } o_i\} \subset \mathbb{O}$.

$$u(o_i) = \frac{1}{|N(i)|} \sum_{j \in N(i)} d_{\mathbb{D}}(w_i, w_j)$$

These so-called *U-Heights* are typically interpreted as height values of a discrete landscape. In analogy to geographic maps, U-heights are depicted by coloring grid nodes according to $u : \mathbb{O} \to \mathbb{R}_0^+$. See Figure 3.16 for illustration. The density of the SOM's codebook vectors roughly follows the probability density function of the data. This means that codebook vectors' neighbour-distances are approximately inversely proportional to the density of the data. Thus, cluster borders can be identified as *mountains* of high distances separating *valleys* of low distances. Proposed by Vesanto and Sulkava [VS02], the so-called *Distance-Matrix* alters the U-Matrix method by applying the median instead of arithmetic mean.

The U-Map method [UM06] is a generalization of the U-Matrix for topographic mappings that are not based on the SOM architecture. Codebook vectors are derived by the help of the SOM algorithm on top of a given, fixed topographic mapping. The U-Matrix then depicts the distances among these codebooks.

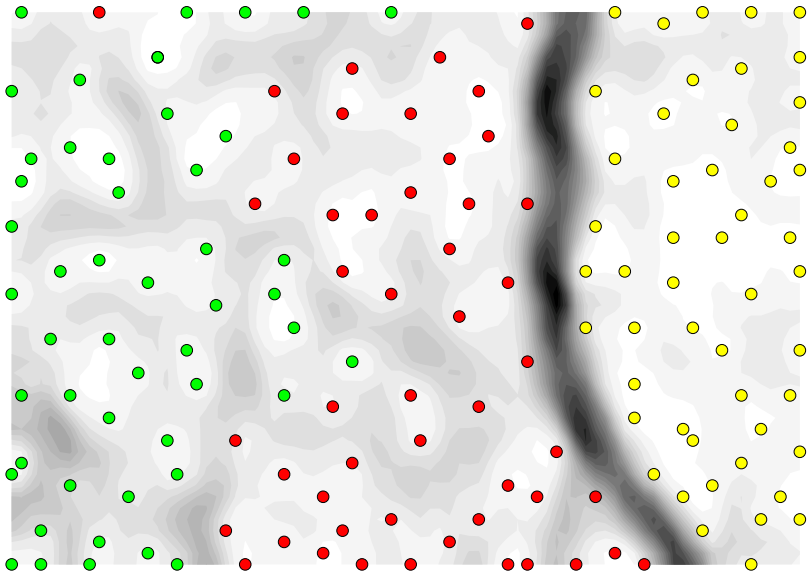


Figure 3.16: U-Matrix depicts distance structure of Fisher's Iris data [Fis36] as gray scales.

Density Maps

Density maps $p : \mathbb{O} \to \mathbb{R}_0^+$ depict the data space density on top of grid $\mathbb{O}$. This means at the position of each node $o_i \in \mathbb{O}$ the supposed density at its preimage in data space is displayed. The P-Matrix by Ultsch [Ult03] uses SOM as a basis for density estimation. On top of grid node $o_i \in \mathbb{O}$ the density estimates in data space measured at w_i are depicted as heights and colors, respectively. The P-height is formally defined as

$$p(o_i) = \frac{1}{\sigma n} \sum_{j=1}^{n} I\left(d_{\mathbb{D}}(w_i, x_j) - \sigma\right)$$

with bandwidth $\sigma \in \mathbb{R}^+$ and Heaviside function $I : \mathbb{R} \to \{0, 1\}$.

Smoothed Data Histograms [PRM02] use the nodes of SOM grids as bins of a rank-based histogram. The bin centers in the data space are defined by the codebook vectors w_i and the varying bin widths are defined through the distances between the codebooks. The membership degree of a data object x to a specific bin is calculated based on the rank of the distances between x and all bin centers w_i. The SDH method has no need for a radius that is to be determined in the data space. Instead, a discrete smoothing parameter is easily determined according to the size of grid $\mathbb{O}$.

Topographic Mappings for U-Matrix

So far it was unknown which sort of topographic mappings actually are adequate for visualization with the U-Matrix method. Again, let $\mathbb{X} = \{x_1, ..., x_n\}$ denote the set of high-dimensional data objects with images $\{y_1, ..., y_n\}$ on a low-dimensional grid. For the sake of simplicity, the learning algorithm of Batch-SOM is used with finite neighbourhood function F_σ, such that $F_\sigma(\delta) = 0$ for all $\delta > \sigma$. The mapped objects $\{y_1, ..., y_n\}$ induce a Voronoi tesselation of grid $\mathbb{O}$. See Figure 3.17 for illustration. This is evident due to the following considerations.

Definition 3.12 *Let* $n(o_i) = \{x_j \in \mathbb{X} \mid \forall x_k \in \mathbb{X} : d_{\mathbb{O}}(o_i, y_j) \leq d_{\mathbb{O}}(o_i, y_k)\}$ *denote the set of mapped nearest neighbours for each node* o_i *in output space.*

Definition 3.13 *Let* $V(x_j) = \{o_i \in \mathbb{O} \mid x_j \in n(o_i)\}$ *denote the discrete Voronoi cell for each data object* $x_j \in \mathbb{X}$.

Definition 3.14 *Each node* $o_i \in \mathbb{O}$ *is called* bordering *iff there exists an immediately neighbouring node* o_j *belonging to another Voronoi cell, that is* $n(o_i) \neq n(o_j)$.

Definition 3.15 *An annealing scheme sequence* $\mathcal{A} = (\sigma_1, ..., \sigma_t) \in \mathbb{R}^t$ *is called* sufficiently slow *iff each distance on grid* $d_{\mathbb{O}}(o_i, o_j)$ *for* $\{o_i, o_j\} \subset \mathbb{O}$ *occurs in* $\mathcal{A}$.

Thus Voronoi cells are analogously defined in discrete and continuous spaces. Pairs of such cells $V(x_i), V(x_j)$ may be identical, disjoint or adjacent (i.e. sharing common nodes at their border) . Finally, we will show that the U-Matrix method tends to reproduce the Voronoi tesselation in output space when applied on Self-Organizing Maps.

Theorem 3.16 *For Batch-SOM with finite neigbourhood functions and sufficiently slow annealing scheme holds:* $u(o_i) = 0$ *for each non-bordering node* $o_i \in \mathbb{O}$.

Proof

1. Node $o_i \in \mathbb{O}$ is non-bordering.
2. Let o_j be an immediately neighbouring node of o_i.
3. From definition 3.14 then follows: $n(o_i) = n(o_j)$.
4. The Batch-SOM algorithm uses a sufficiently slow annealing scheme.
5. Due to definition 3.15 then follows: for nodes o_i, o_j exists a smallest $\sigma \in \mathcal{A}$ with $F_\sigma(d_\mathbb{O}(o_i, y_k)) > 0$ and $F_\sigma(d_\mathbb{O}(o_j, y_k)) > 0$ and $F_\sigma(d_\mathbb{O}(o_i, y_l)) = F_\sigma(d_\mathbb{O}(o_j, y_l)) = 0$ for all $x_k \in n(o_i), x_l \in \mathbb{X} \setminus n(o_i)$.
6. The codebook vectors are $w_i = w_j = \frac{\sum_{x \in n(o_i)} x}{|n(o_i)|}$ due to Formula 3.6.
7. The codebook vectors' distance is $d_\mathbb{D}(w_i, w_j) = 0$ for all immediately neighbouring nodes o_j.
8. The U-height follows from Formula 3.7.

□

From Theorem 3.16 follows that the U-heights' distribution approximates the borders of Voronoi cells (i.e. a tesselation) in case of sufficiently slow annealing schemes. Nodes on the Borders between two Voronoi cells $V(x_i), V(x_j)$ usually have U-Matrix heights in $[0, \frac{d_\mathbb{D}(x_i, x_j)}{2}]$, depending on the exact geometrical configuration of cells. For illustration see Figure 3.17. Faster annealing schemes blur the borders of Voronoi cells, which leads to the familiar appearance of U-Matrices (cf. Figure 3.16). Obviously, a cluster in data space can be retrieved in output space by means of the U-Matrix and related methods, iff its Voronoi cells in output space are neighbouring. For non-finite neighbourhood functions F_σ the statement of Theorem 3.16 is likewise obtained. by means of the limit of $d_\mathbb{D}(w_i, w_j)$ as σ approaches 0.

1. Let $F_\sigma(x) = e^{-\frac{x^2}{2\sigma^2}}$ be the Gaussian focus funcion.
2. Node $o_i \in \mathbb{O}$ is non-bordering with immediately neighbouring $o_j \in \mathbb{O}$
3. From definition 3.14 follows $n(o_i) = n(o_j)$.
4. For $x_k \in n(o_i)$ let $\delta_{ik} = d_\mathbb{O}(o_i, y_k)$ and $\delta_{jk} = d_\mathbb{O}(o_j, y_k)$ denote the distances towards the nearest images y_k.
5. For farther images with distances $\delta > \delta_{jk}$ holds $\lim_{\sigma \to 0} \frac{F_\sigma(\delta_{jk})}{F_\sigma(\delta)} = \infty$
6. From weighted sum in Formula 3.6 follows $\lim_{\sigma \to 0} w_j = \frac{\sum_{x \in n(o_j)} x}{|n(o_j)|}$

7. For farther images with distances $\delta > \delta_{ik}$ holds $\lim_{\sigma \to 0} \frac{F_\sigma(\delta_{ik})}{F_\sigma(\delta)} = \infty$

8. From weighted sum in Formula 3.6 follows $\lim_{\sigma \to 0} w_i = \frac{\sum_{x \in n(o_i)} x}{|n(o_i)|}$

9. According to definition 3.14 the limit of $d_{\mathbb{D}}(w_i, w_j)$ becomes 0 as σ approaches 0.

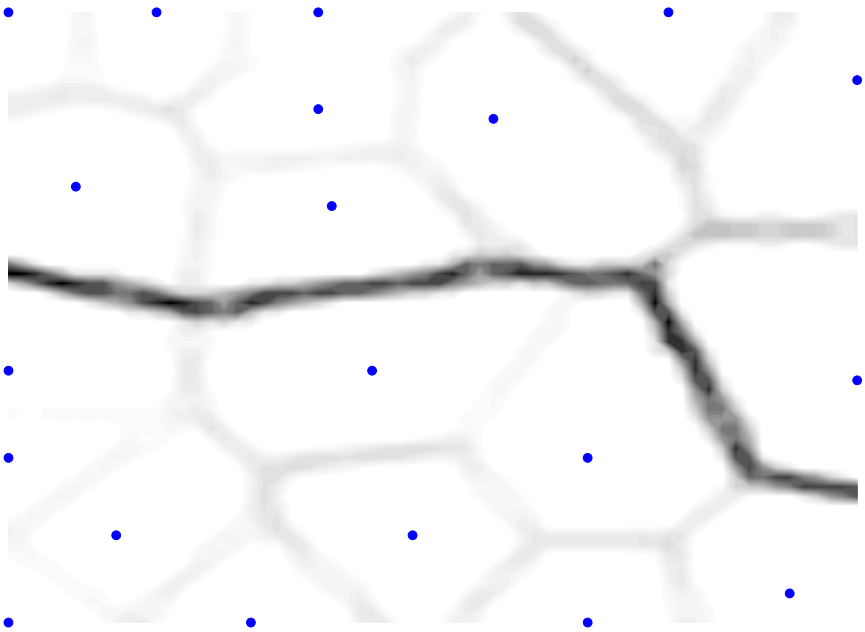


Figure 3.17: U-Matrix heights' distribution reproduces a low-dimensional Voronoi tesselation. Dots depict mapped data points. Shades of gray indicate distances between data objects. Zero U-heights are white.

As shown in Theorem 3.16 the U-Matrix method approximately depicts the Voronoi tesselation of the output space. It is evident that for each meaningful class $C \subset \mathbb{X}$ the U-Matrix method can only depict the concerning inner-class distances iff the image of C is not disrupted in output space, i.e. all Voronoi cells of C are connected. From the latter insights follows a crucial definition. The dual of the Voronoi tesselation is the Delaunay graph (cf. Section 2.2).

Definition 3.17 *Let $\mathcal{MD} \subset \mathbb{X} \times \mathbb{X}$ denote the Delaunay graph determined by $\{y_1, ..., y_n\}$ such that $(x_i, x_j) \in \mathcal{MD}$ iff Voronoi cells $V(x_i), V(x_j)$ are adjacent in output space.*

Definition 3.18 *A topographic mapping* cohesively *maps class C on the output space iff the relevant subgraph of $\mathcal{MD}$ is connected, i.e. $\mathcal{MD} \cap C \times C$ is connected.*

A topographic mapping $m : \mathbb{X} \to \mathbb{O}$ is refered to as *cohesive* if it cohesively preserves each class of $\mathbb{X} = \{x_1, ..., x_n\}$ in output space. From Theorem 3.16 follows that cohesiveness is a minimum requirement for visual cluster analysis. Cluster boundaries are depicted by means of large U-heights. Images of data vectors with small U-heights inbetween are supposed to indicate inner cluster structures, even though the images may be located faraway. This means that the scale of mappings' images can be disregarded when using U-Matrix methods for visual cluster analysis.

3.8 Conclusions

Topographic mapping methods are classified as focusing or non-focusing with respect to the parametrized concept of neighbourhood that is applied for construction of correct topographic mappings. Focusing methods are based on an annealing scheme, whereas non-focusing methods usually rely on a predefined concept of neighbourhood. In both cases, mismatching parametrization causes misrepresentations of cluster structures. It turns out that trustworthy visualization of cluster structures by means of U-Matrix like techniques relies on *cohesive* topographic mappings. For a brief summary see Table 3.1.

Method	Mapping	Learning
Batch-SOM	topological	focusing
CCA	topographic	focusing
XOM	topographic	focusing
Databots	topological	focusing
GTM	topological	focusing
ABC	topological	non-focusing
SNE	topological	non-focusing
PCA, ICA	topographic	non-focusing
Kernel PCA	topographic	non-focusing
LLE	topographic	non-focusing
Isomap	topographic	non-focusing

Table 3.1: Summary of topographic learning methods. Topographical mappings are supposed to display the pairwise dissimilarities of input data. Topological mappings depict neighbourhood relations of input data, whereas scale is largely disregarded.

Chapter 4

Swarm-Organized Mappings

In this chapter, two novel method for visual cluster analysis purposes are introduced. The Swarm-Organized Projection (SOP) method combines elements from the fields of swarm intelligence and artificial neural networks into a self-adaptive topographic mapping method. The Swarm-Organized Quantization (SOQ) is a derivative of SOP for vectorial spaces in order to decrease the computational complexity.

4.1 Inspiration

Ant-Based Clustering & Databots

As outlined in Section 3.4, the Ant-Based Clustering Method (ABC) proposed by Lumer/Faieta [LF94] usually produces too many and too small piles of data objects. A structural inability to form large piles of coherent objects decreases the method's suitability for cluster analysis of data. Despite its weaknesses, ABC is a promising technique that enables unsupervised machine learning on continuously changing data, e.g. data streams and incremental financial data. Therefore, a revision of ABC method aims at overcoming the limitations concerning topographic mapping quality. As in Databots (cf. Section 3.6) agents are to be identified with data objects in order to avoid an additional search for scattered data objects.

Self-Organizing Maps

The Self-Organizing Batch Map (Batch-SOM) faces a parametrization problem with respect to the annealing scheme applied to its neighbourhood radius. Nybo et al. [NVK07] have indicated the vital influence of annealing schemes on the obtainable topographic quality. However, an adequate annealing scheme for a given data set is an unknown quantity in beforehand. Despite its weaknesses, Batch-SOM promise sufficiently cohesive mappings of high-dimensional data due to a self-organizing process based on interacting neurons. The SOM clearly outperforms the ABC method by Lumer/Faieta in terms of topology preservation. Therefore, a revised Batch-SOM aims to overcome the annealing problem.

Schelling's Model and its Physical Analogue

Schelling's segregation model [Sch69] is one of the first Swarm-Intelligence algorithms that aims at the preservation of discrete metrics (cf. Section 3.6). Vinkovic and Kirman [VK06] have shown how an increas of agents' mobility facilitates class-wise segregation. Despite their discrete nature, Schelling's agents offer a simple yet effective way to cause segregation of objects according to pairwise dissimilarities. This mechanism is to be combined with the topology-preserving utility functions found in ABC and Batch-SOM in order to derive a novel swarm-intelligence method for topographic mapping purposes.

4.2 Swarm-Organized Projection

The Swarm-Organized Projection (SOP) method provides an algorithm that realizes topographic mapping for dissimilarity data, i.e. pairwise dissimilarities are available but vector-space axioms are not required. Again, let $\mathbb{O} \subset \mathbb{N}^2$ denote the regular finite grid that acts as output space. A probability distribution $p : \mathbb{O} \to [0, 1]$ is centered around $o_i \in \mathbb{O}$ iff $d_{\mathbb{O}}(o_i, o_j) \leq d_{\mathbb{O}}(o_i, o_l) \Leftrightarrow p(j) \geq p(l)$ holds for all $o_j, o_l \in \mathbb{O}$, i.e. the probability increases with shrinking distance toward the center. See Figure 4.1 for illustration.

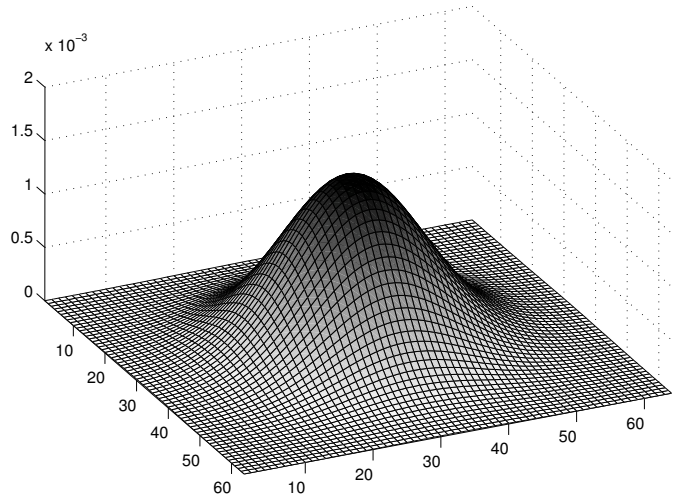


Figure 4.1: Probability distribution of gaussian shape centered around node $(32, 32) \in \mathbb{O}$.

Definition 4.1 *The set of* sensorial samples $\mathcal{N}(p, k, o) = \{o_1, ..., o_k\} \subset \mathbb{O}$ *contains* $k \in \mathbb{N}$ *locations in output space that are randomly selected according to the discrete probability distribution* $p : \mathbb{O} \to [0, 1]$ *centered around* $o \in \mathbb{O}$.

Sensorial samples $\mathcal{N}_0(p, k, o)$ with an umbilicus additionally contain $o \in \mathbb{O}$ such that $\mathcal{N}_0(p, k, o) = \{o, o_1, ..., o_k\} \subset \mathbb{O}$. The definiton of $\mathcal{N}$ is not compatible with traditional set theory, but can be thought of as a three-ary function with an additional,

hidden argument that modifies contents of $\mathcal{N}$ on each evaluation. Random-walking ants, as used for Ant-Based Clustering (ABC), illustrate the intended purpose of sensorial samples. The presence of a random-walking ant placed at the center of an infinite grid follows a binomial distribution. For large numbers of steps, the binomial distribution approximates the normal distribution. Here, sensorial sampling is based on discretized normal probabilities centered around the umbilicus. The main purpose of $k \in \mathbb{N}$ sensorial samples is to decrease the computational effort of search-based techniques. In contrast to that, Self-Organizing Maps rely on an exhaustive search over the whole output space.

Algorithm

The Swarm-Organized Projection (SOP) algorithm operates on a finite data set $\mathbb{X} = \{x_1, ..., x_n\}$ with pairwise dissimilarities $d_{\mathbb{D}} : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$. Each object $x_i \in \mathbb{X}$ is identified with an agent, i.e. the number of agents corresponds to the cardinality of $\mathbb{X}$. Mapping $m : \mathbb{X} \to \mathbb{O}$ reflects the agents' current locations, which are denoted with $\{m(x_1), ..., m(x_n)\} = \{y_1, ..., y_n\}$. An agent moves to a another grid node in its sensorial samples iff its topographic stress $\Phi : \mathbb{X} \times \mathbb{O} \to \mathbb{R}_0^+$ becomes smaller. The topographic stress Φ is the weighted sum of dissimilarities toward neighbouring objects (see Formula 4.2), at which $F_\sigma \circ d_{\mathbb{O}}$ realizes a neighbourhood function by means of focus F_σ. On each iteration all agents are allowed to move simultaneously. An epoch ends if no agent has caused an update, i.e. $\{y_1, ..., y_n\}$ did not change. The global learning radius σ is decreased after each epoch. σ_{max} is the maximum distance of map space. The algorithm ends iff the smallest possible radius 1 is reached. See Algorithm 4.1 for details.

Formula 4.2

$$\Phi(x, o) = \frac{\sum_{i=1}^{n} F_\sigma(d_{\mathbb{O}}(y_i, o)) \cdot d_{\mathbb{D}}(x_i, x)}{\sum_{i=1}^{n} F_\sigma(d_{\mathbb{O}}(y_i, o))}$$

Algorithm 4.1 Swarm-Organized Projection

```
1: function SWARMPROJECTION({x_1, ..., x_n})
2:     randomize {y_1, ..., y_n}
3:     for σ ← σ_max, σ_max − 1, ..., 1 do
4:         p ← gaussian(σ)
5:         repeat
6:             for i ← 1, ..., n do
7:                 y_i ← arg min_{o ∈ N_0(p,1,y_i)} Φ(x_i, o)
8:             end for
9:         until {y_1, ..., y_n} fix
10:    end for
11: end function
```

First, the agents are randomly located on the grid (cf. Line 2). Then, for each radius σ a fixed point iteration with respect to mapping $\{y_1, ...y_n\}$ is performed

(cf. Lines 5 to 9). Agents move simultaneously iff they can decrease their personal amount of topographic stress (cf. Line 7). The set of sensorial samples $\mathcal{N}_0(p, 1, y_i)$ consist of two elements, i.e. umbilicus and another random node. Obviously, the umbilicus is used to realize a memory for agent's present topographic stress. The random node is used to sense the agent's environment. See Figure 4.2 for illustration of SOP learning the Chainlink data.

Formally, a fixed point refers to a configuration where no agent is able to move to another location with less topographic stress in terms of Φ. Due to sparse probabilistic movements of agents, a swarm-intelligence setup offers no meaningful reference framework for fixed points in a strict mathematical sense. To meet these concerns, the definition of fixed points is altered.

Definition 4.3 *Mapping* $m : \mathbb{X} \to \mathbb{O}$ *is called* fixed *iff the number of immobile agents does not exceed a given threshold for a given number of iterations, i.e. the agents' locations* $\{y_1, ..., y_n\}$ *were not modified by chance.*

Adaptive Annealing

As known from Batch-SOM (cf. Section 3.2) and CCA (cf. Section 3.3) the annealing of learning radius σ is crucial for the quality of the obtainable topographic mapping. The choice of the annealing scheme is usually left to some default strategy of a particular implementation, e.g. linear decrease from initial to final value. An optimal annealing scheme depends on the structure of the output space (e.g. grid size, periodic boundary conditions, shape of neighbourhoods) and the structure of the data set. The latter is, however, an unknown quantity. A wrong choice of the annealing strategy leads to severe misrepresentation of the data space's topography resulting in a faulty representation of clusters. See Figures 3.5 and 3.6 for illustration.

The annealing in SOP adapts itself to the topographical structures of a data set. A central feature of SOP is the fixed point iteration. For each neighbourhood radius σ, the agents' locations are adapted until a fixed point is reached. Different values of σ lead to different amounts of agents' movements on different data sets. This adaptive mechanism discards the need for a rigid annealing scheme. The fixed point iteration of the SOP algorithm leads to a self adaptation of the number of iterations for each value of σ. In Figure 4.3 the number of iterations that SOP used for each σ on Chainlink data is measured. Starting with random mappings, dense clusters of agents emerge during the training process. It can be seen that initially (a), when the mapping is random, many iterations are necessary to reach convergence. As soon as a raw topographic mapping is achieved, i.e. $\sigma \in \{91, ..., 19\}$, the radius decreases rapidly. For Chainlink data, the decrease is almost linear (b). In general, this depends on the scaling of the structural features of the data. When the global (cluster-specific) features were recognized ($\sigma \leq 19$) a fine grained optimization within the clusters takes place (c-d). In this phase all the map space is covered by the SOP agents. As can be seen that the distance structure of the data set determines which radius σ leads to disintegration of spurious clusters. For SOP no prior knowledge of the structural features is necessary to determine a suiting annealing scheme.

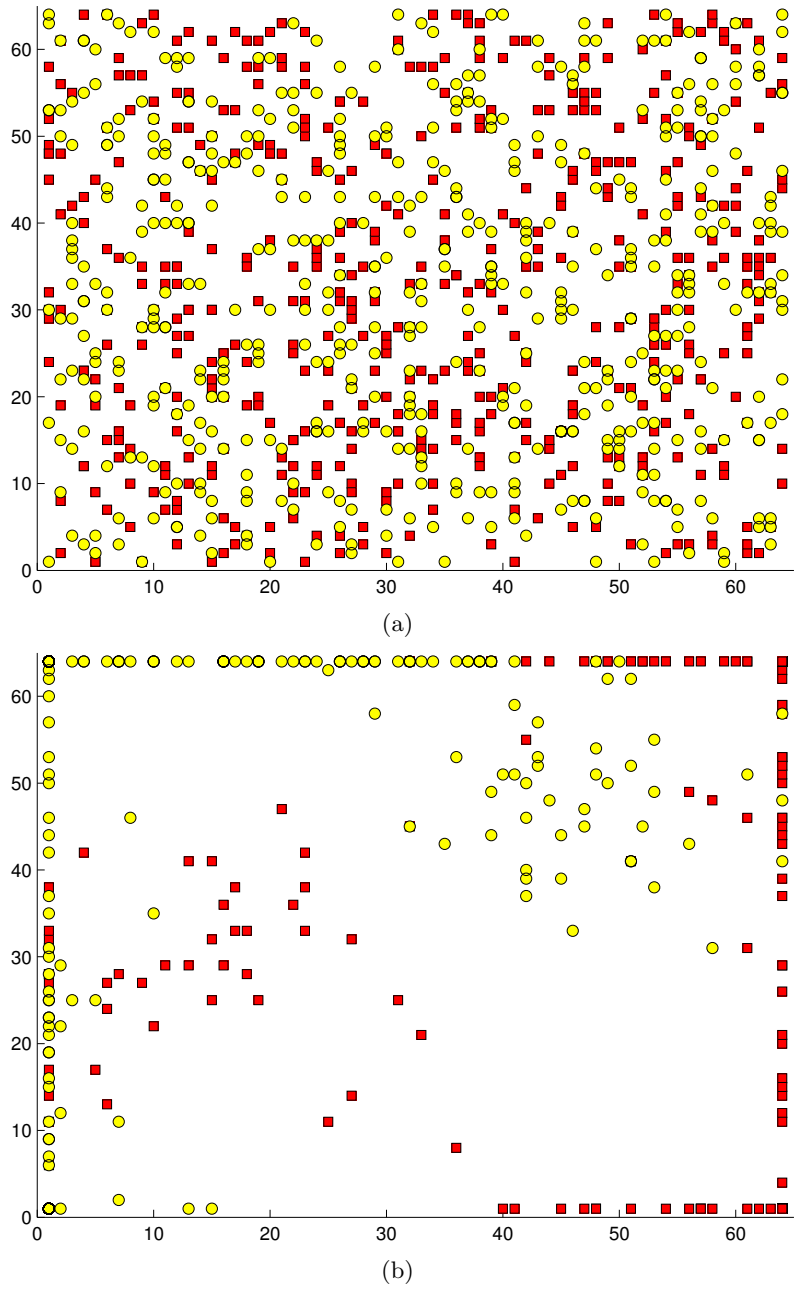
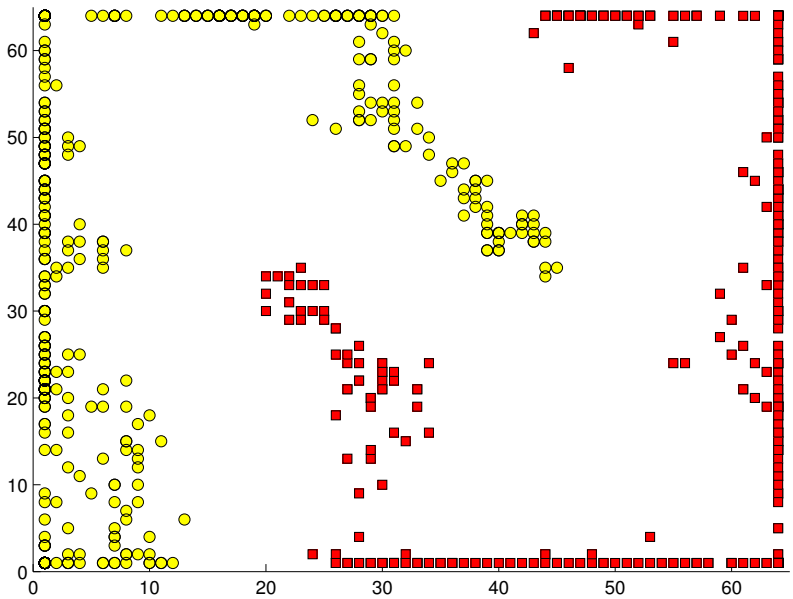
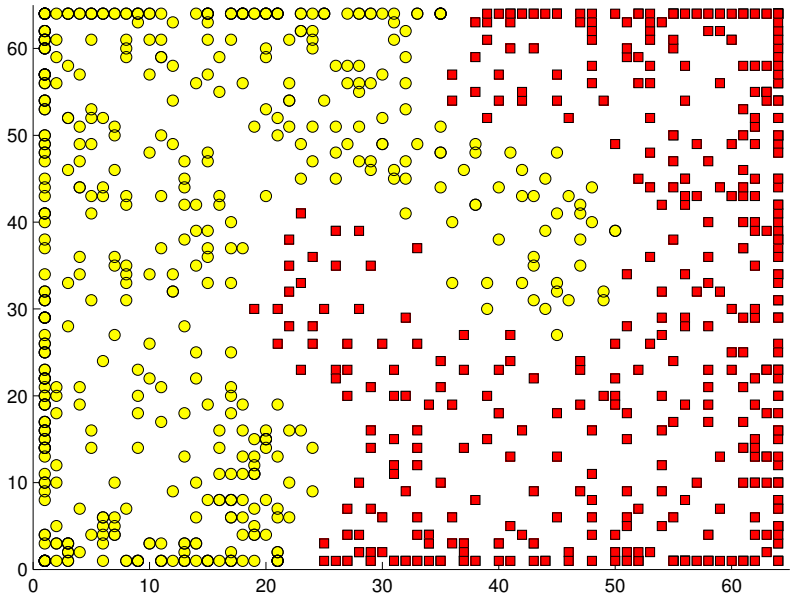


Figure 4.2: SOP maps Chainlink data onto planar 64×64 grid. Radius σ decreases adaptively from 91 to 1. (a) Random initialization, $\sigma = 91$. (b) Overlapping piles have emerged at borders, $\sigma = 19$.

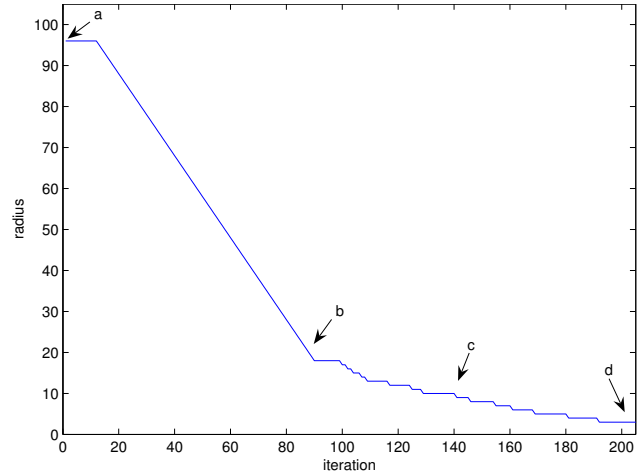


(c)

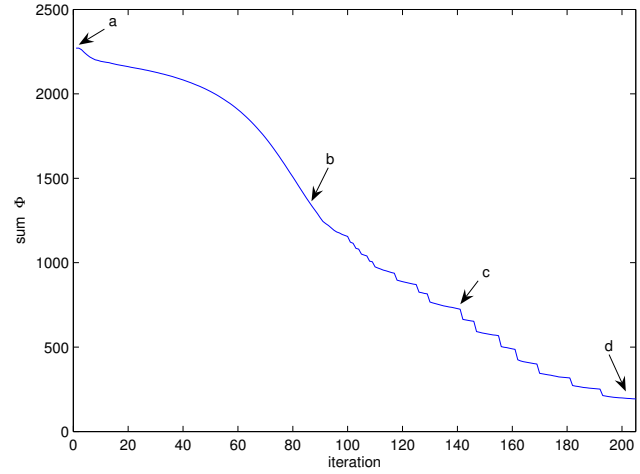


(d)

Figure 4.2: SOP maps Chainlink data onto planar 64×64 grid. Radius σ decreases adaptively from 91 to 1. (c) Resolved overlap at $\sigma = 7$. Two well separated clusters have emerged. (d) Uniformly distributed classes have emerged at $\sigma = 1$.



(a)



(b)

Figure 4.3: Adaptive annealing: (a) neighbourhood radius σ as a function of iterations. (b) $\sum_i \Phi(x_i, y_i)$ as a function of iterations. SOP maps Chainlink onto planar 50×82 grid. Parts of Figure 4.2 are marked: a-b: long-lasting ordering of agents, b-c: overlapping rings unfold, c-d: agents uniformly spread.

Cardinality of $\mathcal{N}_0$

The motion of agents is governed by the topographic stress function Φ and the cardinality of sensorial samples $\mathcal{N}_0$, i.e. the number of evaluations of Φ. Small cardinalities relate to random-walking ants during a small time interval. Large cardinalities relate to random-walking ants during a large time. In order to determine an appropriate cardinality for $\mathcal{N}_0$ the algorithm's decisions are considered. Obviously, each set of sensorial samples $\mathcal{N}_0(p, k, o)$ constitutes of $k \in \mathbb{N}$ two-elemented sensorial samples

$$\mathcal{N}_0(p, k, o) = \underbrace{\mathcal{N}_0(p, 1, o) \cup ... \cup \mathcal{N}_0(p, 1, o)}_{k \text{ times}}$$

This is evident due to the definition of sensorial samples and the fact that random nodes are generated from a given sequence of pseudo-random numbers. This means that the desired probabilistic mechanism can be broken down to neighbourhoods $\mathcal{N}_0$ consisting merely of an umbilicus and an additional random node. Here it has to be mentioned that in case of Schelling's segregation model [Sch69] two-elemented sets of sensorial samples are sufficient for emergence of segregated configurations.

4.3 Swarm-Organized Quantization

Obviously, for data objects $\mathbb{X} = \{x_1, ..., x_n\}$ the SOP method has $O(n^2)$ time complexity. A reduction of computational effort is desirable for large scale input data. From literature [Koh97] it is known that computational complexity of algorithms may be greatly decreased by using codebook vectors. Instead of operating on a large set of data vectors, a modified algorithm is supposed to rely on a small set of representatives in order to capture the structure of the input data. The Swarm-Organized Quantiatzion (SOQ) approach is based on using vector quantization error Ψ instead of topographic stress Φ for formation of correct topographic mappings.

The Swarm-Organized Quantization (SOQ) algorithm operates on a finite set of data vectors $\mathbb{X} = \{x_1, ..., x_n\}$ from a vector space $\mathbb{D}$ with norm $\|\cdot\|_{\mathbb{D}}$. Data vectors are to be mapped onto a finite, low-dimensional grid space $\mathbb{O} \subset \mathbb{N}^2$ with distance function $d_{\mathbb{O}}$. Each vector is identified with an agent. The agents' current locations $\{y_1, ..., y_n\} = \{m(x_1), ..., m(x_n)\} \subset \mathbb{O}$ reflect the topographic mapping $m : \mathbb{X} \to \mathbb{O}$. An agent moves to a another node iff its stress Ψ becomes smaller. Stress function $\Psi : \mathbb{X} \times \mathbb{O} \to \mathbb{R}_0^+$ relies on a set of codebook vectors $W = \{w_i : o_i \in \mathbb{O}\} \subset \mathbb{D}$ used for vector quantization purposes. The neighbourhood function is realized by means of a focus function, i.e. as composition $F_\sigma \circ d_{\mathbb{O}}$.

Formula 4.4

$$\Psi(x, o_i) = \|x - w_i\|_{\mathbb{D}} \qquad \textit{with} \qquad w_i \leftarrow \frac{\sum_{j=1}^{n} F_\sigma(d_{\mathbb{O}}(y_j, o_i)) \cdot x_j}{\sum_{j=1}^{n} F_\sigma(d_{\mathbb{O}}(y_j, o_i))}$$

Algorithm

On each iteration all agents are allowed to move simultaneously. An epoch ends if no agent has caused an update, i.e. $\{y_1, ..., y_n\}$ was not modified. See Section 4.2 for a matching approach. The global learning radius σ is decreased after each epoch. σ_{max} is the maximum distance of map space. The algorithm ends iff the smallest possible radius is reached. See Algorithm 4.2 for details. First, the agents are distributed on the grid $\mathbb{O}$ at random (cf. Line 2). For each radius $\sigma \in \{\sigma_{max}, \sigma_{max} - 1, ..., 1\}$ a fixed point iteration is performed. Each iteration (cf. Lines 6 to 19) leads to an update of $\{y_1, ..., y_n\}$. Each agent carrying x_i is allowed to move (cf. Lines 8, 14 to 17) onto random node $o_j \in \mathcal{N}_0(p, 1, y_i)$ iff it decreases its personal amount of stress $\Psi(x_i, o_j)$. An index set $I \subset \mathbb{O}$ indicates which codebook vectors have to be updated (cf. Lines 6, 10 to 13) in order to reduce the computational complexity. This is different from Self-Organizing (Batch) Maps, where update of all codebook vectors is mandatory. See Figure 4.4 for illustration.

Algorithm 4.2 Swarm-Organized Quantization

function SWARMQUANTIZATION($\{x_1, ..., x_n\}$)
  randomize $\{y_1, ..., y_n\}$
  for $\sigma \leftarrow \sigma_{max}, \sigma_{max} - 1, ..., 1$ **do**
    $p \leftarrow gaussian(\sigma)$
    repeat
      $I \leftarrow \emptyset$
      for $i \leftarrow 1, ..., n$ **do**
        $\psi \leftarrow \infty$
        for $o_j \in \mathcal{N}_0(p, 1, y_i)$ **do**
          if $o_j \notin I$ **then**
            $w_j \leftarrow \frac{\sum_{k=1}^{n} F_\sigma(d_{\mathrm{O}}(y_k, o_j)) \cdot x_k}{\sum_{k=1}^{n} F_\sigma(d_{\mathrm{O}}(y_k, o_j))}$
            $I \leftarrow I \cup \{o_j\}$
          end if
          if $\Psi(x_i, o_j) < \psi$ **then**
            $\psi \leftarrow \Psi(x_i, o_j)$
            $y_i \leftarrow o_j$
          end if
        end for
      end for
    until $\{y_1, ..., y_n\}$ fix
  end for
end function

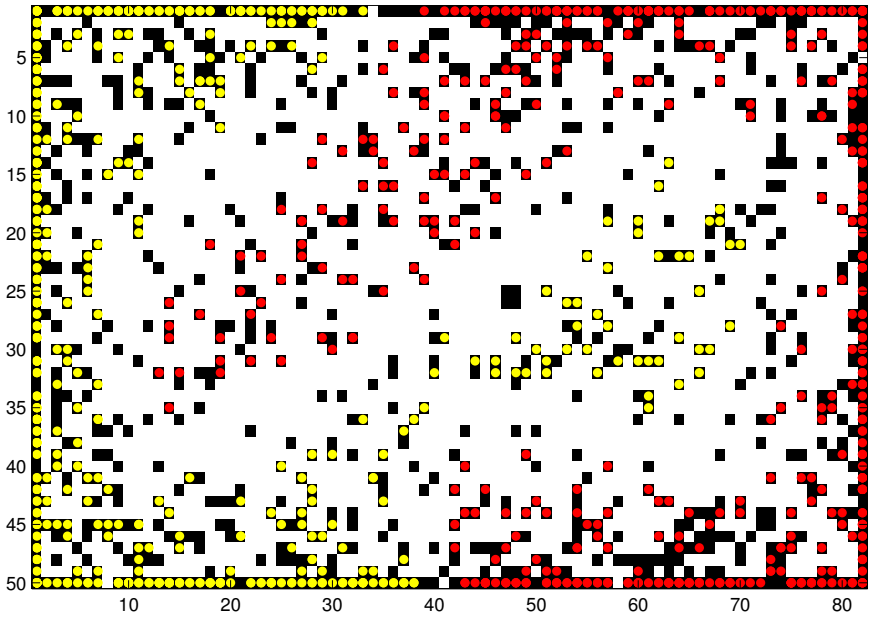


Figure 4.4: Swarm-Organized Quantization: Red and yellow agents represent Chainlink data. Black squares indicate grid nodes with updated codebook vectors according to agents' movements.

4.4 Dissimilarity Visualization

A given mapping $m : \mathbb{X} \to \mathbb{O}$ assigns data objects $\mathbb{X} = \{x_1, ..., x_n\}$ with pairwise dissimilarities $d_{\mathbb{D}} : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ onto $\{y_1, ..., y_n\} \subset \mathbb{O}$. The visualization of $\{y_1, ..., y_n\}$ is easily achieved using low-dimensional scatter plots. Topology preserving mappings (such as Batch-SOM, SOP and SOQ) do not preserve distances or densities of input data on the output space. Instead, approximately uniform distributions are obtained. However, well-established methods such as the U-Matrix and P-Matrix, rely on normed vector spaces. Here, methods are proposed for metric spaces in order to enable the visualization of distance and density structure without the limits of vectorial spaces.

A novel method is introduced for depicting the dissimilarity structure of $\{x_1, ..., x_n\}$ when images $\{y_1, ..., y_n\}$ are available only. The Generalized U-Matrix depicts on each node of the discrete output space the expected data space distance between objects drawn from the node's preimage. See Figure 4.5 for illustration. The U-Matrix follows as $u : \mathbb{O} \to \mathbb{R}_0^+$ with:

Formula 4.5

$$u(o) = \frac{\sum_{i,j} F(d_{\mathbb{O}}(o, y_i)) \cdot F(d_{\mathbb{O}}(o, y_j)) \cdot d_{\mathbb{D}}(x_i, x_j)}{\sum_{i,j} F(d_{\mathbb{O}}(o, y_i)) \cdot F(d_{\mathbb{O}}(o, y_j))}$$

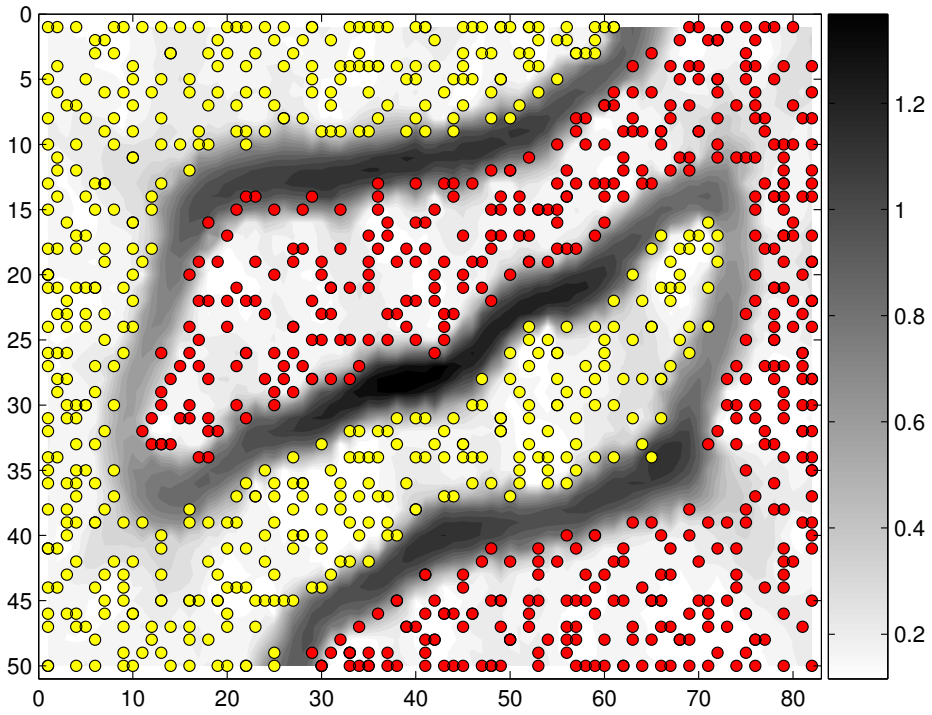


Figure 4.5: Generalized U-Matrix for Chainlink data. Darker shades of gray indicate larger distances in data space.

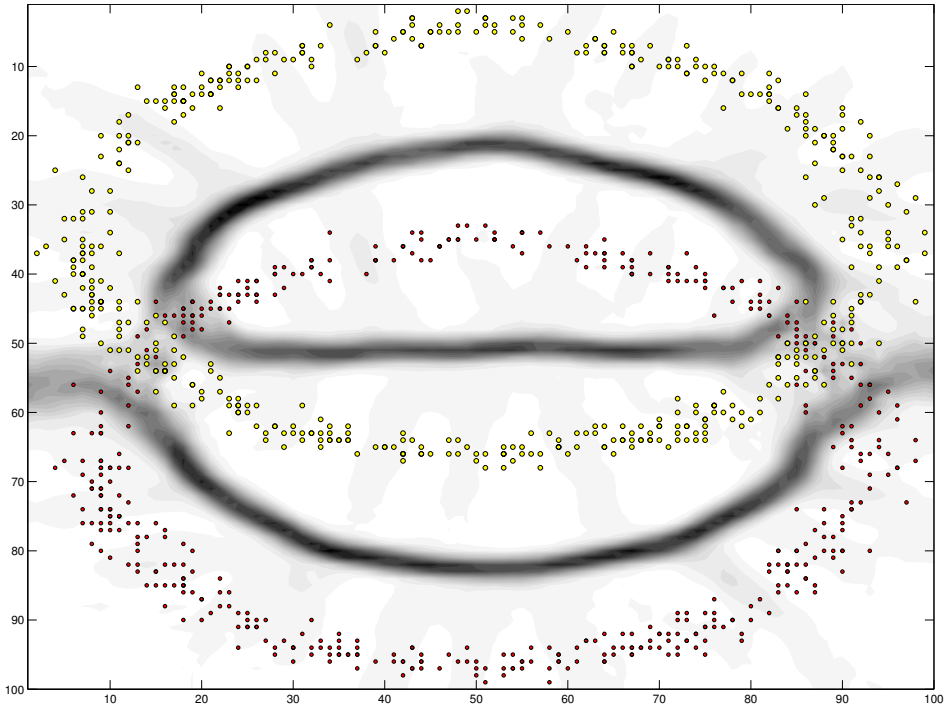
Again $F : \mathbb{R}_0^+ \to [0,1]$ denotes a given focus function in order to realize the local influence of mapped data objects with respect to locations in the output space by means of $F \circ d_{\mathbb{O}} : \mathbb{O} \times \mathbb{X} \to [0,1]$. Obviously, F must be is monotonically decreasing in order to increase the weight of nearby mapped data objects and their dissimilarities with respect to the appearance of the U-Matrix. As a naive approach, F is chosen to be a gaussian kernel function with fixed kernel width on all locations in output space.

The traditional U-Matrix of Ultsch [US90] may be obtained as a special case of the Generalized U-Matrix. Consider codebook vectors W instead of data objects $\{x_1, ..., x_n\}$ by mapping each codebook vector on its own node in output space such that $m(x_i) = m(w_i) = y_i = o_i$ for all nodes' indices i. Furthermore, let the focus function $F : \mathbb{R}_0^+ \to \{0,1\}$ account for immediately grid-neighbouring nodes only. It is evident that the traditional U-Matrix method becomes a special case of the generalized U-Matrix method due to the symmetry of F.

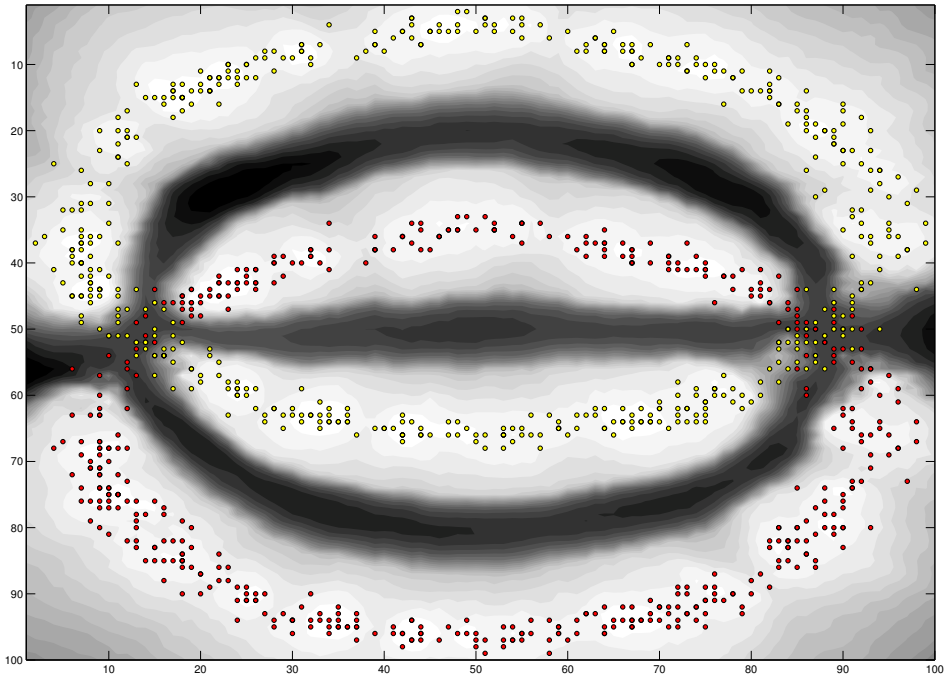
$$\begin{aligned} u(o_k) &= \frac{\sum_{i,j} F(d_{\mathbb{O}}(o_k,y_i)) \cdot F(d_{\mathbb{O}}(o_k,y_j)) \cdot d_{\mathbb{D}}(x_i,x_j)}{\sum_{i,j} F(d_{\mathbb{O}}(o_k,y_i)) \cdot F(d_{\mathbb{O}}(o_k,y_j))} \\ &= \frac{\sum_i F(d_{\mathbb{O}}(o_k,y_i)) \cdot \|w_i - w_k\|}{\sum_i F(d_{\mathbb{O}}(o_k,y_i))} \\ &= \frac{1}{|N(k)|} \sum_{i \in N(k)} \|w_i - w_k\| \end{aligned}$$

The traditional U-Matrix method usually underestimates the dissimilarities of input data $\mathbb{X}$ due to the underlying SOM architecture. Misrepresentations of cluster structures usually are not reflected by height values of U-Matrix. Well-separated clusters being mapped onto the same region, i.e. coherent subset of nodes, do not imply large U-heights despite of occurring large input space distances. The learning method of the Self-Organizing Map (SOM) relies on averaging of data objects in normed vector spaces. Averaging of distant data vectors, however, does not lead to distant codebook vectors and large U-heights due to the triangle inequality in normed vector spaces. For a basic example imagine a 1-dimensional output space with at least $\mathbb{O} = \{1,2,3\}$ nodes arranged in series. Data vectors $\mathbb{X} = \{x_1, ..., x_4\}$ with $\frac{x_2+x_3}{2} = x_1 = x_4$ and $x_2 \neq x_1$ and $m(x_1) = 1$, $m(x_4) = 4$ and $m(x_2) = m(x_3) = 2$. According to Formula 3.6 all codebook vectors are the same with $w_i = x_1$ for all $i = 1, ..., 3$. The traditional U-Matrix is 0 on all nodes, although x_1, x_4 are clearly separated by dissimilar x_2. In contrast to that, the generalized U-Matrix cannot not vanish on $2 \in \mathbb{O}$ due to Formula 4.5. For illustration with the Chainlink data see Figure 4.6.

In case of Self-Organizing Maps, the generalized U-Matrix method does not depict the distance structure of underlying codebook vectors, but data objects. Contrary to the traditional U-Matrix method, the generalized U-Matrix method depicts dissimilar data objects sharing the same space on grid space as large height values.



(a)



(b)

Figure 4.6: Chainlink data mapped onto planar $\mathbb{R}^2$ with PCA from [UM06]. Topographic distortion: red and yellow ring overlap. (a) Traditional U-Matrix depicts small input space distances where rings overlap. (b) Generalized U-Matrix depicts huge input space distances where rings overlap.

4.5 Density Visualization

Embedded Distances for Dissimilarity Data

Usually density estimation is not possible in non-vectorial, discrete data spaces because sampling points for kernel density estimation cannot be provided due to the absence of vector space axioms. On grid-based methods, such as Self-Organizing Maps, codebook vectors act as sampling points for density estimation. Usually, there are no codebooks available on nodes $\mathbb{O} \setminus \{y_1, ..., y_n\}$. Instead, density estimates are available for input data $\{x_1, ..., x_n\}$ and, respectively, $\{y_1, ..., y_n\}$ only. It is evident that sparse regions dividing clusters are impossible to detect in such domains when using traditional density estimation methods. Nevertheless, pairwise dissimilarity data very often is embeddable in normed vector spaces such as the euclidean $\mathbb{R}^k$.

Definition 4.6 *A data set $\mathbb{X} = \{x_1, ..., x_n\}$ with dissimilarities $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ is called* embeddable *in euclidean space if it satisfies the following conditions:*

- $\|.\|$ *is the euclidean norm.*
- *It exists $k \in \mathbb{N}$ and*
- *a set $X' = \{x'_1, ..., y'_n\} \subset \mathbb{R}^k$ with*
- $\forall x_l, x_{l'} \in \mathbb{X} : d(x_l, x_{l'}) = \|x'_l - x'_{l'}\|$.

However, this approach is not based on finding an appropriate set $X' \subset \mathbb{R}^k$. Instead, the embedding condition (see above) is used to derive distances for points graspable only in $\mathbb{R}^k$. As shown by Hasenfuss and Hammer [HH07] distances $\|x'_j - w_i\|$ between data points and codebook vectors can be expressed without finding an embedding in $\mathbb{R}^k$. Assume any codebook vector $w_i \in \mathbb{R}^k$ can be expressed as linear combination of data points, such that $w_i = \sum_{l=1}^n \alpha_{il} \cdot x'_l$ with $\sum_{l=1}^n \alpha_{il} = 1$. Resolving w_i then leads to the following equation:

$$\|x'_j - w_i\|^2 = \sum_l \alpha_{il} \|x'_j - x'_l\|^2 - \frac{1}{2} \sum_{l,l'} \alpha_{il} \|x'_l - x'_{l'}\|^2 \alpha_{il'}$$

This means that distances $\|x'_j - w_i\|$ between data points and codebook vectors can be expressed my means of pairwise dissimilarities $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$. The coefficients α_i for codebook vector w_i of node $o_i \in \mathbb{O}$ are available in SOM-like architectures (cf. Batch-SOM update rule in Equation 3.6). For x_j the coefficient α_{ij} is obtained by means of a given neighbourhood function $F_\sigma \circ d_{\mathbb{O}}$ in Formula 4.7. For σ choose the final radius applied by the learning algorithm.

Formula 4.7

$$\alpha_{ij} = \frac{F_\sigma(d_{\mathbb{O}}(o_i, y_j))}{\sum_l F_\sigma(d_{\mathbb{O}}(o_i, y_l))}$$

Clusters refer to dense regions in data space surrounded by sparse (i.e. less dense) regions. Density information has been shown to be valuable for cluster methods like DBscan [EKSX96] and, especially, for visualization. For each node $o_i \in \mathbb{O}$, the P-Matrix method [Ult03] depicts the estimated density at codebook vector w_i. See Section 3.7 for details. Formally, the kernel density estimate (cf. Section 2.4) is a function of distances toward a sample point chosen from independent and identically-distributed samples of a random variable. In case of dissimilarity data, the kernel density estimate relies on distances that are evaluated by means of coefficients $(\alpha_{ij})_{x_j \in \mathbb{X}}$. The α_{ij} are obtained from a given grid-based topographic mapping. This means that for SOM-like architectures, density estimates are obtained by means of pairwise dissimilarities only. Let $\delta : \mathbb{X} \to \mathbb{O} \to \mathbb{R}_0^+$ denote the distances in embedding space, i.e. $\delta(x_j, o_i) = \|x'_j - w_i\|$. The density estimate f_h then follows as Formula 4.8. For illustration see Figure 4.7.

$$\delta(x_j, o_i) = \sqrt{\sum_l \alpha_{il} d^2(x_j, x_l) - \frac{1}{2} \sum_{l,l'} \alpha_{il} \alpha_{il'} d^2(x_l, x_{l'}) \alpha_{il'}}$$

Formula 4.8

$$f_h(w_i) = \frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{\delta(x_j, i)}{h}\right)$$

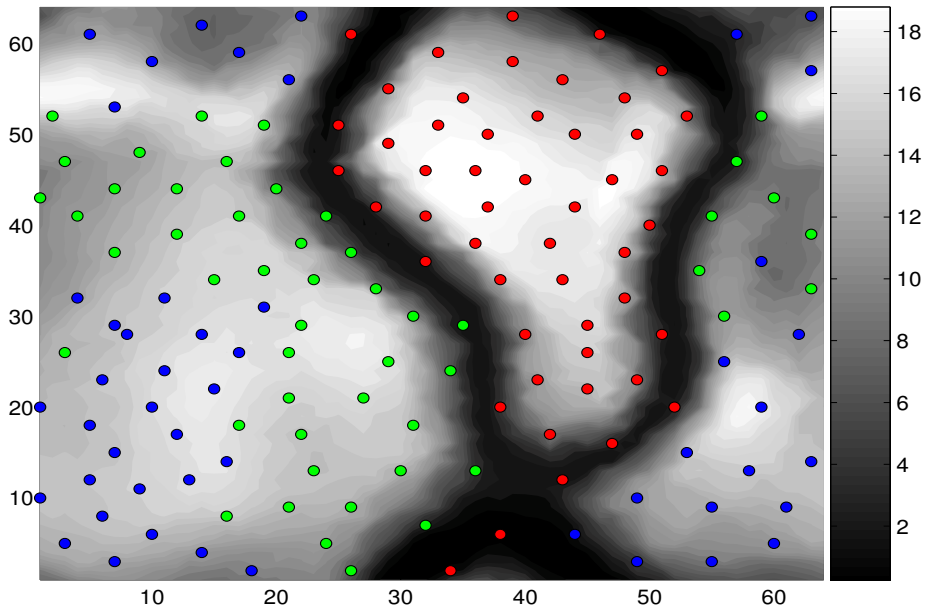


Figure 4.7: Iris data on 64×64 sized toroidal grid. Density estimates for $h = 0.4$ as shades of gray. Low density estimates surrounding Setosa (red).

4.6 Clustering with Swarms

Output-Space Density Estimation for Dissimilarity Data

As indicated in Section 4.2, a large neighbourhood radius $\sigma \in \mathbb{R}_0^+$ forces agents of SOP and SOQ to assemble in dense, sufficiently ordered piles. The density on the grid-shaped output space therefore indicates the structure of input data. The proposed method for depiction of density of data set $\mathbb{X} = \{x_1, ..., x_n\}$ is as follows:

1. Choose a sufficiently large radius $\sigma \in \mathbb{R}^+$ according to the size of the grid $\mathbb{O}$.
2. Use the SOP method (cf. Algorithm 4.1) for creation of a topographic mapping with images $\{y_1, ..., y_n\}$. The radius σ is kept constant. The obtained fixed point contains dense piles of data objects representing clusters in data space.
3. For each node $o_i \in \mathbb{O}$ depict the kernel density estimate

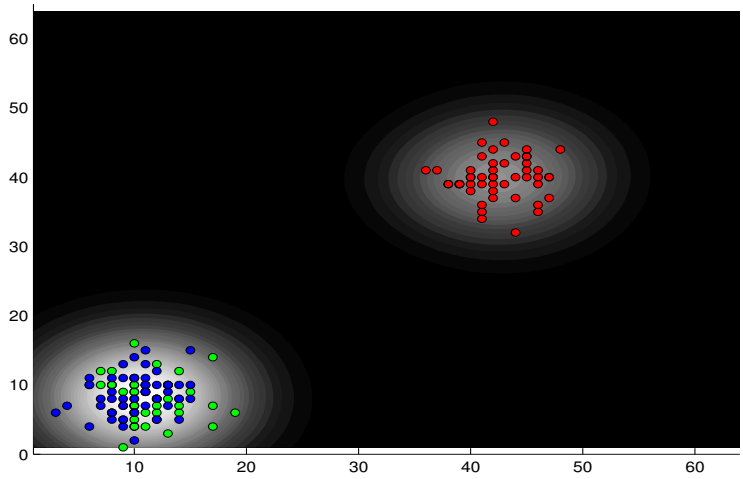
$$f_h(o_i) = \frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{d_{\mathbb{O}}(y_j, o_i)}{h}\right)$$

with $h \leq \sigma$.

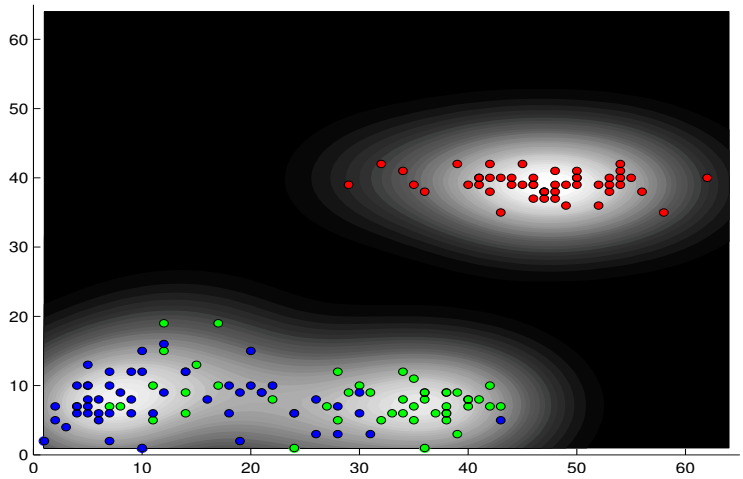
The image $\{y_1, ..., y_n\}$ obtained by SOP emphasizes the cluster structure of the data set, due to the agents' tendency to *flee* agents of other clusters. See Figure 4.8 for illustration.

For cluster analysis with Swarm-Organized Projection (SOP) there are two ways to perform an unsupervised classification of dissimilarity data. A large neighbourhood radius σ forces SOP agents to assemble in dense piles consisting of similar data objects. An additional visual representation, using using scatter plots and distance maps for instance, enables a classification of data objects by hand. Hereby a manually drawn polygon selects a subset of mapped data objects. See [UM06] for a similar approach.

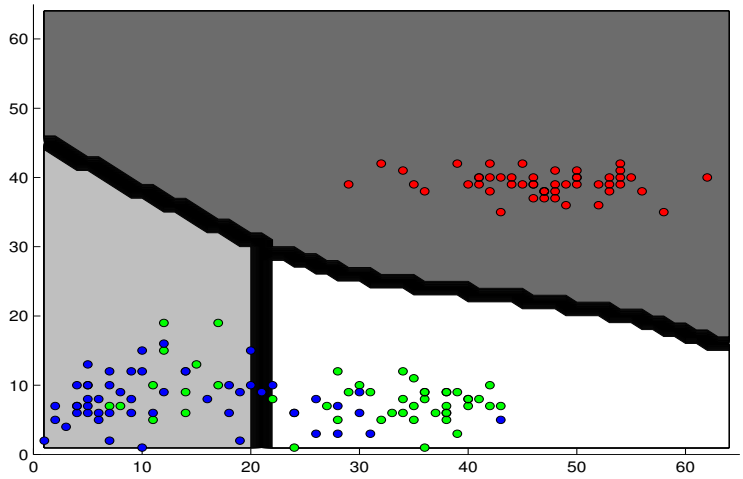
In contrast to that, an automatic method is proposed here. As shown above, kernel density estimation methods are used to quantify the output space density of mapped data objects. Let f_h denote the estimated density function in output space. A dense pile of mapped data objects is indicated by a maximum of f_h. Minimum values of f_h indicate cluster borders. The so-called watershed transformation [RM01] is a geographical segmentation algorithm that derives the divide lines of attraction of rain falling over a landscape. Thus the output space is segmented (classified) in catchment basins of local maxima of the density landscape. See Figure 4.8 for illustration.



(a)



(b)



(c)

Figure 4.8: Iris data on 64×64 toroidal grid. (a) For $\sigma = 46$ the Setosa class (red) is well-separated. (b) For $\sigma = 10$ the Setosa class is well-separated from Versicolor (green) and Virginica (blue). (c) Classification by means of watershed transformation.

4.7 Conclusions

The Swarm-Organized Projection (SOP) is an algorithm at which stochastic agents move on a grid for topographic mapping purposes. SOP addresses the annealing problem of Batch-SOM and CCA by means of probabilistic fixed points, i.e. the agents' behaviour determines the shrinking of neighbourhoods. Swarm-Organized Quantization (SOQ) uses the swarm of agents to guide the update and usage of codebook vectors. This is done in order to decrease the computational effort. The generalized Unified Distance Matrix depicts the distance structure of dissimilarity data on top of topographic mappings even in non-vectorial spaces.

Chapter 5

Assessment

In this chapter, the topographic mapping methods from Chapter 3 and 4 are formally assessed with respect to properness for visual cluster analysis.

5.1 Ant-Based Clustering

Sampling with Ants

In literature little can be found about the influence of ants on the abilities of Ant-Based Clustering (ABC) by Lumer and Faieta [LF94]. Here, it will be argued that ants of ABC are a sampling method, i.e. randomized selecting of data objects. A large number of ants will cause many pick and drop actions in a certain time interval, in comparison with few ants. Thus the adjusted probability $\hat{p}_x(o_i \to o_j)$ is considered, i.e. the probability for x to be moved from node o_i to o_j in relation to all movements.

Theorem 5.1 *The adjusted probability $\hat{p}_x(o_i \to o_j)$ is the same for $n, n' \in \mathbb{N}$ many ants.*

Proof Let $p_x(o_i \to o_j)$ denote the probability for object x being moved from node o_i to o_j. Obviously, $p_x(o_i \to o_j)$ can be split up into several independent terms, namely comprising the probability for an ant_k to hit node o_i for ant_k to pick up x, for ant_k to hit node o_j and, finally, drop x onto node o_j.

$$p_x(o_i \to o_j) = p(\text{ant}_k \text{ hits } o_i) \cdot p_{pick}(x, o_i) \cdot p(\text{ant}_k \text{ hits } o_j) \cdot p_{drop}(x, o_j)$$

The adjusted probability $\hat{p}_x(o_i \to o_j)$ is then obtained by normalization with the probability for moving $x \in \mathbb{X}$ onto any node.

$$\hat{p}_x(o_i \to o_j) = \frac{p_x(o_i \to o_j)}{\sum_{o_l \in \mathbb{O}} p_x(o_i \to o_l)}$$

Picking and dropping probabilities p_{pick}, p_{drop} are not affected by the number of ants. Also $p(\text{ant}_k \text{ hits } o_j)$ remains constant because ant_k is not affected by other ants. If the number of ants increases from n to n' the probability $p(\text{ant}_k \text{ hits } o_i)$ for node $o_i \in \mathbb{O}$ to get hit by an ant also increases, such that $p(\text{ant}_k \text{ hits } o_i)$ cancels

out in the fraction above. This means that $\hat{p}_x(o_i \to o_j)$ is the same for $n, n' \in \mathbb{N}$. $\square$

It is evident that ants in ABC act as an arbitrary sampling mechanism for data objects, due to the following reasoning. According to Theorem 5.1 the number of ants can be reduced to one, without any loss of computational abilities. The central limit theorem in statistics states that the sum of independent observations approximates the normal distribution. During $t \in \mathbb{N}$ steps, the probability distribution of a random-walking ant approximates a normal distribution with zero mean and variance of t. On a finite grid the random walking ant, therefore, approaches each node with equal probability for large numbers of steps. Ants do not affect the obtainable piles of objects, but the attractiveness function ϕ does (cf. Section 3.4). This has been empirically verified by Tan et al. [TTT06]. It could be shown that few ants lead to the same cluster structures as many ants. Furthermore, same results could be obtained when using ϕ directly for probabilistic cluster assignments.

Unifying Framework with SOM

Self-Organizing Batch Maps (Batch-SOM) and Ant-Based Clustering (ABC) are based on different architectures (see Section 3.2 and 3.4). Batch-SOM modify codebook vectors, whereas ABC relies on stochastic sampling and permutation of mapped data objects. A unifying framework for both algorithms exists by means of objective functions that are to be denoted by means of three functions: norm $\|.\|_{\mathbb{D}}$, focus $F : \mathbb{R}_0^+ \to [0, 1]$ and mapping $m : \mathbb{X} \to \mathbb{O}$.

The **Batch-SOM** offers a meaningful objective by means of quantization error Ψ, because its minimization determines the update of mapping. Resolving codebook vectors (cf. Formula 3.6) leads to Formula 5.2. $\Psi(x_i)$ represents the norm of averaged differences $x_i - x_j$ over grid-neighbouring data points $x_j \in \mathbb{X}$.

Formula 5.2

$$\Psi(x_i, o_l) = \frac{\left\| \sum_{x_j} F(d_{\mathbb{O}}(y_j - o_l)) \cdot (x_i - x_j) \right\|_{\mathbb{D}}}{\sum_{x_j} F(d_{\mathbb{O}}(y_j - o_l))}$$

The **Dissimilarity-SOM** chooses codebooks among the given data set $\mathbb{X} = \{x_1, ..., x_n\}$ (cf. Section 3.2). For the sake of simplicity, the norm $\|.\|_{\mathbb{D}}$ is used. The quantization error $\Psi(x, o_i) = \|x - w_i\|_{\mathbb{D}}$ determines the update of the SOM with (cf. Formula 3.2):

$$w_i \leftarrow \arg\min_{w \in \mathbb{X}} \Phi(w, o_i) \quad \text{with} \quad \Phi(w, o_i) = \frac{\sum_{x_j} F(d_{\mathbb{O}}(o_i, y_j)) \cdot \|w - x_j\|}{\sum_{x_j} F(d_{\mathbb{O}}(o_i, y_j))}$$

Thus, the Dissimilarity-SOM incorporates a combination of Ψ and Φ, i.e. quantization error and topographic stress.

Ants in **Ant-Based Clustering** act as an arbitrary sampling method, such that the pick/drop mechanism is to be omitted in favor of an analysis of underlying formulae. First, a meaningful focus function $F : \mathbb{R}_0^+ \to [0, 1]$ for ABC is derived by means of ants' perceptive neighbourhood radius σ:

$$F(\delta) = \begin{cases} 1 & : \quad \delta \leq \sigma \\ 0 & : \quad \text{else} \end{cases}$$

Thus, the objective function ϕ of ABC is restated as Formula 5.3 by use of $|N(x, o_i)| = \sum_{x_j} F(d_{\mathbb{O}}(y_j, o_i))$. The objective ϕ incorporates Φ of Dissimilarity-SOM that quantifies the local stress of topographic mapping, similar to Ψ of Batch-SOM. Φ even acts as an upper limit to Ψ due to the triangle inequality in normed vector spaces.

Formula 5.3

$$\phi(x, o_i) = \frac{|N(x, o_i)|}{\sigma^2} \cdot \left(1 - \frac{\Phi(x, o_i)}{\alpha}\right)$$

Formula 5.4

$$\Phi(x, o_i) = \frac{\sum_{x_j} F(d_{\mathbb{O}}(y_j, o_i)) \cdot \|x - x_j\|}{\sum_{x_j} F(d_{\mathbb{O}}(y_j, o_i))}$$

Insights

ABC relies on the topographic stress term Φ that is found in Dissimilarity-SOM and SOP, and serves as an upper limit to Ψ of Batch-SOM. However, ABC uses a fixed neighbourhood function with small radius, whereas Batch-SOM and Dissimilarity-SOM both use neighbourhood functions with large but shrinking radiuses. ABC has a probabilistic update of images $\{y_1, ..., y_n\}$ whereas Batch-SOM and Dissimilarity-SOM are deterministic. The ABC objective ϕ decomposes into two factors: the output density term $\frac{|N(x)|}{\sigma^2}$ and $1 - \frac{\Phi}{\alpha}$ which is obviously related to topographic quality. Φ is easily identified as a topographic distortion measure because of its relation to Ψ of Batch-SOM. For a brief overview of differences see Table 5.1.

	Batch-SOM	Dissimilarity-SOM	ABC
neighbourhood $h : \mathbb{O} \times \mathbb{O} \to [0, 1]$	large, shrinking	large, shrinking	small, fixed
update of $m : \mathbb{X} \to \mathbb{O}$	deterministic	deterministic	probabilistic
searching for update of $\{y_1, ..., y_n\}$	global $\mathbb{O}$	global $\mathbb{O}$	local $\subset \mathbb{O}$
signature term	Ψ	Ψ, Φ	$\frac{\lvert N\rvert}{\sigma^2}(1 - \frac{\Phi}{\alpha})$
termination	user-defined annealing	user-defined annealing	never

Table 5.1: Characterization of Batch-SOM, Dissimilarity-SOM and Ant-Based Clustering.

In ABC ants perform pick/drop actions in order to maximize ϕ which consists of two factors: First, $\frac{|N_x(i)|}{\sigma^2}$ denotes the density in output space around node i. This density is not related to any structural feature of the data set $\{x_1, ..., x_n\}$. Furthermore, output space density is easily maximized by ants and, therefore, will dominate the product ϕ. Ants quickly form clusters without any topological ordering. The term Φ is a weighted and normalized sum of data space distances, known from Dissimilarity-SOM and SOP to produce sufficient topographic mappings. The focus function F introduced for ABC realizes a neighbourhood that is too small for obtaining good topographic mappings, when compared with Self-Organizing Batch Maps. Instead, large and shrinking neighbourhoods should be used in order to

enable the formation of global topographic structures. Thus $(1 - \frac{\Phi}{\alpha})$ determines topographic quality. The scalar $\alpha \in \mathbb{R}^+$ is a crucial parameter used for normalization of Φ.

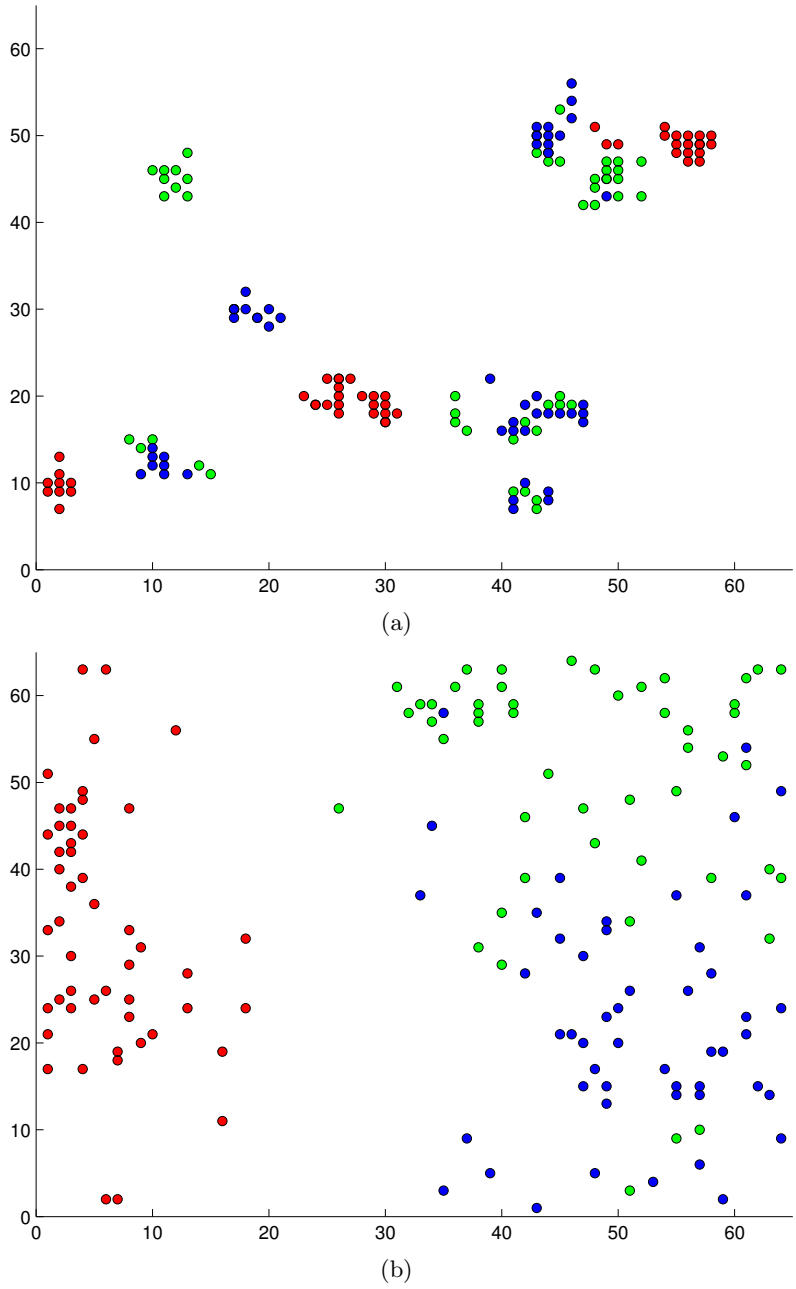


Figure 5.1: Ant-Based Clustering mapping Iris data [Fis36] onto planar grid. (a) Original method [LF94] produces too many piles. (b) Improved method [HU08] does not account for output space densities and rather uses large neighbourhoods. No dense piles emerge. The Setosa class (red) is well separated. Versicolor (green) and virginica (blue) appear sufficiently sorted.

Experimental Validation

Ant-Based Clustering (ABC) was modified in [HU08] in order to achieve improvements in terms of topographic mapping:

- apply larger neighbourhoods,
- discard the output space density term $\frac{|N(x,o_i)|}{\sigma^2}$.

According to theory, this scheme is supposed to produce better topographic mappings than the original ABC method on an average. See Figure 5.1 for illustration. On vectorial data, such as the Fundamental Clustering Problem Suite [Ult], the U-Map method [UM06] derives a set of codebook vectors for arbitrary mappings. The so-called Minimal Path Length (MPL, [DM90] [GFS95]) is a coarse indicator for distortions of topographic mappings. See Formula 5.5 for details. MPL can be applied for topographic mappings that rely on contrastable output spaces. Neighbouring codebooks are highly dissimilar where mappings are non-cohesive. The improved ABC in fact produces significantly less distorted mappings (see Table 5.2), i.e. the MPL values are smaller on an average. A two-sample Kolmogorov-Smirnov test [Smi48] was applied because the distribution of MPL values usually is not normally distributed but skewed. These findings were significant below the $\alpha = 10^{-4}$ level. This means that the theoretical analysis of ABC algorithms is consistent with the empirical results obtained in [HU08].

Formula 5.5

$$mpl = \sum_{o_i \in \mathbb{O}} \sum_{o_j \in N(o_i)} \|w_i - w_j\|_{\mathbb{D}}$$

Data set	Original ABC		Improved ABC	p-value
Atom	161 ± 15.6	$>$	142 ± 6.2	$1.24E{-}12$
Chainlink	6.33 ± 0.33	$>$	6.19 ± 0.12	$1.38E{-}05$
Hepta	11.16 ± 0.66	$>$	9.86 ± 0.54	$2.65E{-}13$
Iris	11.8 ± 0.65	$>$	10.02 ± 0.57	$1.03E{-}17$
Target	6.69 ± 0.41	$>$	5.35 ± 0.33	$8.79E{-}23$
2Diamonds	3.86 ± 0.09	$>$	3.28 ± 0.10	$1.08E{-}23$
Wingnut	5.64 ± 0.32	$>$	5.07 ± 0.23	$9.91E{-}11$

Table 5.2: Topographic distortion measured by MPL method: mean values $\pm$ standard deviation, p-values of statistical testing indicate that Ant-Based Clustering (ABC) produces less distorted mappings when using larger neighbourhood and discarding output space densities [HU08].

5.2 Sound and Complete Learning

Neighbour retrieval is the task of finding data space neighbours of few interesting data objects solely based on a low-dimensional display, i.e. topographic mapping [Ven07] [NVK07]. Topographic mapping methods are formally assessed whether neighbour retrieval is in principle achievable or not. Here, the quality of the update

rule is to be assessed using the concepts of soundness and completeness in terms of neighbour retrieval.

Definition 5.6 *A learning algorithm is* sound *iff the update rule, i.e. for each x_i the transition from $\{y_1, ..., y_n\}$ to $\{y_1, ..., y_{i-1}, y_i', y_{i+1}, ..., y_n\}$, leads to an improved neighbour retrieval regarding x_i and y_i, respectively.*

Definition 5.7 *A learning algorithm is* complete *iff an improvement of neighbour retrieval regarding x_i and, respectively, y_i may be achieved by means of the update rule, i.e. the transition from $\{y_1, ..., y_n\}$ to $\{y_1, ..., y_{i-1}, y_i', y_{i+1}, ..., y_n\}$.*

Completeness is the inverse concept of soundness. Both concepts are important for the understanding of learning algorithms' aims. However, topography (pairwise distances, ranks of distances, neighbourhood relations) usually cannot be preserved in general (cf. Section 3.1). Thus, sound and complete learning of topographic mappings is not achievable in practice, i.e. the algorithms' update rules contradict for each $x_i \in \mathbb{X}$. In addition to formal analysis of learning algorithms, an empirical evaluation of topographic mapping quality is therefore necessary.

Self-Organizing Batch Maps

Quantization error Ψ for Batch-SOM is:

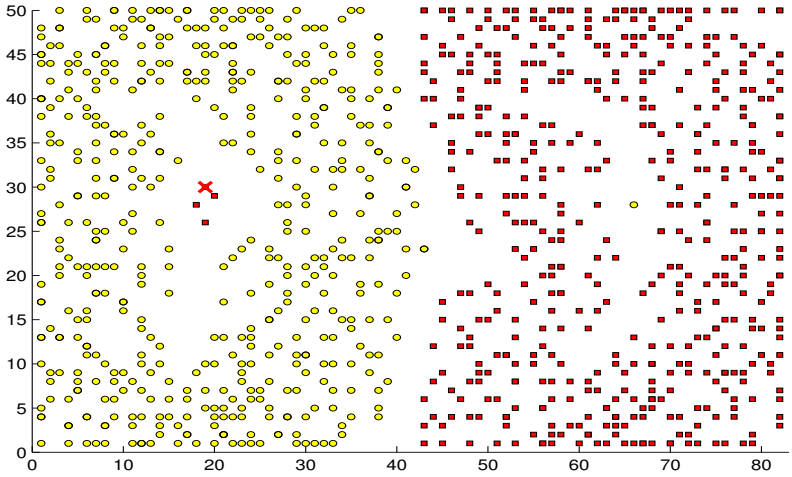
$$\Psi(x, o_i) = \|x - w_i\|_{\mathbb{D}} = \frac{\left\| \sum_{x_j} F(d_{\mathbb{O}}(o_i, y_j)) \cdot (x - x_j) \right\|_{\mathbb{D}}}{\sum_{x_j} F(d_{\mathbb{O}}(o_i, y_j))}$$

For large values of radius σ, i.e. early in the learning process, the term Ψ results in the average of a large number of randomly chosen vectors. This means that Ψ arbitrarily approximates 0. The deterministic and exhaustive bestmatch search will therefore map data vectors on random locations in output space. Later in the learning, when σ is small, the topography of the input space is more and more important for Ψ. However, these updates may not be able to correct an erroneous mapping during the early learning phase. See Figure 5.2(b) for an example where the Chainlink data is misrepresented in the map space. As the example shows, learning in SOM is neither sound nor complete. Therefore, it is prone to misrepresent cluster structures. This holds for the Swarm-Organized Quantization, too, since it is based on Ψ.

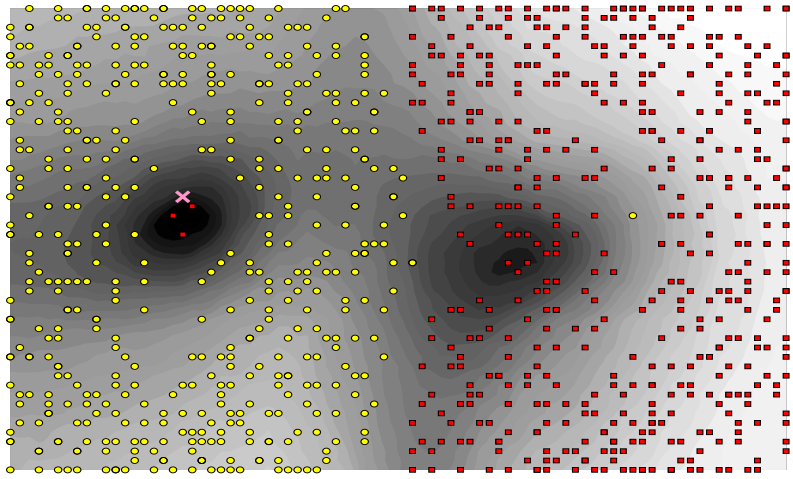
Curvilinear Component Analysis

In CCA an arbitrary projection point $y_i = m(x_i)$ is temporarily fixed, and all other y_j move around in order to adjust the pairwise distances. The update rule [DH97] for CCA is $\Delta y_j = \alpha \cdot F_\sigma(d_{\mathbb{O}}(y_i, y_j)) \cdot (d_{\mathbb{D}}(x_i, x_j) - d_{\mathbb{O}}(y_i, y_j)) \cdot \frac{y_j - y_i}{d_{\mathbb{O}}(y_y, y_i)}$ where $\alpha \in (0, 1)$ denotes the learning rate that is to be decreased down to 0. This is a difference to Self-Organizing Maps where bestmatching neurons are sought.

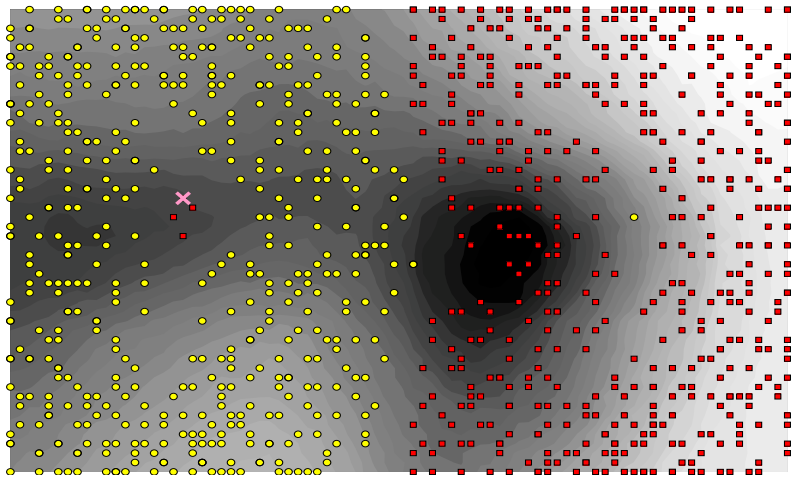
The different parts of Δy_j contribute to complete and sound leaning rule. This can be seen as follows: the term $\frac{y_i - y_j}{d_{i,j}}$ is a directional unit vector, due to $d_{\mathbb{O}}(y_i, y_j) =$



(a) Erroneous Topographic Mapping



(b) Quantization Error Ψ



(c) Topographic Stress Φ

Figure 5.2: Chainlink data: criterion Ψ of Batch-SOM is neither complete nor sound. Darker shades of gray indicate smaller mapping errors. (a) Few red points are located in the yellow cluster. Falsely mapped element of red cluster depicted as cross. (b) $\Psi(x)$ falsely assumes minimum inside the yellow cluster. (c) $\Phi(x)$ of SOP correctly assumes minimum inside the red cluster.

$\|y_i - y_j\|_{\mathbb{O}}$, that is valid unless $y_i = y_j$. The term $(d_{\mathbb{D}}(x_i, x_j) - d_{\mathbb{O}}(y_i, y_j))$ adjusts the images y_j due to data space distances. Repeated application of Δy_j forces all data space neighbours of x_i to have exactly the same distances in map space as in data space. Non-neighbouring data objects are forced to have a greater distance in output space. However, the repeated application of CCA's Δ rule is only complete and sound in terms of neighbour retrieval if the condition holds that the neighbourhood function h_σ captures all neighbours and non-neighbours that happen to be misplaced as map space neighbours. Thus, the ability to produce cohesive mappings essentially depends on the applied annealing of neighbourhood radius [DH97]. An optimal annealing scheme depends on the data manifold's shape. This is, however, an unknown quantity.

Stochastic Neighbour Embedding

The probability for x_i, x_j being neighbours is denoted as $p_{ij}, q_{ij} \in [0, 1]$ in data space and output space. For t-SNE [vdMH08] the update of an image y_i in output space follows as:

$$\Delta y_i = \sum_{x_j} (p_{ij} - q_{ij}) \cdot (y_i - y_j)$$

This term is similar to the one used in CCA (see above). The main difference consists of replacing $(d_{\mathbb{D}}(x_i, x_j) - d_{\mathbb{O}}(y_i, y_j))$ with the difference of probabilties $(p_{ij} - q_{ij})$, which makes the update rule less susceptible to outliers in data space.

Swarm-Organized Projection

Soundness directly follows from the definition of the update rule, which minimizes the sum of data space distances toward neighbouring objects. Agents move simultaneously iff they can decrease their personal amount of topographic stress $\Phi(x)$. Criterion $\Phi(x)$ assumes its minimum value iff neighbourhoods coincide in both data space and map space. See Figure 5.2(c) for illustration. The SOP algorithm therefore has a sound update rule.

On the other hand, each decrease of the average distance toward neighbouring objects is enabled by the probabilistic SOP update rule. Formally, SOP is complete. However, it remains unclear whether all improvements of neighbourhood preservation are achievable by search-based techniques or not, if agents act independently. Take for instance a setup where the k-nearest-neighbour relation is not symmetrical. An agent can in general not affect the location of his $i = 2, ..., k$ nearest neighbours

Method	Soundness	Completeness
Batch-SOM	no	no
SOQ	no	no
CCA	yes	yes
t-SNE	yes	yes
SOP	yes	conditional

Table 5.3: Formal assessment of learning algorithms.

5.3 Swarm-Organized Mappings

Complexity

Let $n \in \mathbb{N}$ denote the number of input samples. The number of grid nodes is given by $m \in \mathbb{N}$. On each epoch the Batch-SOM creates m new codebook vectors. Each codebook vector is a weighted sum of n input samples. Additionally, each data vector is compared against the m codebook vectors for bestmatch retrieval. For a given cooling scheme with $\approx \sqrt{m}$ epochs in case of a two-dimensional output space, this results in $\approx 2n\sqrt{m}m$ operations. Therefore, the Batch-SOM has $O(n\sqrt{m}m)$ time complexity.

The algorithms of both CCA and SNE are expressed as two nested loop, each of which has linear complexity in the order of n. There both algorithms have $O(n^2)$ time complexity.

On each epoch the SOP method evaluates Φ exactly n times. Each evaluation sums up a small fraction of available n distances. The initial learning radius is chosen in order to cover the whole map, i.e. a function of $\sqrt{m}$. So the number of learning epochs is proportional to $\sqrt{m}$ because each epoch the learning radius is decreased constantly. Therefore, the SOP method has order of $\sqrt{m} \cdot n^2$ time complexity.

On each epoch the SOQ method evaluates the Ψ criterion n times. In analogy to the Batch-SOM, there are no more than m codebook vectors to compute. In analogy to the SOP method, there are no more than m evaluations of Ψ to be carried out. Each codebook vector is a weighted sum of n input samples. The initial learning radius is chosen in order to cover the whole map, i.e. a function of $\sqrt{m}$. The number of learning epochs is proportional to $\sqrt{m}$ because each epoch the learning radius is decreased constantly. Therefore, the SOQ method has order of $\sqrt{m} \cdot n \cdot (\min(m, n) + 1)$ time complexity. The Swarm-Organized Quantization clearly benefits from the swarm-driven update of codebook vectors.

Method	Time Complexity
Batch-SOM	$O(m\sqrt{m} \cdot n)$
CCA	$O(n^2)$
SNE	$O(n^2)$
SOP	$O(\sqrt{m} \cdot n^2)$
SOQ	$O(\sqrt{m} \cdot n \cdot (\min(m, n) + 1))$

Table 5.4: Complexity of topographic mapping methods.

Fixed Points & Halting

Execution of SOP depends on fixed point iteration. With regard to probabilistic agents, fixed point refers to the state where no agent has moved. Here, the term *probabilistic* fixed point is used in order to prevent confusion with (deterministic) fixed points in mathematics.

Let $0 \leq p < 1$ be the upper bound of probability for an agent to move. The probability $P(t)$ that during $t \in \mathbb{N}$ iterations at least one of $n \in \mathbb{N}^n$ moved follows

as $P(t) = (1-(1-p)^n)^t$. Since $\lim_{t\to\infty} P(t) = 0$ the fixed point iteration is likely to stop after a sufficient number of iterations. Therefore, SOP and SOQ learning algorithms will halt. This is an advantage over Batch-SOM, that cannot guarantee to halt. However, this argument is not a proof for convergence in the sense of continuously diminishing movements of agents.

In contrast to that, the Batch-SOM [Koh97] relies on simultaneous update of all codebook vectors. A formal proof for existence of fixed points in SOM, i.e. convergence of codebook vectors, does not exist for a given neighbourhood radius. For example, let $\mathbb{X} = \{x_1, ..., x_4\}$ denote an arbitrary set of data objects with $d(x_2, x_3) = d(x_3, x_2) > d(x_i, x_j)$ for all $(i, j) \notin \{(2, 3), (3, 2)\}$. The output space $\mathbb{O} = \{1, ..., 6\}$ is planar with focus function $F(x) = 1$ for $x \leq 1$ and 0 elsewhere. Thus the Batch-SOM update rule diverges between two states $(y_1, ..., y_4) = (1, 2, 5, 6)$ and $(1, 3, 4, 6)$. For practical use, convergence is forced by means of a given annealing scheme for neighbourhood radius σ.

Both methods CCA [DH97] and t-SNE [vdMH08] were derived by means of an objective function which is to optimize. This is realized by means of (stochastic) gradient descent methods, whose termination is specified by a given number of iterations. So convergence is forced by means of parametrization.

Method	Termination	Fixed Point
Batch-SOM	annealing	no
CCA	annealing, stochastic gradient descent	yes
t-SNE	gradient descent	yes
SOP	probabilistic	yes*
SOQ	probabilistic	yes*

Table 5.5: Termination of topographic mapping methods. Fixed points of SOP and SOQ methods are probabilistic.

5.4 Conclusions

A unifying framework for ABC, Batch-SOM and Dissimilarity-SOM was derived. Compared with SOM, the ABC method misleadingly accounts for output space density and, furthermore, uses too small but fixed neighbourhood functions. Naive improvements empirically verify the analytical results.

In contrast to Batch-SOM, the update rule of most topographic mapping algorithms (including SOP) is complete and sound. The computational complexity of Swarm-Organized Quantization is in the worst case contrastable with the Batch-SOM. It can be shown that both Swarm-Organized Mapping methods will halt.

Chapter 6

Experimental Validation

In this chapter the abilities of the Swarm-Organizing Projection (SOP) are empirically demonstrated by few selected problems. The quality of topographic mappings is investigated in comparison with Self-Organizing Maps (SOM) and Curvilinear Component Analysis (CCA). These methods are challenged to produce faithful representations of several data sets containing cardinal cluster problems.

6.1 Quality Assessment

External Measures

A quality measure is called *external* if the obtained configuration is compared against some *ground truth* in the sense of a given solution that is assumed to be true. Popular examples of external measures are F-Measure and Rand-Index [Ran71]. The Rand-Index quantifies the agreement of two classifications c, c' on all pairs of data objects, i.e. same or different classifications in both classifications. The F-Measure [vR79] is the harmonic mean of precision and recall. Both methods rely on classifications, whereas one is assumed to be ground truth. Topographic mapping methods, such as SOM and SOP, usually do not yield classifications, even though clustering methods may operate on top of mappings. Therefore, external measures are hardly applicable when evaluating the quality of topographic mapping methods.

Internal Measures

Another topic is concerned with the internal structure of the obtained solutions. A quality measure is called *internal* if the obtained solution is assessed without involving a comparison with an external, known-to-be-true solution. Popular internal measures are Intra-Cluster Variance [Llo03] and the Dunn-Index [Dun74]. The intra-cluster variance (or *sum-of-squared-errors minimum variance criterion*) is the sum of squared distances between the centroid of each cluster and the corresponding data points. The Dunn-Index measures the ratio between cluster distances and diameters, which should be large in a good clustering.

The Silhouette plot [KR90] is a popular internal method for visual evaluation of clusterings. A score function $s : \mathbb{X} \to [-1, 1]$ 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 $s(x) = \frac{b(x)-a(x)}{\max\{a(x),b(x)\}}$. 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. However, it is evident that silhouette scores assume clusters of spherical or gaussian shape.

Topographical Measures

Topographical measures quantify the amount of topography that is preserved when mapping data objects $\{x_1, ..., x_n\}$ onto $\{y_1, ..., y_n\}$. Topographical measures are internal measures in the sense that comparisons with external solutions are not involved. Forward projection errors [UH05] occur if similar data objects x_i, x_j are mapped onto faraway points $y_i, y_j \in \mathbb{O}$. Backward projection errors occur if the reverse mapping assigns nearby locations $y_i, y_j \in \mathbb{O}$ to dissimilar data objects $x_i, x_j \in \mathbb{D}$. An appropriate topographic measure might quantify the amount of one of these errors. Topographical preservation is denoted by means of several formalisms. See [BHV99] [GFS95] for details.

*D*istance preservation is not possible in general when trying to map high-dimensional data onto lower-dimensional spaces [Dry78] [Kir78] [Sch80]. However, the objective function of MDS and Sammon's Mapping are widely used for assessment of arbitrary topographic mappings [HKD05] [Wis08]. Quantification of distance preservation is applicable on different architectures, such as CCA and ABC. This approach penalizes topology preserving mappings, such as SOM and SOP.

*R*anks of distances usually cannot be preserved entirely when mapping high-dimensional data onto low-dimensional spaces [Sch80]. The euclidean distance $d_{\mathbb{O}}$ used in output space implies radial neighbourhoods. For example, Bezdek and Pal [BP93] propose to evaluate the correspondence of rankings of all distance pairs occurring in data space and output space. Spearman's ρ is a statistical measure that quantifies the degree of correlation between ranking orders.

*G*eometrical neighbourhood relations, such as Delaunay graphs, can be preserved iff the intrinsic dimension of the data matches the dimension of the output space [BHV99]. Neighbourhood relations can be assessed in principle for any type of topographic mapping. However, toroidal and planar output spaces (and therefore mappings) are hardly comparable with these approaches because violations of neighbourhoods are expected due to mismatching topologies. For example, mapping an intrinsically planar manifold onto a toroidal output space leads to additional backward projection errors that are avoided on planar output spaces, despite of occurring cluster disruptions.

Zrehen's measure [Zre93] quantifies the local organization of codebook vectors. Immediately grid-neighbouring codebooks w_i, w_j are not allowed to

have *intruders*, i.e. codebooks w_k that violate the condition $\|w_i - w_k\|^2 + \|w_k - w_j\|^2 \leq \|w_i - w_j\|^2$. The Z-measure is the sum of intruders. It relies on vector spaces with euclidean norm $\|.\|$ and regularly arranged codebook vectors, i.e. requirements that do not met with arbitrary topographic mappings. The Minimal Pathlength method [DM90] [GFS95] is the sum of data-space distances of grid-neighbouring codebook vectors, i.e. $E = \sum_i \sum_{j \in N_{\mathbb{O}}(i)} d_{\mathbb{D}}(w_i, w_j)$. It relies on SOM-like structures and, furthermore, will first penalize the mismatching topology of the output space instead of cluster disruptions. The Topographic Function [BHV99] quantifies the identity of the Delaunay graphs in input space and output space, whereas (mapped) data points represent the vertices. In high-dimensional spaces, Delaunay graphs are too costly to retrieve.

Measure	Preservation	Architecture	Remarks
MDS	distance	arbitrary	penalizes topology preserving mappings
Spearman's ρ	ranks of distances	SOM-like	penalizes topology preserving mappings
Kaski's Trustworthiness	limited ranks of distances	SOM-like	penalizes topology preserving mappings
Zrehen	topology	SOM-like	relies on neighbourhood relation in output space
Minimal Pathlength	topology	SOM-like	relies on neighbourhood relation in output space
Topographic Function	topology	SOM-like	too costly

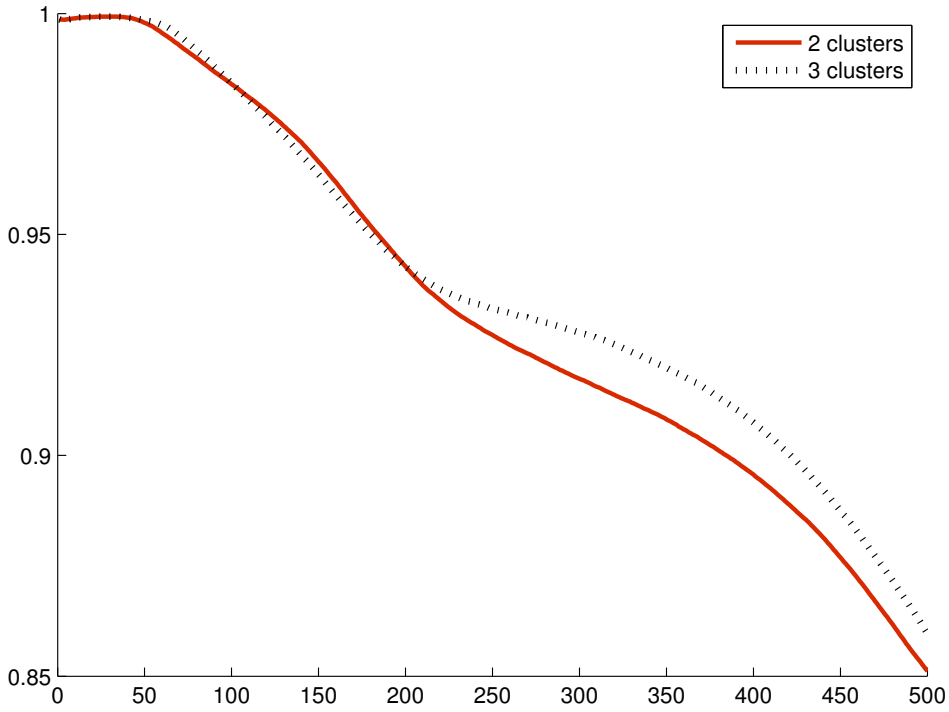
Table 6.1: Topographical measures.

Kaski's Dilemma

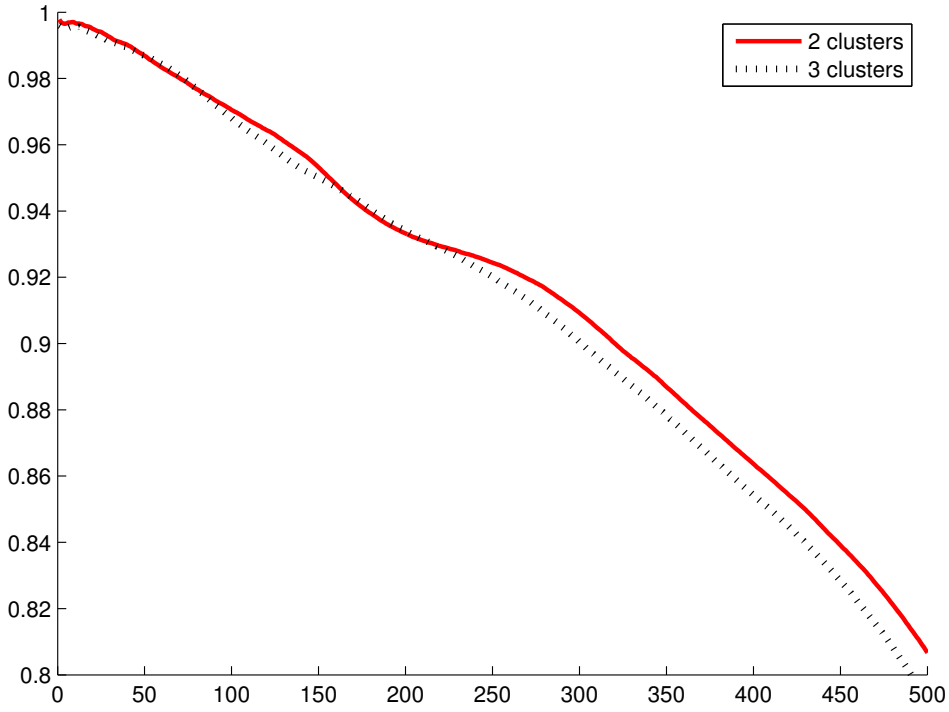
Kaski [KNO+03] proposed topographic measures called trustworthiness and continuity, in order to quantify the overlap of small rank-based neighbourhoods in data space and output space. This approach penalizes SOM-like methods due to radial neighbourhoods. A major drawback is its inability to correctly distinguish whether a class of data objects is disrupted in output space or not. In Figure 3.6 the two-class Chainlink data is mapped onto two and, additionally, three well separated clusters in output space. Trustworthiness and continuity hardly differ, as seen in Figure 6.1. Kaski's approach usually cannot distinguish which topographic mapping provides the best depiction of cluster structures.

Nybo's Dilemma

In case of comparing topographic and topological mappings we have to consider the following arguments. Mappings that rely on SOM-like structures are often topological in the sense that pairwise dissimilarities among data objects in data space cannot be deduced from the configuration in output space, in case of uniformly distributed images in output space for instance. Thus well-separated clusters are



(a) trustworthiness



(b) continuity

Figure 6.1: Trustworthiness and continuity as a function of neighbourhood size. Two or three clusters of Chainlink data of Figure 3.6 cannot be distinguished Plots by Jan Kohlhof.

projected closely together. If we want to compare topographic and topological mappings the euclidean distance in output space penalizes SOM-like topological mappings when comparing ranks in data space and output spaceï¿½ by means of euclidean radial neighbourhoods.

Nybo et al. [NVK07] use geodesic distances along neighbouring neurons for more realistic neighbourhoods in case of SOM-like topological mappings. By doing so, the SOM clearly outperforms other methods (such as MDS, CCA and LLE) that have to settle for euclidean distances in data space and output space. Otherwise the SOM is not as good. Nybo et al. demonstrated the impact of applied distance functions on the outcome of empirical comparisions. This is evident because Kaski's rank-based measurements are merely definitions of quality that neither derive from the learning algorithm of the SOM nor from its competitors' learning algorithms.

Third-party Approval

Conventional (external and internal) evaluation measures usually do not assess the meaningfulness, interestingness and value of the knowledge obtained by the KDD process, i.e. the semantics of discovered patterns. In contrast to that, "third-party approval" methods rely on a model or a (human) expert that did not contribute to the creation of the data and was not involved in the process of knowledge discovery. More formally, these methods apply cost functions that

- are linked to the underlying domain (finance or medicine for instance),
- cannot be derived trivially from the features of the data set,
- act as subjective interestingness measures that are based on experts' belief in the data.

Segmenting financial data into homogeneous classes of similar acting entities, customers for instance, may be realized by k-means like methods that minimize the *sum-of-squared-errors minimum variance* criterion. The quality of the solution is appropriately assessed by means of raised profits, instead of cluster centroids and distances. The KDD process might uncover thousands of patterns, many of which may be uninteresting to the (human) expert because they represent common knowledge or lack novelty. The development of methods for assessment of discovered patterns' interestingness is a challenging task, e.g. the use of interestingness measures to guide the KDD process in order to reduce the search space. Patterns are expected to be interesting, for instance, if they confirm a hypothesis that the (human) expert wishes to validate.

6.2 Dispersion

As outlined in Section 6.1 ordinary topographical quality measures usually do not derive from the learning algorithms of interest. Topographical and topological mappings can hardly be compared. Therefore the dispersion measure is introduced as a novel method to assess the quality of any given topographic mapping. Theorem 3.16 states that the U-Matrix heights approximate the Voronoi tesselation

of the output space. Correct U-Matrices can be derived from cohesive mappings only, i.e. Voronoi cells of data objects from the same class are adjacent in output space. Dispersion quantifies cohesiveness by means of connected Delaunay graphs in output space.

Again, let $\mathcal{MD} \subset \mathbb{X} \times \mathbb{X}$ denote the Delaunay graph of input samples $\mathbb{X} = \{x_1, ..., x_n\}$ in output space, i.e. $(y_i, y_j) \in \mathcal{MD}$ iff y_i, y_j have adjacent Voronoi cells. A topographic mapping cohesively maps class C on the output space iff the relevant subgraph of $\mathcal{MD}$ is connected, i.e. $\mathcal{MD} \cap C \times C$ is connected. A topographic mapping is called cohesive iff each class $C_1, .., C_k$ is cohesively mapped onto $\mathbb{O}$ (cf. Section 3.7). Dispersion quantifies the class-wise (dis)connectedness of subgraphs of the Delaunay graph in output space. A class-sensitive weight function $w_C : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ is defined as follows:

$$w_C(x, x') = \begin{cases} 0 & : \quad \{x, x'\} \subseteq C \\ d_{\mathbb{D}}(x, x') & : \quad \text{else} \end{cases}$$

Dispersion $disp(C)$ of class $C \subset \mathbb{X}$ is quantified by the help of a minimum spanning tree (MST) on $\mathcal{MD}$ with edge weights w_C. Let $\triangleright_C \subset \mathbb{X} \times \mathbb{X}$ denote the parental relation on MST with $x \triangleright_C x'$ iff x is parental node to x'. An ancestor relation $\triangleright_C^*$ is then obtained by $x \triangleright_C^* x'$ iff $\exists l \in \mathbb{N}_0 \exists x'_1, ..., x'_l \in \mathbb{X}$ with $x \triangleright_C x'_1 \wedge x'_1 \triangleright_C x'_2 \wedge ... \wedge x'_l \triangleright_C x'$. Dispersion of class $C \subset \mathbb{X}$ follows as:

Formula 6.1

$$disp(C) = \sum_{x \in \mathbb{X}} \begin{cases} w_C(x, x') & : \exists x' : \; x' \triangleright_C x \;\wedge\; (x \in C \;\vee\; \exists x'' \in C \; x \triangleright_C^* x'') \\ 0 & : \; else \end{cases}$$

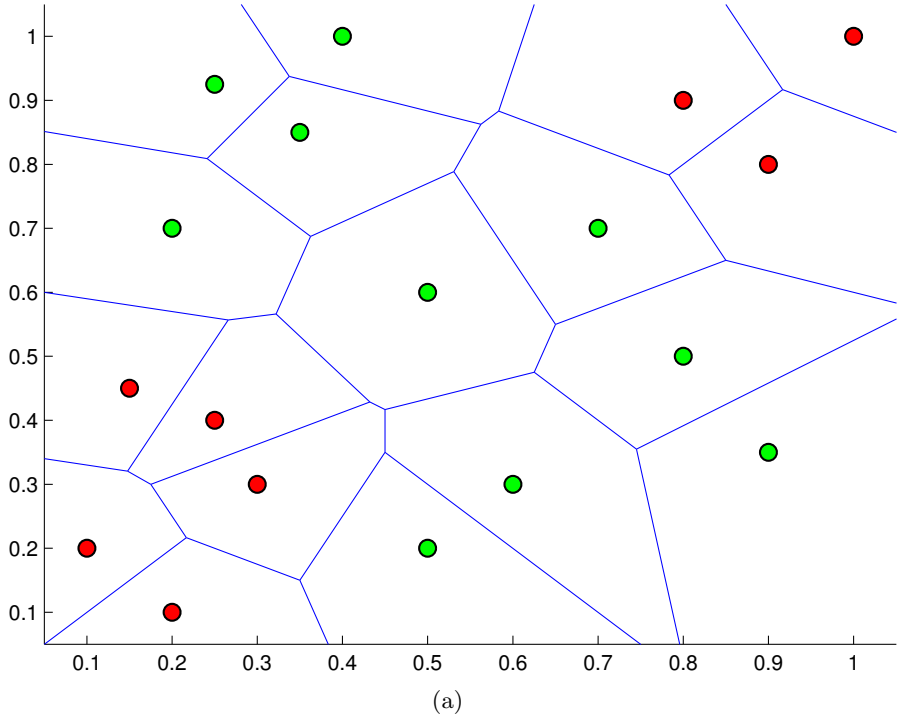
The overall dispersion $disp(C_1, .., C_k)$ of a topographic mapping is the sum of class-wise dispersions divided by mid with

$$disp(C_1, ..., C_k) = \frac{1}{mid} \sum_{i=1}^{k} disp(C_i)$$

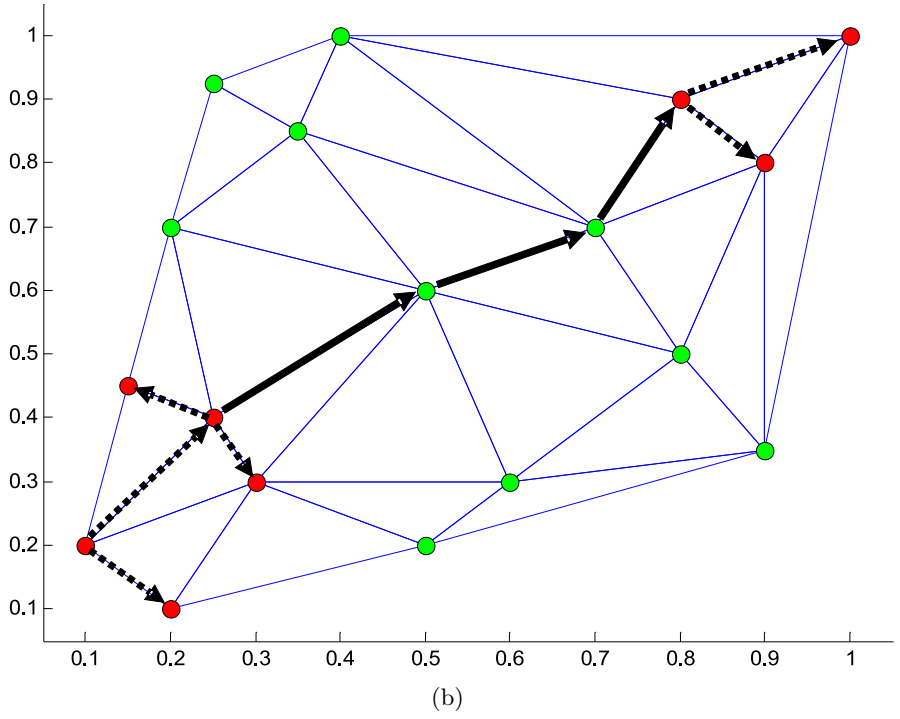
where $mid \in \mathbb{R}^+$ is the median of inter-class distances. Dispersion $disp(C)$ adds path lengths, measured by input space distances, when reaching input samples of class C on output space. Inner-class distances are not accounted for. Therefore, cohesive topographic mappings result in $disp(C_1, .., C_k) = 0$. Normalization by mid accounts for data-dependent levels of scaling, different cardinalities and data manifold structures.

6.3 Experimental Setup

The concepts of completeness and soundness are not sufficient in order to assess the quality of the learning algorithms with respect to topological mapping. The main aim of the experiments is to show that the conceptual advantage of the Swarm-Organized Projection method (SOP) translates into an actual performance advantage compared to well established topographic mapping techniques. The following methods are considered for empirical comparison: Self-Organizing Batch Map, Curvilinear Component Analysis, Stochastic Neighbour Embedding and Swarm-Organized Projection.



(a)



(b)

Figure 6.2: Dispersion of red class. (a) Red and green data objects mapped onto $\mathbb{R}^2$. Borders of Voronoi cells shown. (b) Resulting Delaunay graph with minimum spanning tree (MST) with regard to the red subgraph. Dotted arrows indicate zero edge weights between red vertices. The red subgraph is not connected. The MST bridges the green class. Solid arrows indicate non-zero edge weights between non-red vertices.

Parametrization

For each data set the algorithms were run 100 times using each time a different random initialization. By default, the euclidean distance was used as dissimilarity function. The parametrization of each algorithm was carefully chosen:

- The CCA software was realized by Vesanto et al [VHAP00]. 100 epochs were used in order to effectively decrease the amount of topographic distortion. Other parameters were used according to software defaults, i.e. linear annealing scheme and gaussian neighbourhood. The output space is planar by default.

- The Batch-SOM is applied on a 64×64 sized grid with gaussian neighbourhoods during 100 iterations. The neighbourhood radius was linearly decreased from $\sigma_{initial} = 48$ down to $\sigma_{final} = 1$. Initial neighbourhoods completely cover the output space. The output space is toroid by default.

- The t-SNE method was realized by van der Maaten [vdMH08] and is freely available. It is applied with software defaults, i.e. $k = 30$ perplexity. The output space is planar by default.

- The Swarm-Organized Projection (SOP) is used on a 64×64 sized toroidal grid with gaussian neighbourhoods. Further parametrization is not necessary.

Evaluation

Statistical tests are considered for each data set in order to assess the relation among different distributions of performance measurements originating from different topographic mapping algorithms (cf. Section 2.4). Let $M, M' \subset \mathbb{R}$ denote two finite multisets of performance measurements describing the outcome of a test series, i.e. several runs of two algorithms on the same data set. The statistical tests are usually designed as null hypothesis tests, at which the null hypothesis states that M, M' were drawn from the same distribution. The p-value of the test then indicates if the null hypothesis is to be rejected on a chosen significance level. This means that a certain topographic mapping algorithm is likely to produce better mappings because the corresponding performance measurements are likely to differ. The following performance measures are used for evaluation of topographic mappings' qualities:

- Dispersion quantifies the disconnectedness of projected clusters in output space.

- The classification accuracy of a 1-nearest neighbour classifier in output space serves as a coarse indicator of class separation. For each point of the image $\{y_1, ..., y_n\}$ a classification is derived from its nearest neighbours and compared against the true classification.

6.4 Data

In this section, a number of data sets for which the construction of a correct topographic mapping is hard to reach, are used to compare the quality of SOM, CCA, t-SNE and SOP. Except for the data sets Iris, Wine, GPD194 and SwissBanknotes, all data sets are taken from the public repository FCPS. The Fundamental Clustering Problem Suite [Ult] contains a number of data sets for benchmarking of clustering algorithms. Each data sets represents a certain problem that arbitrary clustering algorithms shall be able to handle when facing real world data sets. See Section B.1 for illustration.

Atom

The *Atom* data set 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 in $\mathbb{R}^3$ surrounded by a well separated, but sparse hull of 400 points (see Figure B.1). Both classes are not linearly separable and many algorithms cannot construct a cohesive mapping in $\mathbb{R}^2$. The core is located in the center of the hull, which makes it hard to solve for some methods based on averaging. 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 class.

Chainlink

The *Chainlink* data set consists of two clusters in $\mathbb{R}^3$. Together, both clusters form intricate links of a chain and therefore cannot be separated by linear decision boundaries. The rings are cohesive in $\mathbb{R}^3$, however, many topographic mappings are non-cohesive in $\mathbb{R}^2$. The data is crucial for the demonstration of several topographic mapping challenges: data on two well-separated manifolds at which global proximities contradict local ones in the sense that the center of each ring is closer to some elements of the other class than toward elements of its own class. Both rings are intertwined in $\mathbb{R}^3$, and have the same average distances and densities.

EngyTime

The EngyTime data [Bag02] contains more than 4000 points of two classes in $\mathbb{R}^2$. The classes overlap and cluster borders may only be defined using density information. There is no *empty space* between the clusters.

Iris

The *Iris* data set consists of three clusters in $\mathbb{R}^4$. It was introduced by Fisher [Fis36] that describes the geographic variation of Iris flowers. The dataset consists of 50 samples from each of three species of Iris flowers, namely Setosa, Virginica and Versicolor. Four features were measured from each sample: length and width of

sepal and petal. The setosa class is well-separated, whereas virginica and versicolor are slightly overlapping. The data was not transformed.

Swiss Banknotes

The *Swiss Banknotes* data set was introduced by Flury and Riedwyl [FR88]. It consists of six features measured on 100 genuine and 100 counterfeit old Swiss 1000-franc bank notes. The features are: length of the bank note, height of the bank note (measured on the left), height of the bank note (measured on the right), distance of inner frame to the lower border, distance of inner frame to the upper border and length of the diagonal. Robust normalization (cf. Section 2.4) is applied in order to prevent few features from dominating the obtained distances. See Figure B.6 for illustration on transformed features.

Wine

The Wine data [ACdV92] is a 13-dimensional, real-valued data set. It consists of chemical measures of wines grown in the same region in Italy but derived from three different cultivars. Robust normalization (cf. Section 2.4) is applied in order to prevent few features from dominating the obtained distances.

GPD194

The GPD194 data was published by Popescu et al. [PKM06] and contains 194 proteins, which belong to three distinct classes of proteins. No vector space axioms but pairwise dissimilarities are available. The dissimilarities are derived from the output of the BLAST algorithm [Alt97] for amino acid sequence alignment of proteins. See Chapter B.5 for details.

6.5 Results

Dispersion

The obtained dispersions of topographic mappings are summarized in Table 6.2 as mean values and standard deviations. The standard deviations may exceed the mean values because to the positive skew of the dispersions' distributions. The distributions were additionally analyzed with a two-sided Kolmogoroff-Smirnov statistical test [Smi48] because they cannot be assumed as being normally distributed. For resulting p-values see Section B.3. The distributions of obtained dispersion values are depicted in Section B.4.

- The CCA method fails to reproduce the most high-dimensional data sets: Wine, SwissBanknotes and GPD194 proteins. Furthermore, CCA often misrepresents the two rings of Chainlink data as three. CCA sufficiently reproduces the low-dimensional data sets EngyTime and Iris.

- The Batch-SOM outperforms the other methods on most data sets due to a carefully chosen parametrization. It fails, however, to reproduce the proximity structure of the Atom data. The Batch-SOM cannot not be applied on the GPD194 data.
- In contrast to that the t-SNE clearly fails to produce cohesive mappings on Chainlink, EngyTime, Iris and Swissbanknotes data. The software did not produce any results on dissimilarity data with a random initialization.
- The SOP produced the smallest dispersions on Chainlink (together with Batch-SOM), Atom and SwissBanknotes data. The obtained dispersion values never show a large deviation from the best method. SOP performs second best on the Wine data, and almost identically with Batch-SOM and CCA on the Iris data.

	CCA	Batch-SOM	t-SNE	SOP
Chainlink	0.99 ± 0.41	0.0028 ± 0.0282	$\mathbf{1.54 \pm 1.69}$	0.014 ± 0.14
Atom	0 ± 0	$\mathbf{2.14 \pm 0.79}$	0 ± 0	0 ± 0
EngyTime	2.62 ± 0.31	2.71 ± 0.46	$\mathbf{6.6 \pm 2.17}$	2.97 ± 0.45
Iris	0.013 ± 0.065	0.019 ± 0.056	$\mathbf{0.5 \pm 0.7}$	0.066 ± 0.1
SwissBanknotes	3.97 ± 2.55	1.03 ± 1.71	$\mathbf{5.89 \pm 3.02}$	0.49 ± 1.31
Wine	$\mathbf{15.1 \pm 8.11}$	0.24 ± 1.12	2.28 ± 1.98	1.57 ± 2.95
GPD194	$\mathbf{15.7 \pm 6.71}$	-	-	0.94 ± 1.09

Table 6.2: Mean values ± standard deviation of mappings' dispersions obtained with: CCA, Batch-SOM, t-SNE and SOP. Worst results are bold.

Accuracy

The percental classification accuracy of a 1-nearest neighbour classifier is summarized in Table 6.3. Due to software problems, the t-SNE software did not produce any results on dissimilarity data with a random initialization. The obtained accuracies are normally distributed. All methods perform alike on the EngyTime data. On the Chainlink data the Batch-SOM obtains the smallest accuracies. On the other data sets, both CCA and t-SNE method outperform the Batch-SOM and SOP method. The SOP method performs slightly better than the Batch-SOM.

	CCA	Batch-SOM	t-SNE	SOP
Chainlink	100 ± 0	97.8 ± 0.4	100 ± 0	99.6 ± 0.16
Atom	100 ± 0	97.2 ± 0.4	100 ± 0	99.4 ± 0.16
EngyTime	95.9 ± 0.1	95.3 ± 0.2	95 ± 0.2	95.2 ± 0.2
Iris	96 ± 0.6	83 ± 1.83	95.1 ± 0.7	85.3 ± 1.9
SwissBanknotes	97.5 ± 0.7	92.1 ± 1.2	94 ± 0.8	92.3 ± 1.3
Wine	92 ± 1.7	85.9 ± 1.7	94.6 ± 0.9	87.1 ± 1.7
GPD194	94.8 ± 1.6	-	-	94.7 ± 1.3

Table 6.3: Mean values ± standard deviation of 1-nearest neighbour classifier accuracy obtained with: CCA, Batch-SOM, t-SNE and SOP.

6.6 Conclusions

Standard approaches for assessment of topographic mapping techniques do not produce reliable results on different architectures. Due to this, SOM-like algorithms on toroidal grids are hardly comparable with distance-preserving projections on euclidean spaces. The dispersion measure quantifies the cohesiveness of arbitrary mappings by means of connected subsets of the Delaunay graph in output space. In terms of cohesiveness, the Swarm-Organized Projection (SOP) is almost as as good, or even better, than the best of its carefully parametrized competitor methods, namely CCA, t-SNE and Batch-SOM. These methods show severe misrepresentations of the class structure on several data sets. SOP does not.

Chapter 7

Knowledge Discovery on MicroRNA Data

This chapter demonstrates how the Swarm-Organized Projection is applied for analysis of data from molecular biology. The targeting structure of small RNA fragments, so-called microRNA, is examined in order to discover which kinds of genes are actually regulated by microRNA.

7.1 Molecular Biology

Genes and Gene Products

In cells of living organisms, a *gene* is a portion of DNA that contains both coding sequences that determine what the gene does, and non-coding sequences that determine when the gene is expressed (active). When a gene is expressed, the coding and non-coding sequences are copied in a process called transcription, producing a messenger RNA (mRNA) copy of the gene's information. This piece of RNA can then direct the synthesis of proteins via the genetic code. This procedure is usually referred to as *central dogma of molecular biology* [Cri70]. For illustration see Figure 7.1.

The molecules resulting from gene expression, whether RNA or protein, are known as *gene products*, and are responsible for the development and functioning of all living things. Sets of proteins interact and are involved in cellular processes, such as metabolism, signal transduction or RNA processing. Proteins can act in different cellular localizations, such as nucleus or membrane. Each protein has elementary molecular functions that normally are independent of the environment, such as catalytic or binding activities. Scientists study the kinds and amounts of mRNA produced by a cell to learn which genes are expressed, which in turn provides insights into how the cell responds to its changing needs. In order to infer how active a gene is, the amount of mRNA is measured, with microarray technologies for instance. A microarray [SSDB95] exploits the ability of a given kind of mRNA molecule to bind specifically to the DNA template from which it originated. By using an array containing many DNA samples, the expression

levels of thousands of genes within a cell are determined by measuring the amount of mRNA bound to each site on the array in a single experiment. The outcome of $n \in \mathbb{N}$ experiments is summarized in a n-dimensional real-valued expression vector for each gene. These vectors enable the retrieval of groups of genes with similar expression vectors, i.e. similarly behaving genes. See [Qua01] for details. Abnormal amounts of mRNA, and therefore gene product, can be correlated with disease-causing alleles, such as the over activity of oncogenes which causes cancer.

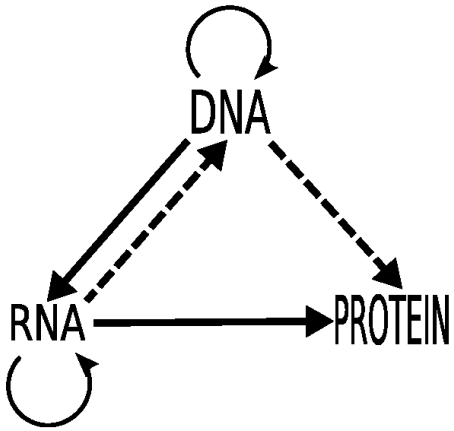


Figure 7.1: Central Dogma of Molecular Biochemistry [Cri70]: information is transferred from nucleic acids DNA, RNA to proteins.

MicroRNA

In genetics, microRNAs (miRNA) are single-stranded RNA molecules of 21 - 23 nucleotides (nt) in length, that regulate the stability or translational efficiency of target messenger RNAs [LFA93] [Ruv01]. See Figure 7.2 for illustration. Like normal RNA, miRNAs are encoded by genes from whose DNA they are transcribed. MicroRNAs are not translated into protein (non-coding RNA). Instead each primary transcript is processed into a short stem-loop structure called a pre-miRNA and finally into a functional miRNA. Mature miRNA molecules are partially complementary to one or more messenger RNA (mRNA) molecules. The function of miRNAs appears to be in gene regulation. For that purpose, a miRNA is complementary to a part of one or more messenger RNAs (mRNA). MicroRNAs pair to 3'UTRs (untransformed regions) of mRNAs to direct their posttranscriptional repression. The pairing of the miRNA to the mRNA then blocks protein translation.

Several miRNAs have been found to be linked with some types of cancer [HTH+05] [OWZ+05] [LGM+05]. Evidence has been assembled that indicates that miRNAs are associated with cancer because of deregulation. Genome wide studies demonstrate that miRNA genes are frequently located at cancer-associated genomic regions. This suggests that miRNA might be attributed to a new class of genes involved in human tumorigenesis. Down regulation of miRNA is observed frequently in cancer samples. For example, miR-15 and miR-16 were down regulated in about

68% of B-cell chronic lymphocytic leukemia (CLL) cases [CLS+04]. See Figure 7.3 for illustration.

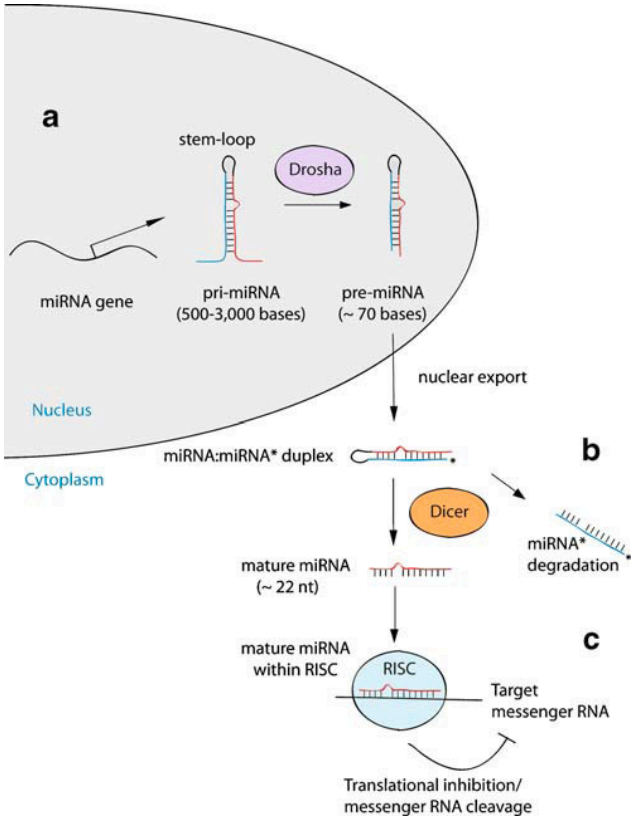


Figure 7.2: The biogenesis and function of miRNAs [SMC08]. (a) Primary miRNAs are transcribed from miRNA genes. (b) The pre-miRNA is cleaved by the ribonuclease dicer to generate a short RNA duplex in the cytoplasm. (c) The miRNA can bind to the target mRNA by base pairing, causing inhibition of protein translation and/or degradation of the target mRNA.

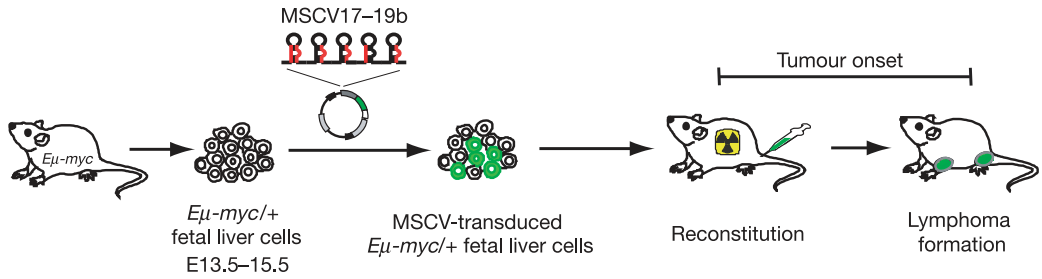


Figure 7.3: MicroRNA as an oncogene for mice [HTH+05]: murine stem cell virus (MSCV) causes liver cells to express *mir-17-19b* which leads to cancer.

Gene Target Prediction

MicroRNAs pair to the messages of protein-coding genes to direct the posttranscriptional repression of these mRNAs. Grimson et al. [GFJ+07] proposed the Targetscan method for prediction of pairing of microRNA with arbitrary mRNA and, therefore, regulation of genes. For each miRNA-mRNA pair there are five features that contribute to a meaningful estimation of posttranscriptional repression:

- It has been experimentally suggested that nucleotides immediately flanking the binding sites were highly enriched for adenine and uracil content relative to the non binding sites.
- Selective depletion of seed-matching sites in messages highly expressed in the same tissues as the miRNAs implies frequent targeting.
- Watson-Crick pairing to miRNA nucleotides 12-17, especially nucleotides 13-16, was most associated with down regulation of mRNA activity.
- Binding sites preferentially reside in the 3' UTR (untransformed region) but not too close (about 15 nt) to the stop codon.
- Site depletion in messages preferentially coexpressed with miRNAs is more severe near the ends of long UTRs than near the center.

For each feature a numerical score was heuristically derived. These scores were combined into a single context score using linear regression in order to predict the fold change of experimentally derived down regulations of mRNA. For details see [GFJ+07].

7.2 Gene Ontology

The Gene Ontology (GO) is a major bioinformatics project hosted on an internet platform [ABB+00]. It was created to describe and unify the attributes of genes and gene products across all species using a controlled vocabulary. The aims of the GO project are threefold: (1) to maintain and further develop its controlled vocabulary of gene and gene product attributes, (2) to annotate genes and gene products, and assimilate and disseminate annotation data, and (3) to provide tools to facilitate access to all aspects of the data provided by the GO project.

Architecture

The GO covers three domains, which means it is made up of three separate ontologies.

- **Cellular component**: the parts of a cell or its extracellular environment.
- **Molecular function**: the elemental activities of a gene product at the molecular level, such as binding or catalysis.
- **Biological process**: operations or molecular events with a defined beginning and end, such as mitosis or apoptosis.

Each node of the ontology represents an ontology term, at which terms are used to represent biological concepts. If two terms have some relationship, an edge is drawn from one to the other. Each of these ontologies is a connected directed acyclic graph (DAG), which means that a child (more specialized) term can have multiple parents (less specialized terms). There is only one root node in each ontology. The GO only uses *is_a* and *part_of* relationships. The *is_a* relationship indicates that the term in the in-node of the edge is a subset of the term in the out-node. The *part_of* relationship denotes that the in-node term has the out-node term as one of its parts. If an instance of a child concept is a complete instance of the parent concept, the is_a relation is used. Otherwise, if a term describes a concept being only a portion of the parent concept, the part_of relation is used. More specialized but less frequently occurring concepts exist. See Figure 7.4 for illustration. For example, the term "ATP-dependent DNA helicase" is a child of several less specific concepts: "DNA helicase", "ATP-dependent helicase" and "DNA-dependent adenosine triphosphate". Several gene products of the MCM and CDC family are annotated with this term.

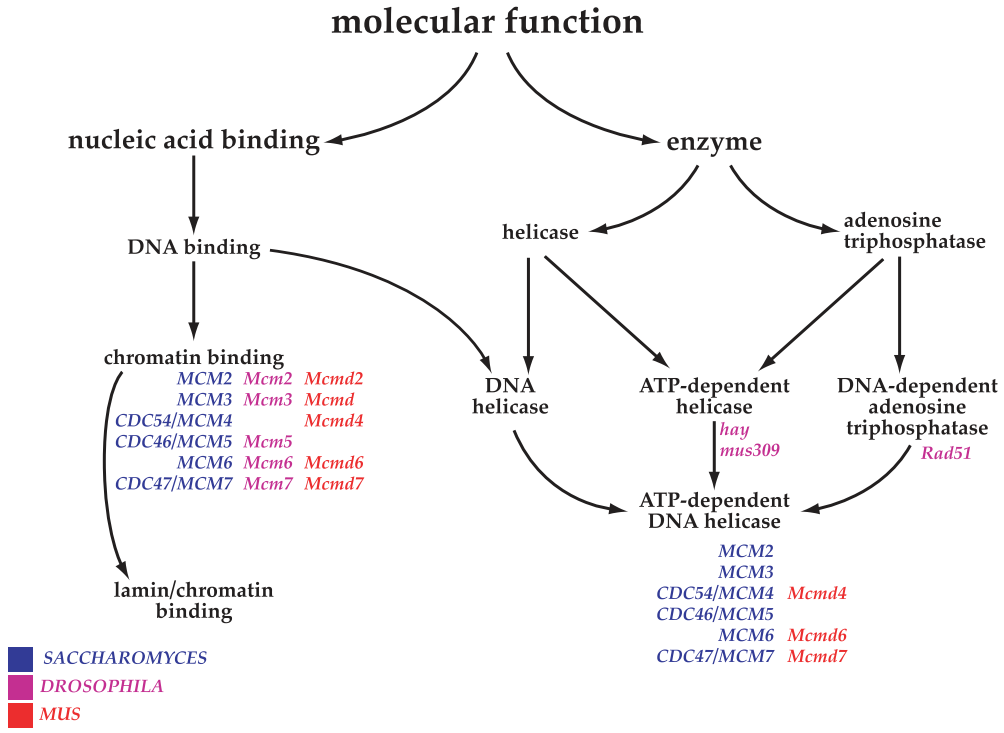


Figure 7.4: Small excerpt of the Gene Ontology [ABB+00]: the "molecular function" ontology with annotations from species saccharomyces (yeast), drosophila (fly) and mouse.

Annotations

The operating principles of proteins in a living cell have been studied for many different organisms in the last decades. These informations are shared as scientific publications. However, this textual procedure does not allow a fast, computer-driven analysis of these protein-related descriptions. The Gene Ontology (GO) itself does not contain gene products, but terms corresponding to biological concepts from the broad field of cellular components, molecular functions and biological processes. The Gene Ontology Annotation project [CMB+04] aims at capturing data about gene products (proteins). Therefore, the GO annotates genes and gene products, respectively, using GO terms. GO terms are assigned to gene products using a combination of high-quality electronic mappings and manual curation, the latter of which employs a team of biologists.

Let $\mathcal{GP}$ denote the set of genes (and gene products), and $\mathcal{T}$ is the set of GO terms. Formally, an annotation $a : \mathcal{GP} \rightarrow 2^{\mathcal{T}}$ assigns sets of GO terms to given gene identifiers. The so-called *true path rule* states that the path from a child term all the way up to its top level parent(s) must always be biologically valid. Furthermore, genes annotated with a child term are also annotated with its parental terms. This means that each gene product is annotated with the root node of each ontology.

7.3 Gene Ontology for Analytical Methods

The discovery of functional classes within collections of genes often relies on similarity functions in order to determine which genes most likely belong to the same class. Genes are regarded as similar if their sequences of nucleotides are similar, or experimental data shows similar behavioural patterns [Qua01]. When experimental data is not available, genes' similarity is quantified by means of the Gene Ontology and its annotations. The Gene Ontology (GO) has been used to facilitate the discovery of functional groups within collections of genes

Semantic similarity functions of gene products quantify the functional overlap with respect to molecular biology. Small semantic similarities show few functional overlaps, whereas large semantic similarities denote many functional overlaps. Cellular functions can only be understood by considering complex protein interactions, which has been hardly realized. Therefore, semantic similarity usually is an unknown quantity for a given pair of genes. Gene Ontology (GO) together with gene annotations offer a huge amount of information concerning genes and their biological context. The GO acts as a database that offers functional categories for gene products, and multiple hierarchical relations among these categories. Thus, the GO may be used to estimate the semantic similarity of pairs of gene products by means of annotations and the GO graph. Two genes are considered as similar if their annotated GO terms are similar.

Term Similarities

Information-theoretic methods were originally described for the analysis of any corpus of text [Res95] [JC97] [Lin98] and were adapted for use with GO by Lord et al. [LSBG03b]. Let $\mathcal{T}$ be the set of GO terms, and $S(t_i, t_j) \subset \mathcal{T}$ denotes the

set of parental terms shared by terms $t_i, t_j \in \mathcal{T}$ since GO allows multiple parents for each term. The most informative common ancestor $mia : \mathcal{T} \times \mathcal{T} \to \mathcal{T}$ of two terms is then defined as

$$mia(t_i, t_j) = \arg \min_{t \in S(t_i, t_j)} p(t)$$

where $p : \mathcal{T} \to [0, 1]$ denotes the probability of gene products to be annotated with a certain GO term and its children. These methods are based on the assumption that the more information two terms share, the more similar they are. The information content of a GO term $t \in \mathcal{T}$ is defined as:

$$IC(t) = -\log(p(t))$$

Resnik's measure [Res95] calculates the similarity s_R of two terms t_i, t_j by using only the information content of the most informative common ancestor. Resnik's measure assumes maximum values for each term t_i in the sense that $s(t_i, t_i) \geq s(t_i, t_j)$ for all terms t_j. Resnik's measure roughly indicates the depth of the most informative ancestor, such that large similarities can only occur between more specialized terms at the lower levels of the GO. Empirical evaluation of Resnik's measure can be found in literature: Pesquita et al. [PFB+08] found that s_R outperforms the other semantic measures according to its correlation with gene products' sequence similarity. Sevilla et al. [SSP+05] assessed the correlation between genes' expression data and semantic similarities, whereas s_R shows the best results.

$$s_R(t_i, t_j) = IC(mia(t_i, t_j))$$

Lin [Lin98] proposed a measure of similarity s_L that takes into consideration the IC values of terms t_i, t_j in addition to their most informative common ancestor. Lin's measure has a limited range of $[0, 1]$ with $s_L(t, t) = 1$ for all terms $t \in \mathcal{T}$. Lin's measure does not indicate whether two terms are located near the root node or at the lower levels of the ontology.

$$s_L(t_i, t_j) = \frac{2 \cdot IC(mia(t_i, t_j))}{IC(t_i) + IC(t_j)}$$

Jiang and Conrath [JC97] proposed a semantic distance d_{JC} based on information content. The distance d_{JC} does not indicate whether two terms are located near the root node or at the lower levels of the ontology. Empirical evaluation of d_{JC} can be found in literature: Couto et al. [CSC07] investigate the correlation between semantic measures and shared protein families, at which d_{JC} obtained the strongest correlations.

$$d_{JC}(t_i, t_j) = IC(t_i) + IC(t_j) - 2 \cdot IC(mia(t_i, t_j))$$

Gene Similarities

For research purposes, it is of interest to determine semantic similarity between genes (and gene products) rather than GO terms per se. Combination of these measures is needed when a gene product was annotated with several terms. According to Lord et al. [LSBG03a] the average similarity between all terms should be used, because a gene product will generally have all of the roles attributed to it by the annotators at the same time. However, the average is prone to underestimate the true similarity of gene products, e.g. the self-similarity usually does not assume a maximum value. If the maximum is used instead, the similarity is overestimated since it is enough that the two gene products share one term for the similarity to assume its maximum value.

Fröhlich et al. [FSSZ06] have realized a method to compare gene products $g, g' \in \mathcal{GP}$ with annotated lists of GO terms $a(g) = \{t_1, ..., t_n\}$ and $a(g') = \{t'_1, ..., t'_m\}$. Let $s_\mathcal{T}$ be a similarity function for comparison of GO terms. Then a way of comparing g and g' is to assign each term of the smaller of both lists to exactly one term in the longer one, such that the sum of term similarities is maximized. Formally, by using permutation π the gene product similarity $s_{\mathcal{GP}}$ is given below. The computation of this *optimal assignment* (OA) problem corresponds to the solution of the classical maximum weighted bipartite matching problem in graph theory. This is realized by means of the *GOSim* software [FSPB07].

$$s_{\mathcal{GP}}(g, g') = \begin{cases} \max_\pi \sum_{i=1}^{n} s_\mathcal{T}(t_i, t'_{\pi(i)}) & : \text{ if } m > n \\ \max_\pi \sum_{i=1}^{m} s_\mathcal{T}(t_{\pi(i)}, t'_i) & : \text{ else} \end{cases}$$

Vector space methods operate on a binary valued annotation matrix where each column represents the annotations of a gene (product). 1 means the presence of the GO term in the gene's annotation and 0 represents its absence. For example, the cosine similarity between columns is calculated to obtain pairwise gene similarities. See [CMB07] [PKM06] for details. The term overlap approach represents genes as sets of annotated GO terms. The *term overlap similarity* is the (normalized) size of the sets' intersection. See [MP08] for details.

Over-Representation Analysis

Results derived from experimental gene analysis are often interpreted by manually reviewing the function of each gene based on literature or database searches, or by prior familiarity with the gene and a plausible link to the biological context. This ad hoc process is both time-consuming and prone to user bias. Instead, automated approaches have been developed to facilitate the knowledge conversion process on the genetic domain. Knowledge conversion is the process of putting found patterns down on a symbolic formalism (cf. Section 2.3). Here, found patterns refers to sets of genes (and gene products) that have been found to be meaningful, for example by microarray experiments or cluster analysis. A symbolic formalism abstracts such sets by means of GO terms that are significantly overrepresented (or underrepresented) in the genes' annotations. Such statistical methods are collectively known as over-representation analysis (ORA). ORA deals with the question what

GO terms are represented in the gene list more often than expected by chance. The most common approach to evaluate this statistically is the hypergeometric test (or variants such as Fisher's exact test) that calculates the probability of seeing at least a particular number of genes containing the biological term of interest in the gene list.

Let $n \in \mathbb{N}$ be the number of genes in the basic population, at which $n_t < n$ are associated with a certain GO term $t \in \mathcal{T}$ of interest. If $m < n$ genes were randomly selected from the basic population without replacement, the probability of seeing exactly $m_t < m$ genes associated with term t can be modeled by means of the hypergeometric distribution [BKK+07]. The probability of seeing m_t or more genes containing term t in a random gene list of m genes can be calculated as the cumulative probability:

$$p = 1 - \sum_{i=0}^{m_t} \frac{\binom{n_t}{i}\binom{n-n_t}{m-i}}{\binom{n}{m}}$$

This is a one-sided test for over-representation. Usually, a term is considered significantly enriched if its p-value is less than a certain threshold p_0 after adjusting for multiple hypothesis testing. For implementation details see the GeneTrail [BKK+07] and GOstat [BS04] software.

In order to cope with the dependencies resulting from the hierarchical structure of GO, the standard hypergeometric approach was enhanced by Grossmann et al. [GBRV07]. Let $pa(t) \subset \mathcal{T}$ denote the parents of term t. For the sake of simplicity suppose that t has only a single parent. The overall cardinalities of the gene list and basic population are denoted as $m_{pa(t)}, n_{pa(t)}$. The probability of m_t or more genes being annotated with term t follows as:

$$p = 1 - \sum_{i=0}^{m_t} \frac{\binom{n_t}{i}\binom{n_{pa(t)}-n_t}{m_{pa(t)}-i}}{\binom{n_{pa(t)}}{m_{pa(t)}}}$$

An empirical estimate of p-values is obtained as follows [Fis35]: For an experimentally derived gene list $\{t_1, ..., t_n\} \subset \mathcal{T}$ the permutation approach creates random lists of size n from the underlying set of genes $\mathcal{GP}$. This permutation procedure is repeated, e.g. 100000 times. An empirical p-value of over-representation can then be calculated for each term t_i with $i = 1, ..., n$ as fraction of times its frequency in the random gene lists is equal to or greater than that seen in the experimentally defined gene list.

Related Works

Cluster analysis methods may be applied on gene products $\mathbb{X} = \{x_1, ..., x_k\} \subset \mathcal{GP}$ by means of semantic (dis)similarities $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$. Speer et al. [SFSZ05] proposed a straightforward approach for clustering genes (and gene products) based on spectral clustering [CSTK02] and the semantic distance d of Jiang and Conrath [JC97]. For each gene product $x_i \in \mathbb{X}$ the empirical feature vector $\phi(x_i) = (d(x_i, x_1), ..., d(x_i, x_k))$ is used to derive a kernel function used for spectral clustering. This method is claimed to produce better results than k-means and

single linkage clustering. The impact of the empirical feature encoding remains unclear since it is used for all clustering methods.

Cheng et al. [CCM+04] proposed a method for integration of GO-based semantic dissimilarities and expression based distances of genes in microarray experiments for the purpose of cluster analysis. This is supposed to accentuate genes with both similar expression profiles and similar biological characteristics. To quantify the similarity of two GO nodes, the number of shared edges in their respective paths to the root node is used. Only the edges that are common to both paths are relevant. Each edge is weighted according to its depth in the graph, using weighting parameter $\lambda \in (0, 1)$. The weight for the partial path consisting of p edges, is the sum of weights $w_p = \sum_{i=0}^{p} \lambda^i$. The pairwise gene similarity follows as as maximum similarity among all possible pairs of annotated GO terms. The euclidean distances between genes' expression profiles are normalized in order to match the range of GO-induced similarities. The dissimilarity of genes finally follows as arithmetic mean of GO-dissimilarities and genes' distances. Hierarchical clustering identifies groups of genes that are similar in both functional annotations and expression profiles. A total of 29 genes expressed on 5 time points were analyzed. Obviously, this approach does not consider the information content of nodes.

Ghous et al. [GKCS08] proposed the use of Kernel Principal Component Analysis [SSM98] for topographic mapping of genes. This is based on the kernel trick which transforms the set of genes $\mathbb{X}$ into a feature space, at which the kernel function $k(x, y) = \varphi(x) \cdot \varphi(y)$ computes the inner product of transformed genes $\varphi(x), \varphi(y)$. In Kernel Principal Component Analysis (KPCA) the principal components are the eigenvectors of the kernel matrix. Gaussian and linear kernels were used. Each gene is represented as an annotation vector, i.e. 1 representing the presence of a GO term and 0 otherwise. A total number of 69 genes from 5 classes were analyzed. The linear kernel produces no trustworthy results. The results of the gaussian kernel are highly sensitive to parametrization, i.e. kernel width.

7.4 Knowledge Discovery on microRNA

This section presents a novel analytical approach concerning the question whether there are differentiated, functional types of genes targeted by microRNA (miRNA) or not. The approach is based on finding clusters (i.e. homogeneous classes) among those genes that are regulated by miRNA. The obtained clusters hopefully contain genes that are overly annotated with meaningful biological categories, i.e. terms of the Gene Ontology (GO). These functional categories enable novel insights into the regulation abilities of miRNA. See Figure 7.5 for illustration of the stepwise approach.

1. TargetScan [GFJ+07] predicts for each miRNA a set of potentially regulated genes and corresponding score values. The distribution of all obtained score values indicates which genes are supposedly regulated by miRNAs, and which are not. The regulated genes are considered for further examination.

2. Information concerning biological function is provided for each gene by means of annotated GO terms. According to this, pairwise semantic dissimilarities

are derived. These dissimilarities quantify the genes' functional difference and, furthermore, enable the retrieval of homogeneous subsets of genes with similar biological features.

3. The Swarm-Organized Projection (SOP) maps the miRNA-regulated genes onto a two-dimensional output space. The proximity structure of these genes is visualized by the Generalized U-Matrix method.

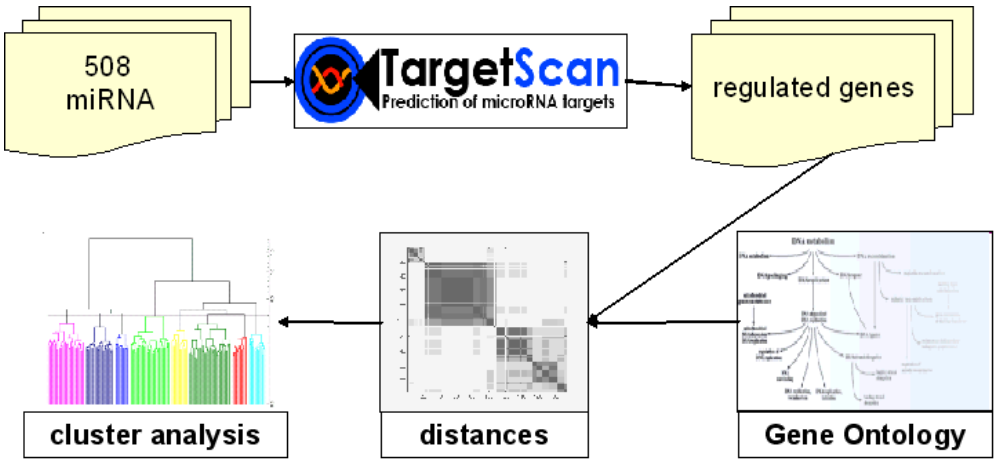


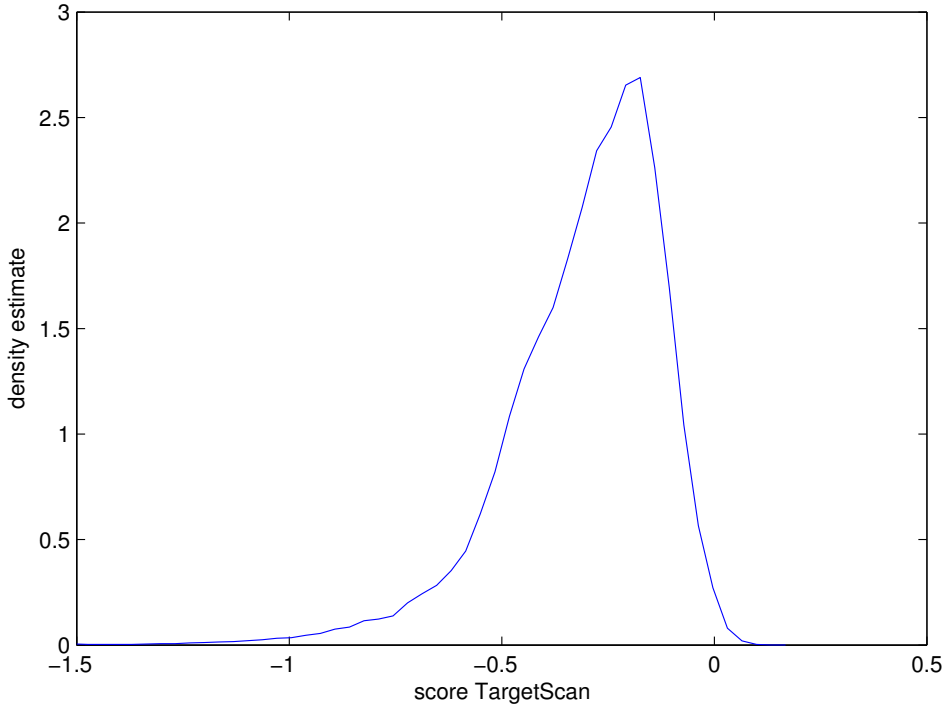
Figure 7.5: Knowledge discovery reveals which functional classes of genes are regulated by miRNA.

Regulated Genes for microRNA

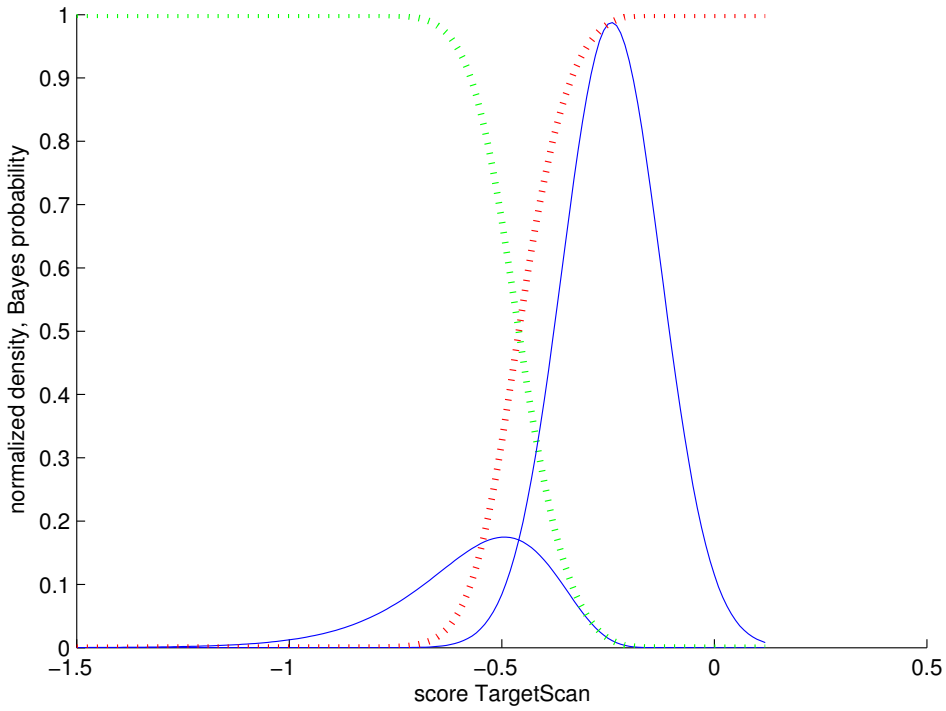
The "mirbase" database [GJGvD+06] provides information about sequence data and structural features of miRNA that have been discovered so far. The miRNA set contains 508 identifiers that are recognized by both mirbase [GJGvD+06] and Targetscan [GFJ+07].

For each miRNA the Targetscan algorithm predicts a set of possibly regulated genes with corresponding score values (cf. Section 7.1). Low scores predict high probabilities for being regulated by miRNA. High scores predict genes that are hardly regulated by miRNA. See Figure 7.6(a)) for the distribution of scores. As suggested by Alfred Ultsch, the distribution of scores can be thought of as a mixture of two components. The bigger part of scores follows a normal distribution U that represents unregulated random bindings of genes's mRNA and miRNA. A smaller part of scores follows a distribution R that derives from binding forces that do not originate from random noise, i.e. regulations of genes. Due to this considerations, TargetScan scores are modeled by means of a mixture model with normal distribution U and log-normal distribution R. The model is obtained by means of the EM algorithm [DLR77]. For a gene with score $s \in \mathbb{R}$ the Bayes probability [Bay63] for being regulated is

$$p(R|s) = \frac{p(s|R) \cdot p(R)}{p(s)} \quad \text{with} \quad p(s) = p(s|R) \cdot p(R) + p(s|U) \cdot p(U)$$



(a)



(b)

Figure 7.6: (a) Density estimate of TargetScan scores. (b) Log-normal/normal mixture model (blue) for TargetScan scores. Bayes probability for regulated genes (green) and unregulated ones (red).

at which $p(R), p(U)$ denotes the a priori probabilities for (un)regulated genes, and conditional probabilities $p(s|R), p(s|U)$. See Figure 7.6(b) for illustration. For further processing, genes g_i are selected whose regulation probability $p(R|s_i)$ exceeds a certain threshold p_0 and, furthermore, that are reported by Targetscan at least $n_0 \in \mathbb{N}$ times. In this case, gene products whose score is below -0.58 have at least a probability of $p_0 = 90\%$ for being regulated. For $n_0 = 2$ the remaining set of gene products $\mathbb{X}$ then consists of 1555 identifiers.

Gene Similarity

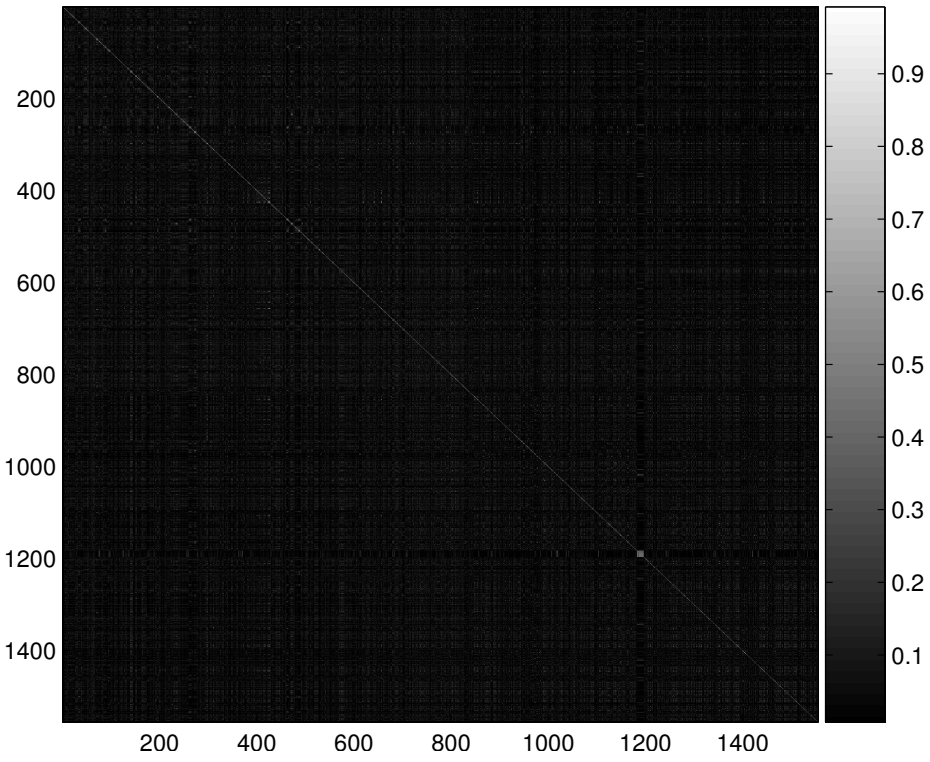
For genes $\mathbb{X} \subset \mathcal{G}P$ meaningful pairwise similarities were derived by means of Gene Ontology (GO) and GO annotations. Here the "biological process" ontology is applied. The approach of Resnik [Res95] was used to determine semantic similarities among GO terms. Terms located in the upper part of the GO cause small similarities. Terms in the lower part of the GO represent less general biological concepts and can therefore cause large similarities. The "optimal assignment" method [FSSZ06] combines the annotated GO terms' similarities into a meaningful similarity measure $s : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ for genes. The distribution is right-skewed, i.e. most similarities are small. See Figure 7.7 for illustration.

Using Resnik's measure is beneficial for further analysis. The GO terms located near the GO root node do not contain much information. Genes mainly annotated with these terms might distort further analysis because they add another source of noise to the data. For each gene $x \in \mathbb{X}$ the self-similarity $s(x, x)$ is the assumed maximum similarity that serves as an indicator for its information content. This is not achieved when using Lin's measure or the distance of Jiang and Conrath, because the main diagonal of the (dis)similarity matrix is a fixed value in these cases. The self-similarities $\{s(x, x) : x \in \mathbb{X}\}$ among genes are approximately normally distributed. See Figure 7.8 for illustration. Small self-similarities refer to genes whose annotated GO terms are located near the root node, which is referred to as upper part of the GO. Large self-similarities refer to genes where annotations are located in the lower part of the GO and, therefore, refer to more special biological concepts. Therefore, genes with small self-similarities $s(x, x) < s_0$ are dismissed for further analysis. Here $s_0 = 0.47$ is chosen as the mean value of the normal distribution which means that 749 genes are kept that are predominantly annotated at the lower part of the GO. Over-representation analysis with the GeneTrail method [BKK+07] reveals the most significant concepts among annotated GO terms to be related with cellular metabolic processes as expected. See Figure C.2 for illustration.

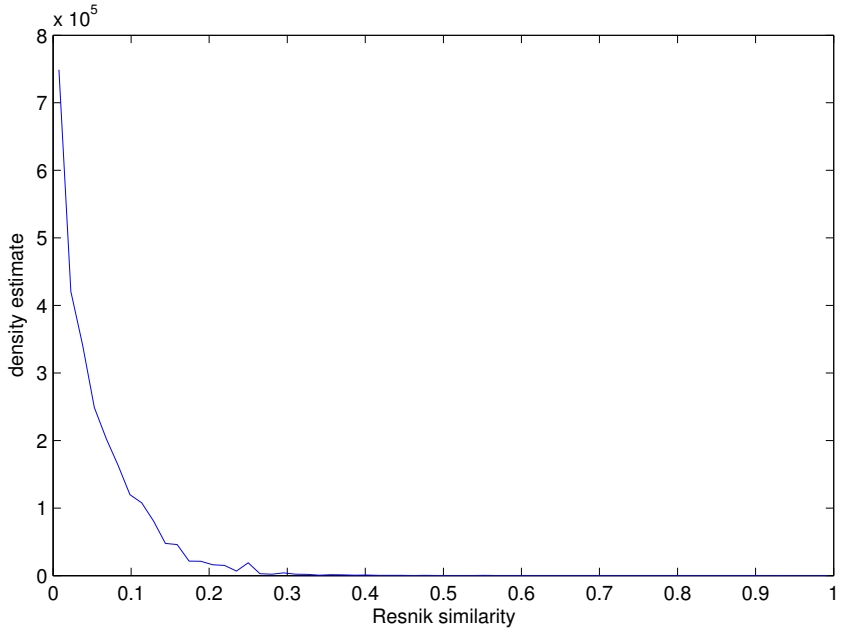
Gene Distance

A meaningful semantic dissimilarity function $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ among genes is obtained by generalizing the semantic term distance of Jiang and Conrath [JC97] to operate on genes. The distance of two genes $x, y \in \mathbb{X}$ is the difference between their inter-gene similarity and their self-similarities:

$$d(x, y) = s(x, x) + s(y, y) - 2 \cdot s(x, y)$$

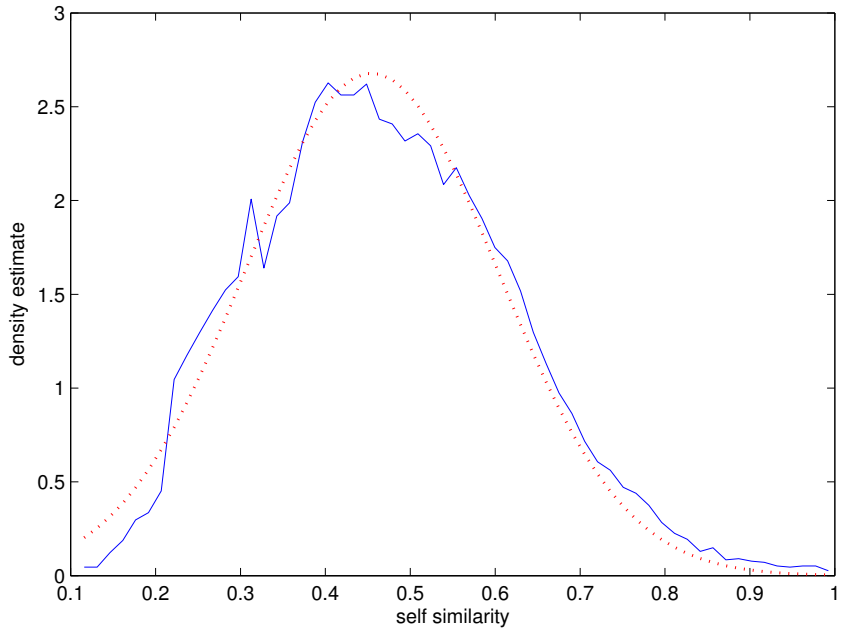


(a)

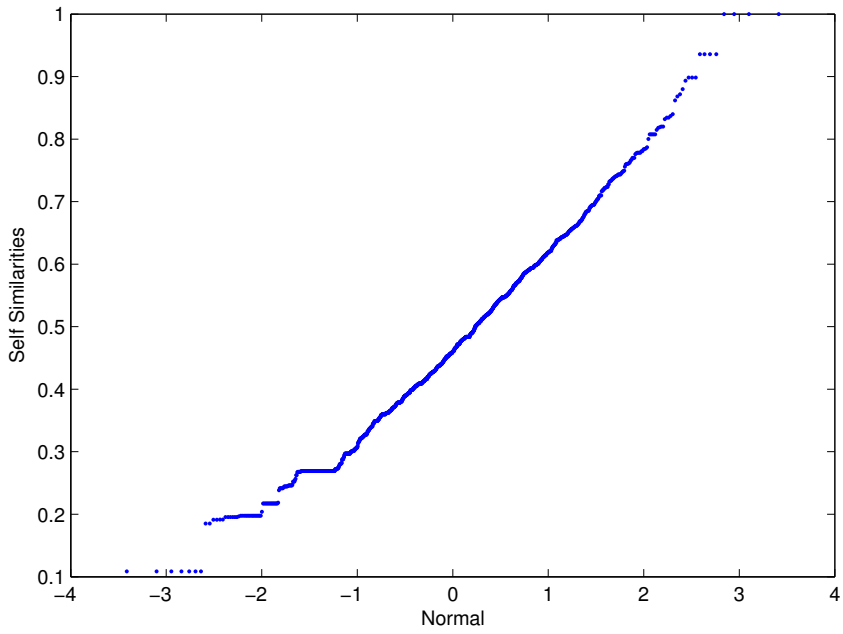


(b)

Figure 7.7: Resnik's semantic similarities among regulated genes.



(a)



(b)

Figure 7.8: (a) Density estimate of of genes' self-similarities suggests: self-similarities are normally distributed. (b) Quantiles of normal distribution against quantiles of genes' self-similarities.

It is based on the gene similarity $s : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ that was obtained from the optimal assignment method [FSSZ06] operating on Resnik's term similarity [Res95]. The gene distance $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ is inversely proportional to the gene similarity s. The gene distance approximately follows a normal distribution. See Figure 7.9 for illustration. Contrary to the gene similarity, the gene distance does not indicate if two genes are predominantly annotated to the lower or to the upper part of the GO. To face this, genes from the upper part of the GO have been removed according to their self-similarities.

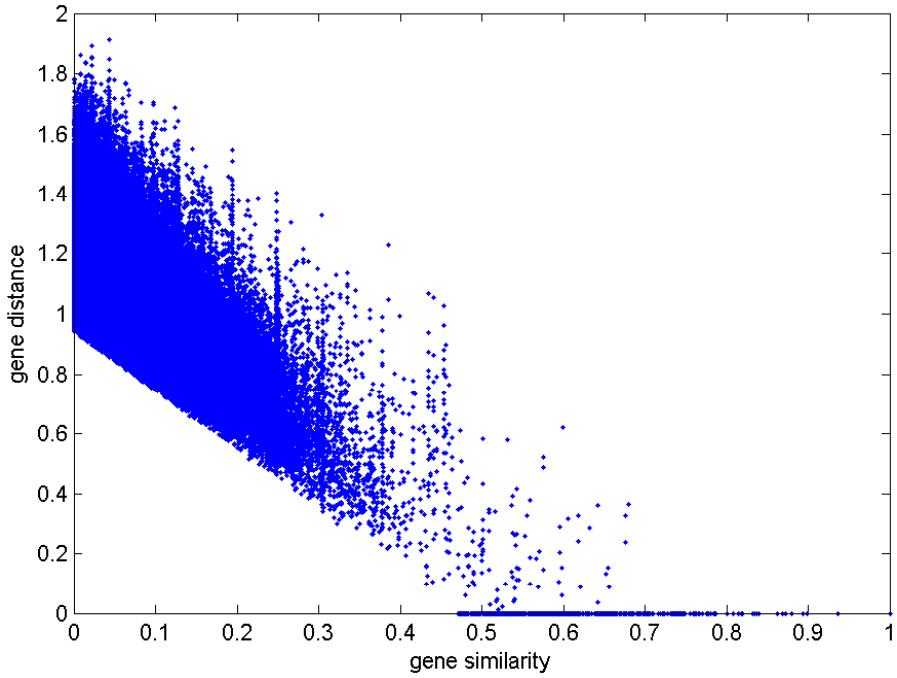


Figure 7.9: Gene distance against gene similarity.

Clustering Genes with Swarm-Organized Projection

The Swarm-Organized Projection (SOP) is applied directly on genes' distance measure d. The genes are mapped onto a 64×64 sized grid. The neighbourhood radius σ is not decreased to a final value of 1. Instead a final radius of $\sigma = 8$ forces SOP agents to assemble in dense piles and reveal the cluster structure more clearly. No additional parametrization is necessary. On top of the obtained mapping, the distance structure of the genes is then visualized by means of the Generalized U-Matrix that was introduced in Section 4.4. The U-Matrix method clearly depicts 8 classes of genes that are regulated by miRNA molecules, see Figure 7.10.

Over-representation analysis was applied on each class in order to assign meaningful biological concepts to these sets of genes. Personal communication with Christian Pallasch confirms the novelty and interestingness of the discovered structures in miRNA-regulated genes. The exact composition of the clusters is to be published, yet.

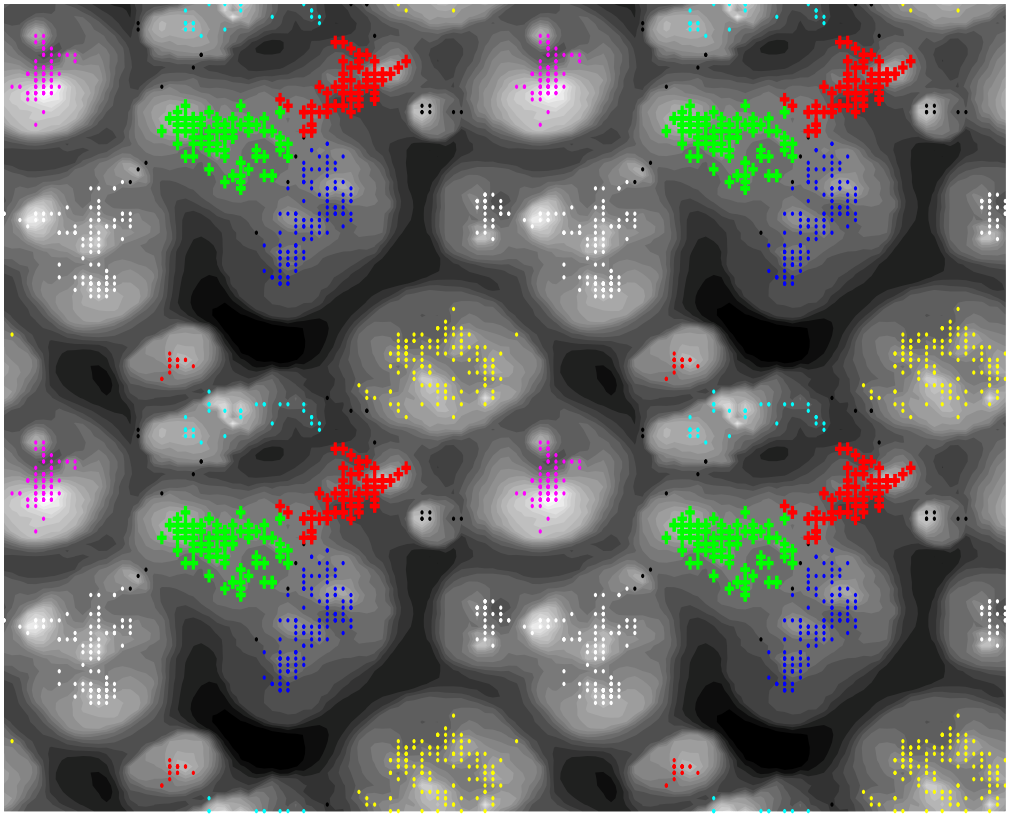


Figure 7.10: Tiled display: SOP maps genes onto two-dimensional toroid output space. Generalized U-Matrix depicts larger distance between genes as darker shades of gray. Eight classes can be distinguished, black dots depict unclassified genes.

7.5 Evaluation

Over representation Analysis (ORA) is used for evaluation of obtained clusters. A class $C \subset \mathbb{X}$ is considered as meaningful if its annotated biological categories are highly overrepresented (or underrepresented) in comparison with the reference set of genes $\mathbb{X}$. In this context, GO terms represent biological categories. ORA methods assign probability values $p : \mathcal{T} \rightarrow [0, 1]$ to GO Terms in order to quantify the probability for terms being annotated to genes of class C by chance (cf. Section 7.3). GO terms that are highly overrepresented (or underrepresented) in class C are assigned small p-values. Small p-values therefore suggest that terms are meaningful for the biological context of genes in class C. Thus the p-value serves as an indicator for biological relevance. A more convenient indicator for biological relevance of classes is the distribution of negative logarithmized p-values, i.e. negated exponents of probability values are used instead of raw probabilities. Here, the probabilities are restricted to non-trivial values, i.e. with $p < 0.05$ in order to discard noise. The GeneTrail algorithm [BKK$^+$07] determines the probabilities $p : \mathcal{T} \rightarrow [0, 1]$ of GO terms for occuring in C by chance compared with the reference set of genes $\mathbb{X}$ (cf. Section 7.3).

SOP is compared against Ward's clustering [War63] which can used with pairwise gene distances. Using MDS [Tor52] the dissimilarity data is embedded into a 358-dimensional euclidean space. Thus, the k-means vector quantization method [Mac67] is applied for unsupervised classification. Both methods are forced for produce $k = 8$ classes. Results are summarized in Table 7.1. On an average the classification obtained with SOP leads to significantly smaller p-values below $\alpha = 0.001$ level. This means that GO terms are more overrepresented in clusters obtained with SOP.

	k-means	Ward	SOP
average	4.46	5.06	5.31
maximum	78.00	71.17	79.39
p-value KS test against SOP	$2.23 \cdot 10^{-16}$	$2.62 \cdot 10^{-04}$	1

Table 7.1: Comparison of negated log-probabilities' distributions: SOP leads to more relevant classes than k-means and Ward's clustering.

See Figure 7.11 for illustration of (empirical) cumulative density functions of (negative) log-probabilities. On an average SOP leads to larger negated log-probabilities because its cumulative distribution density is predominantly smaller than those of Ward's clustering and k-means.

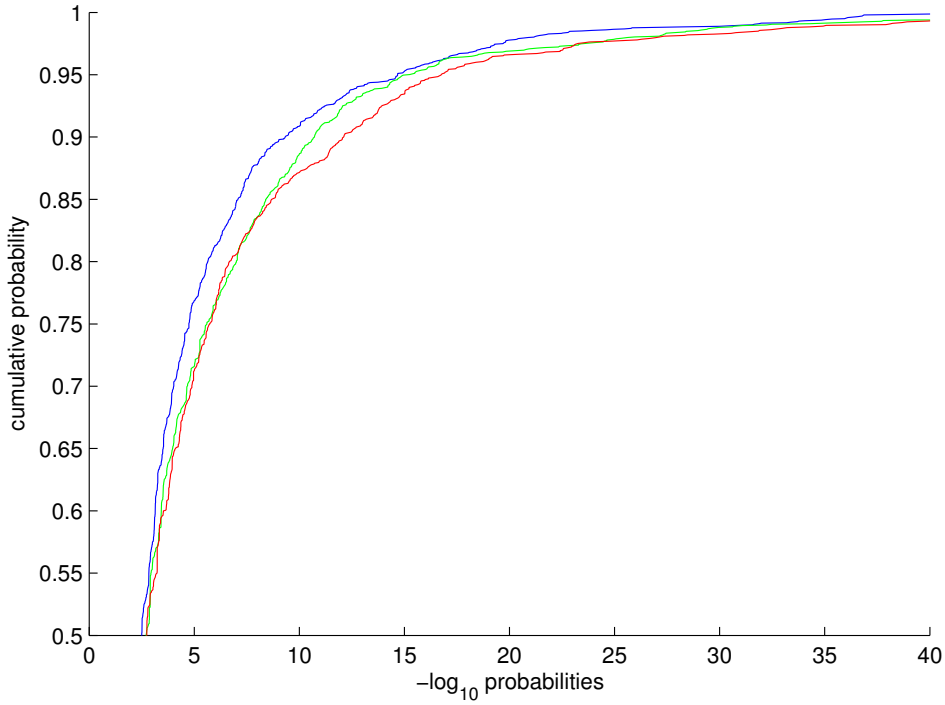


Figure 7.11: Empirical cumulative density function of negative log-probabilities: Ward's clustering (green), k-means (blue) and SOP (red).

7.6 Conclusions

For all miRNA molecules known so far, the most likely regulated genes are retrieved by the help of the TargetScan technique, which provides a numerical indicator score for each gene. The obtained score values are represented as a mixture model, i.e. consisting of a normal and log-normal distribution. The genes originating from the log-normal distribution are supposedly regulated by miRNA. From the Gene Ontology (GO) a meaningful dissimilarity function among these genes was derived, based on the approach of Jiang and Conrath. Swarm-Organized Projection (SOP) clearly depicts eight well-separated classes of genes. For each class the prevailing biological annotations enable novel insights about which types of genes are regulated by miRNA.

Chapter 8

Swarm-Organized Fundamental Analysis

In this chapter, Swarm Organizing Quantization (SOQ) is applied for analysis of high-dimensional stock market data. SOQ projects the fundamental data of american companies onto a low-dimensional output space, where unsupervised classification might be performed. This procedure aims to demonstrate the ability of SOQ to retrieve meaningful classes in non-trivial data on a phenomenological level.

8.1 Fundamental Analysis

Fundamental analysis refers to analytical methods that determine the value of a stock by analyzing the financial data that is *fundamental* to the company. This means that fundamental analysis deals with variables that are directly related to the company itself such as earnings, risk, growth, and competitive position. It focuses on the company's business in order to determine whether the stock should be bought or sold. In contrast to technical analysis [Fam70], fundamental analysis does not consider the overall state of the market nor behavioral features of stocks. For an overview on methods of fundamental analysis see [LT93].

Traditional Approaches

Thomsett [Tho98] gives a phenomenological overview on fundamentals' supposed influence on buy, sell and hold decisions. For example earning ratios, capitalization ratios, profitability ratios and growth ratios are meaningful indicators for the future development of corporations stocks' prices. A formal model for decision or evaluation is not provided.

Lev and Thiagarajan [LT93] have proposed a linear regression method to aggregate stock returns

$$R_i = \beta_0 \cdot \Delta PTE_i + \sum_{j=1}^{n} \beta_j S_{ij}$$

for each company i where ΔPTE_i is the annual change in pretax earnings, S_{ij} are $n = 12$ selected fundamentals and β_i denote the regression coefficients. Accounts

receivable, capital expenditure, gross margin and sales were used for instance. As a benchmark, another regression approach predicts stocks' returns on the basis of past earnings. It could be shown that the fundamental approach outperforms the conventional returns-earnings regression. This suggests that the fundamental signals S_{ij} capture more fully investors' assessment of the persistence of earnings than does the conventional returns-earnings regression.

O'Neil [O'N95] proposed the CANSLIM approach as a strategy for stock selection. CANSLIM is a set of rules for selection of supposedly winning stocks. There are seven characteristics for the selection of stocks:

- **C**urrent (quarterly) earnings increase.
  The quarterly earnings (per share) should increase each period, i.e. when compared to the same quarter of the prior year. Stocks are to be picked that show a major increase. Earnings per share are calculated by dividing companies' total after tax profits by the number of outstanding common shares.

- **A**nnual earnings increase.
  Annual earnings capture a more complete financial situation. Stock strategists access whether the annual earnings of 4-5 past years have been increasing at a healthy rate, i.e. annual earnings increases in the 25-50% range. Companies with better-than-average returns tend to rise their stock prices rapidly. If the earnings' increases are better than the rest of the competitors, then the company is likely to have a strong period of growth.

- **N**ew products, new management.
  Without innovation, a company will eventually vanish and, consequently, its stock prices will decline. In fact, the term "new" may refer to a new product, entrance into a new and untapped market or a recent management replacement.

- **S**upply and demand.
  Companies should have a relatively small number of shares outstanding. The number of shares outstanding should be large enough to trade on a large exchange, but small enough so as not to reduce the stock price and earnings among billions of shares. Although large-cap companies will most likely have a larger increase in sales volume, the percentage of this increase tends to be lesser than smaller companies.

- **L**eader or laggard in the market.
  Companies that manage to continually lead their industry typically have great returns, and are fundamentally sound. Investors can identify leaders by looking at the Relative Price Strength, i.e. the percentage of stocks that have been outperformed by this stock in the market group.

- **I**nstitutional sponsorship.
  Every growing business needs the sponsorship of institutional investors in order to be taken seriously as a sound investment opportunity. For a company

without such investors, the implication is that it is not good enough to be included in a portfolio.

- **M**arket Direction.
  The general market direction determines whether picking stocks according to the above criteria will produce profits or losses.

Neural Approaches

Deboek [Deb98] proposed the use of Self-Organizing Maps (SOM) for unsupervised learning of high-dimensional data concerning mutual funds. This aims at capturing the underlying class structure without a priori knowledge. The data is real valued, i.e. each fund is characterized by a vector of features describing management tenure, fund's annualized returns in 12 months, 3 and 5 years, morningstar agency rating, decile rank in bear markets, mean and standard deviation of returns, size of the fund and supplementary cost measures were selected as features for the creation of the maps. Each feature is standardized so that the variance equals one. The used dataset consists of 122 funds that mainly invest in large and mid-cap stocks. The Self-Organizing Map method [Koh89] maps the data onto a 9×13 sized planar output space. The obtained SOM is visualized by means of the U-Matrix [US90], at which three classes of funds could be manually distinguished. Each class is characterized by its mean values. For a single class, the "annualized return over 5 years" is significantly higher on an average in comparison with other classes.

Deboek and Ultsch [DU00] proposed the usage of Emergent Self-Organized Maps (ESOM) to analyze fundamental data of 2757 stocks. The following features were used for analysis: earnings per share for the first to fourth year, relative strength, percentage of stocks held by funds, outstanding shares, market capitalization, debt to equity. The data is mapped onto a toroidal output space of size 64×64 and visualized by means of the U-Matrix method [US90]. Eight classes could be distinguished manually. Each class is characterized by its features' median values. A single class could be identified that matches the stock-picking criterions proposed by O'Neil [O'N95]. These selected stocks outperform the S&P500 index.

8.2 Data

The *Morningstar* database is a commercial collection of fundamental descriptions of stocks traded on american stock markets. Here, the data published in October of 2004 were used. The data is extracted from the database that contains fundamentals as real valued vectors. Discrete features, such as Morningstar ratings for instance, are discarded. See Table 8.1 for an overview on features. The features are transformed separately in order to give equal weights to each feature. For details on features' distributions see Section D.

Feature	Description	Type	Transform
FndOwn	fund ownership, percental	Size	Box-Cox
ShrOut	shares outstanding in millions	Size	Log
MrktCap	Market capital	Size	Log
DTotYe	Debt to total capital in year 1	Size	Box-Cox
ADV	average daily trading volume	Size	Log
PriCurr	current price	Size	Log
Y1Y2	Revenue percental change Y1-Y2	Return	RelDiff
Y2Y3	Revenue percental change Y2-Y3	Return	RelDiff
Y3Y4	Revenue percental change Y3-Y4	Return	RelDiff
Q1Q5	Revenue percental change Q1-Q5	Return	RelDiff
TRRK1	return versus returns in its industry	Return	-
RS1Y	relative strength	Return	RelDiff
PECurr	current price to earnings per share	Risk	Log
PtBC	current price to book value per share	Risk	Log
PtSC	current price to sales per share	Risk	Log
PtCFC	current price to cash-flow per share	Risk	Log

Table 8.1: Features used for fundamental analysis. Features describe risk, size and return of stocks. Features are non-linearly transformed by means of Box-Cox power transformation, relative difference and logarithm.

Logarithmic Transformation

The log-normal distribution is a continuous probability density distribution defined on positive real values. It describes the distribution of samples $X \subset \mathbb{R}$ at which $\log(X)$ is normally distributed. Log-normal distributions are often skewed, i.e. the distribution's tail on one side of the mean is longer than on the other side at which the bulk of the values is located. A feature might be modeled as log-normal if it can be thought of as the multiplicative product of many independent random variables each of which is positive. In finance for instance, a long-term discount factor can be derived as the product of short-term discount factors. Typically, incomes and stocks' prices are log-normally distributed because they arise from multiplication of normally distributed factors. For details see [AB57] [LSA01].

Thus the following features are log-transformed $x \leftarrow \log(x)$ in order to adjust the distributions such that meaningful expected values and deviations can be obtained: outstanding shares, market capitalization, current price to earnings per share, current price to book value per share, current price to sales per share, current price to cash-flow per share, average daily volume.

Relative Difference

A straight forward way of measuring a stock's return is the arithmetic return $r = \frac{p_{new} - p_{old}}{p_{old}}$ with actual and past prices $p_{new}, p_{old} \in \mathbb{R}_0^+$ on two succeeding points in time. According to Ultsch [Ult08] the arithmetic return follows a leptokurtic distribution, i.e. the normality assumption is appropriate for small absolute returns

only. For larger gains the distribution is not bound. The relative difference [Ult08] alleviates this problems by the use of the following equation:

$$r_{diff} = 2 \cdot \frac{p_{new} - p_{old}}{p_{new} + p_{old}}$$

The relative difference compares gains and losses $p_{new} - p_{old}$ to the average price of both points in time. Relative differences are almost identical to arithmetic returns on the interval $[-0.25, 0.25]$. Furthermore, relative differences are limited to $[-2, 2]$ which facilitates mixture modeling for instance. Each value $x \in \mathbb{R}$ describing arithmetic returns, such as quarterly revenues, is transformed by means of $x \leftarrow \frac{x}{1+\frac{1}{2}x}$.

Thus the following features are transformed into relative differences: percental change of revenue in the first year, percental change of revenue in the second year, percental change of revenue in the third year, percental change of revenue in the fourth year.

Box-Cox Transformation

The Box-Cox transformation [BC64] is a general power transformation

$$x \leftarrow \begin{cases} \frac{x^\lambda - 1}{\lambda} & : \quad \lambda \neq 0 \\ \log x & : \quad \text{else} \end{cases}$$

that depends on the parameter λ. The aim of the Box-Cox transformations is to ensure the normality assumptions holds for the given data. Clearly not all data can be power-transformed to normal. The main objective on Box-Cox transformation is to infer the transformation parameter $\lambda \in \mathbb{R} \setminus \{0\}$. This is often realized by maximization of log-likelihood functions.

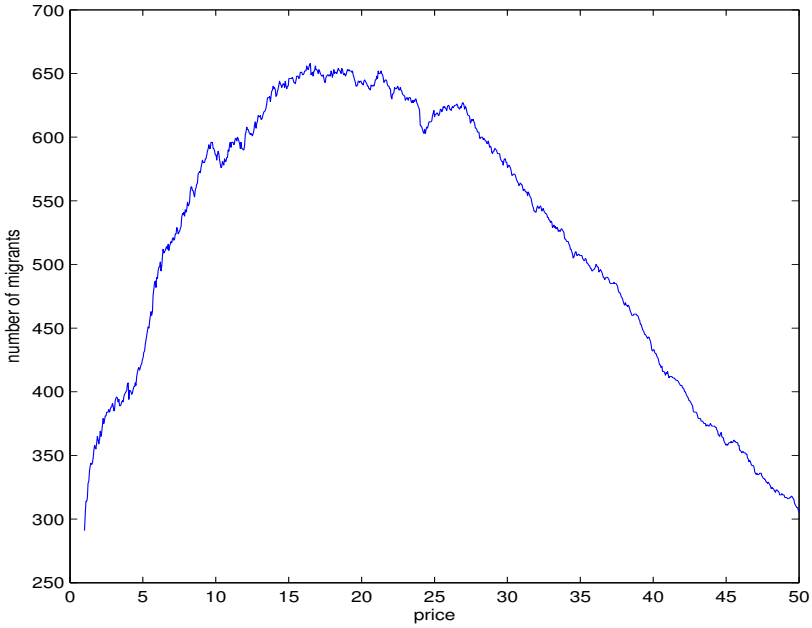
Thus the following features are modified using Box-Cox transformation: percental fund ownership, debt to total capital in the last year.

Robust Standardization

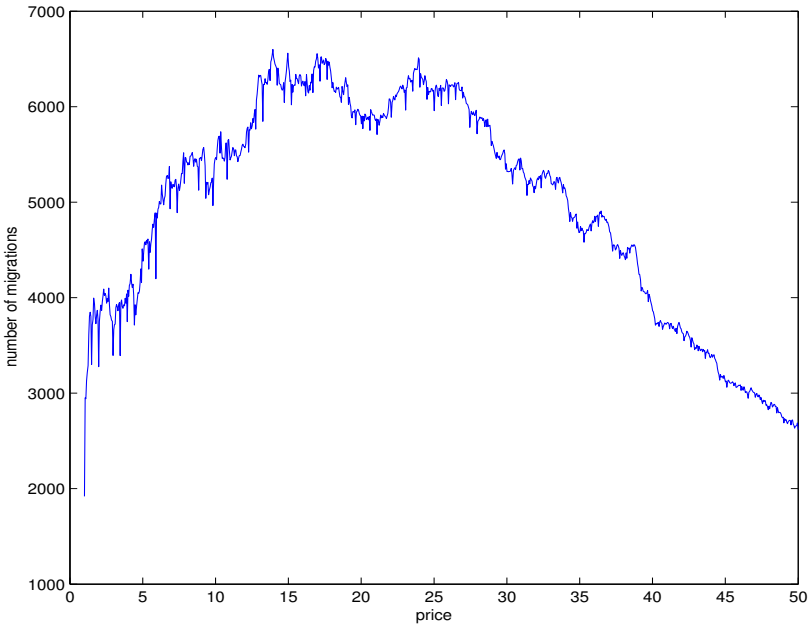
Each feature is linearly scaled such that the deviation is approximately the same on each feature:

$$x \leftarrow \frac{x - x_{min}}{x_{max} - x_{min}}$$

at which x_{max}, x_{min} denote a robust estimate of the feature's maximum and minimum value. According to Milligan and Cooper [MC88] the transformation methods using this standardization scheme offer the best recovery of cluster structures in comparison with other methods like for instance the z-score normalization for equal standard deviations. In this context x_{max}, x_{min} are estimated as distributions' 95 and 5 percentiles.



(a)



(b)

Figure 8.1: (a) Migrants by price. (b) Migrations by price.

Mid-price Stocks

The migration of prices is a particularly interesting subject of investigation for restriction to interesting stocks, i.e. affordable ones. Penny stocks are too risky for reliable portfolio selection. Expensive stocks are not interesting enough. Affordable stocks might be located in the interval $[1, 40]$. A migration$_l$ is a short fluctuation of a price, i.e. the stock's price crosses a given limit $l \in \mathbb{R}^+$ in the upward/downward direction on two succeeding trading days. For each possible financial barrier in $[1, 40]$ the number of migrants is identified. The number of migrations is to be maximized by appropriate limits for stocks' prices. From Figure 8.1 it can be seen that migrations often do occur in the $[9, 25]$ dollar segment. Stocks with prices outside that segment are less likely to migrate into another price segment. This means that these mid price stocks (mid caps) are selected for further portfolio analysis, because small caps and big caps are more likely to show no interesting development. Reportedly, mid cap stocks are more risky than big cap stocks and less risky than small cap stocks. Generally, risk of company failure decreases as the company increases in size. However, a mid cap stock also has better potential for growth than a big cap company. A very large company may have completely saturated its market, while a mid cap company may have room to grow into a big cap company.

8.3 Stock Classification

In order to derive a "ground truth" by which to assess the ordering abilities of SOQ, the performance of companies' stocks is used to obtain a classification. This classification is to be compared against the results of SOQ

Filtering of Financial Time Series

Let $(p_1, ..., p_t)$ with $p_i \in \mathbb{R}^+$ for $i = 1, .., t$ denote the time series of a stock's price during a certain period. Such time series are filtered in order to discard singularities, e.g. peeks of few points in time. Usually, this leads to a *smoothing* of the time series.

Formula 8.1

$$\mathit{filter}_w(p_i) = \frac{p_i + \mathit{mean}(p_i, ..., p_{i-w}) + \mathit{median}(p_i, ..., p_{i-w})}{3}$$

In this context, the filter given in formula 8.1 is used. The window size $w \in \mathbb{N}$ determines the smoothing of time series. The median filter is a non-linear filtering technique, often used to remove noise from images or other signals. This is performed using a window consisting of time series' samples. The values in the window are sorted into numerical order. The median value is selected as the output. In statistics, a moving average is used to analyze time series data. It is applied in finance and especially in technical analysis. See [EM69] for details. It can also be used as a generic smoothing operation, in which case the raw data need not be a time series. It is a linear filter. The identity leaves the time series untouched. The resulting operation $filter$ is a combination of a non-linear low-pass filter, a linear

low-pass filter and the original time series. This operation leads to a smoothing of the time series that is sensitive to sudden peaks, yet.

The filter operation $filter_w$ greatly depends on the applied window size. In order to select an appropriate window size $w \in \mathbb{N}$ for filtering, the following approach was used. The S&P 500 index serves as a representative for typical market situations. Let $(p_1, ..., p_t)$ denote the time series of prices. Let $(r_1, ..., r_{t-1})$ with $r_i = \frac{p_i+1-p_i}{p_i}$ for $i = 1, ..., t-1$ denote the time series of daily returns of S&P 500 index. As suggested by Alfred Ultsch, the distribution of returns is modeled as a mixture of log-normal, normal and log-normal distribution. See Figure 8.2(a) for illustration. The Bayes calculus leads to a decision boundary for *underperforming* below -1.023 percent and for *overperforming* above -1.023 percent.

A smoothing of time series reduces the amount of under- and overperforming returns. This is a loss of information. Therefore, the kept information is to be measured as the percentage of returns that are still determined as under- and overexprimated by the original Bayes decision boundaries. In order to find an appropriate window size $w \in \mathbb{N}$ for smooth but informative time series, a scree plot is used. See Figure 8.2(b) for illustration. It can be seen that a window size of $w \approx 7$ leads to a saturation of information loss.

Classifying Stocks

Stocks are to be classified according to their returns during the following quarterly period into several classes, e.g. winning stocks, losing stocks and equal stocks. A classification into several classes may be obtained by defining strict thresholds for returns. This means: a single return r_i above 10 % classifies a stock as *winning*, whereas a single return below -7 % classifies the stock as *loosing*.

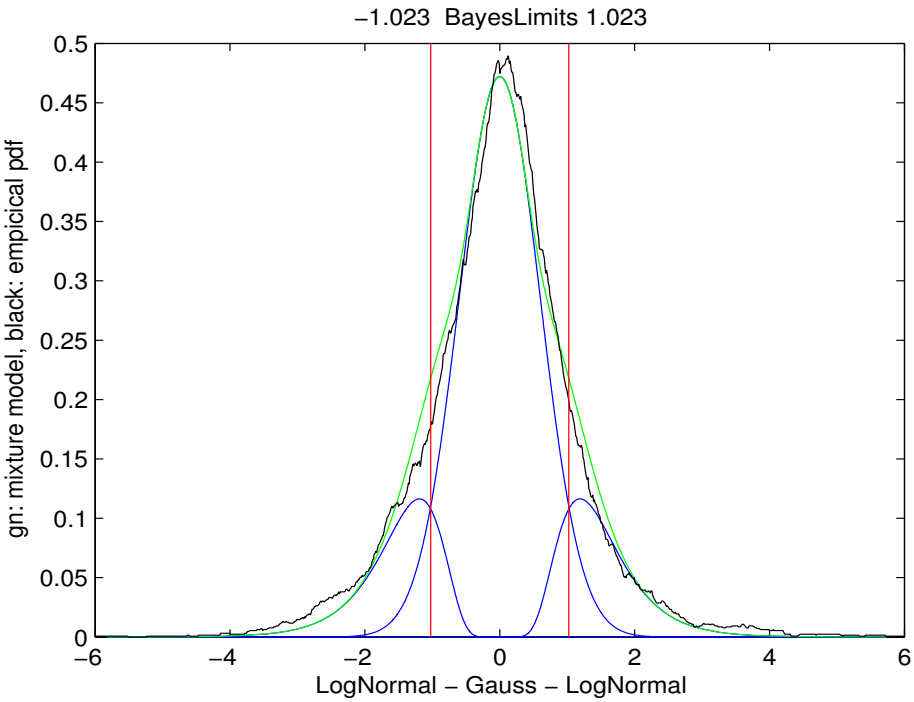
This static approach, usually, does not account for trends of the market. For example, a so-called bear market leads to a massive downturn of stocks' prices and returns, respectively. Nearly all stocks will be classified as *losers* of necessity. Instead, a more dynamic approach is applied for classification. Thus, the definition of winning and loosing becomes more flexible in terms of bear versus bull markets. A realistic measurement for the market development is the S&P 500 index. For further use, stocks' returns are related to the mean returns achieved by the market itself by means of:

$$r_i \leftarrow r_i - r_{S\&P,i}$$

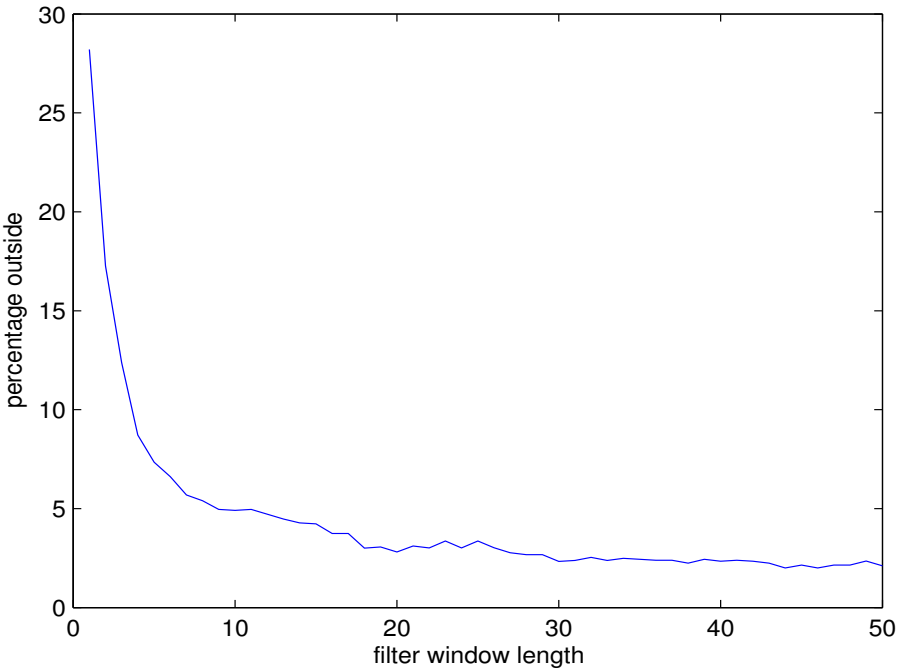
8.4 Results and Evaluation

Findings

In this showcase example, the Swarm-Organized Projection (SOQ) projects the fundamental data $x_1, ..., x_n$ onto a two-dimensional planar grid of size 50×82. No additional parametrization is needed. The SOQ algorithm decreases the radius stepwise from $\sigma_{max} = \sqrt{\left(\frac{50}{2}\right)^2 + \left(\frac{82}{2}\right)^2} \approx 48$ down to $\sigma_{min} = 1$. For each node in the output space, a codebook vector is provided. In contrast to Self-Organizing



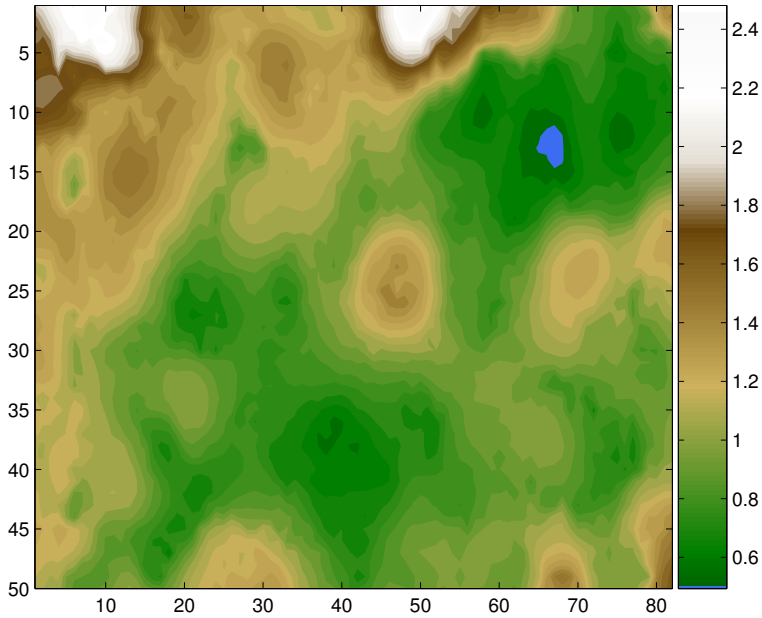
(a)



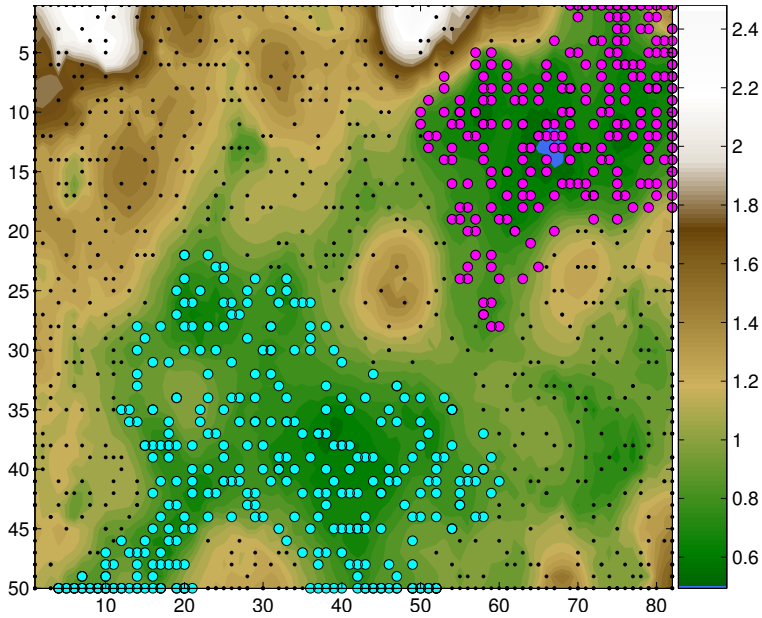
(b)

Figure 8.2: (a) Daily returns of S&P 500 index modeled as Log/Normal/Log-Distribution. (b) Scree-plot: amount of information kept against window size

Batch Maps, the update of codebook vectors happens iff the codebook vector in question contributes to the learning of the topographic mapping. The Generalized U-Matrix method (cf. Section 4.4) depicts the expected distance in data space on each node of the output space. The resulting U-Matrix shows at least two main classes at the center of the output space (see Figure 8.3(a) for illustration).



(a)



(b)

Figure 8.3: (a) U-Matrix of fundamental data depicts two large valleys in the middle. (b) Manually classified U-Matrix. Block dots denote unclassified data vectors.

Evaluation

In order to assess the semantics of the obtained U-Matrix, a "third-party" ground truth about companies is derived by means of the stock market. In Section 8.3 each company is classified as *winner*, *looser* or *equal* on the basis of its stock price development. From the visual impression it is concluded that the "equal" class is overrepresented in the found clusters, in comparison with the (unclassified) outliers. See Figure 8.4 for illustration. This is quantified by means of the Bayesian calculus [Bay63]. Instead of discrete locations, i.e. multiset $\{y_1, ..., y_n\}$, a gaussian probability function $F_{i,\sigma} : \mathbb{O} \to [0, 1]$ is centered on each y_i in output space. The conditional probability $p(C|j)$ for class $C \subset \mathbb{X}$ on a given location $o_j \in \mathbb{O}$ follows as:

Formula 8.2

$$p(C|j) = \frac{p(j|C) \cdot p(C)}{p(j)} = \frac{\sum_{x_k \in C} F_{\sigma,j}(y_k)}{\sum_{x_l \in \mathbb{X}} F_{\sigma,j}(y_l)}$$

The conditional probability for the "equal" class $C \subset \mathbb{X}$ is significantly higher in the valleys (i.e. clusters) of the U-Matrix. See Figure 8.5 for illustration. This hypothesis is quantified by the help of statistical testing and a simple classification rule that mimics the unsupervised classification with elaborated clustering algorithms that operate on top of the U-Matrix. Nodes o_i with small U-heights are considered as "valleys" which indicate clusters in the fundamental data. Let $u_0 \in \mathbb{R}^+$ be a threshold for segmentation of the U-Matrix. Thus, two sets $\{o_i : u(o_i) \leq u_0\} \subset \mathbb{O}$ and its complement are obtained. For each set of nodes the corresponding conditional probabilities follow as $P = \{p(C|i) : u(o_i) \leq u_0\}$ and $P' = \{p(C|i) : u(o_i) > u_0\}$.

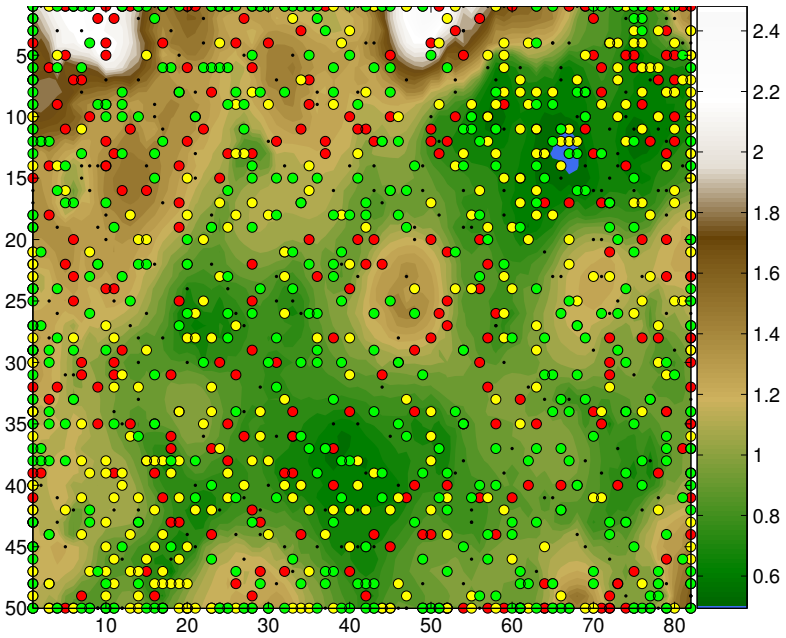


Figure 8.4: Market classification: winners (green), equals (yellow) and losers (red).

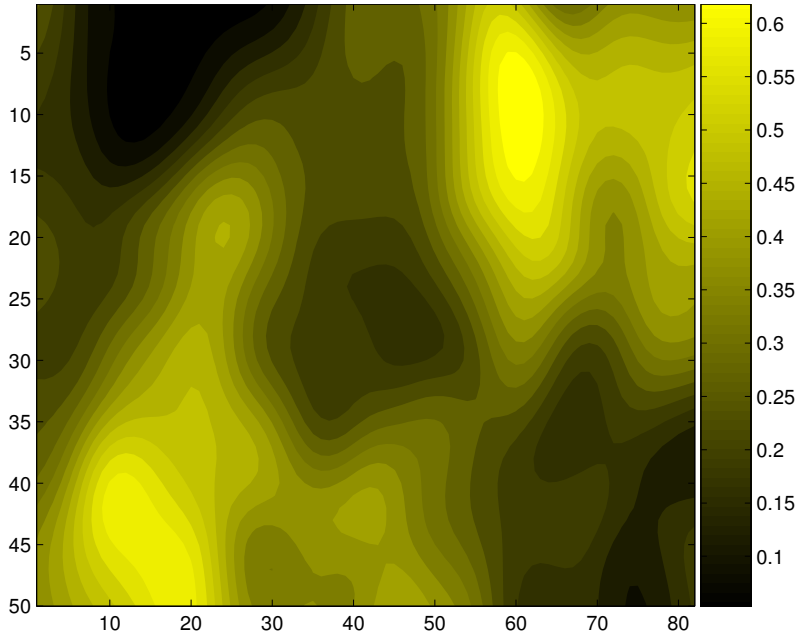


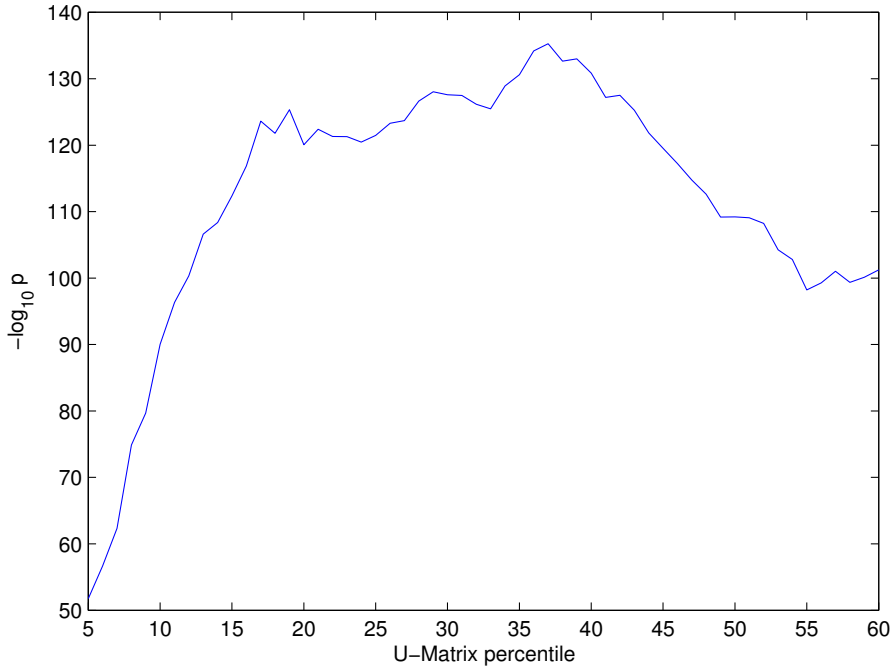
Figure 8.5: Conditional probility for "equals" on the output space.

A statistical test is used to quantify the error probability $p \in [0, 1]$ for rejecting the null hypothesis that P, P' come from the same distribution. In this case, rejecting the null hypothesis would mean that the Swarm-Organized Quantization selected significantly differing sets of stocks (according to the corresponding fundamentals) with differing conditional probabilities P, P' for being an equal-performer to the market index S&P 500. For $\nu = 5, ..., 60$, each ν-percentile of the heights of the U-Matrix serves as as segmentation threshold u_0. This leads to a sequence of error probabilities p_ν. The conditional probabilities P, P' are not normally distributed, i.e. coming from multi modal distributions. Thus, the two sample Kolmogorov-Smirnov test [Smi48] is applied instead of Student's t-test [Stu08]. The negative log-probability $-\log_{10} p_\nu$ indicates the exponent of the error probabilities, which is more convenient to read. See Figure 8.6(a) for illustration of p-values's exponent against percentiles ν. This indicates that for a wide range of $\nu \in \{15, ..., 45\}$ the SOQ method leads to a meaningful segmentation of stocks, at which lower U-heights indicate a higher probability for equal-performing stocks. The probability of error for this finding is below the $\alpha = 10^{-100}$ level.

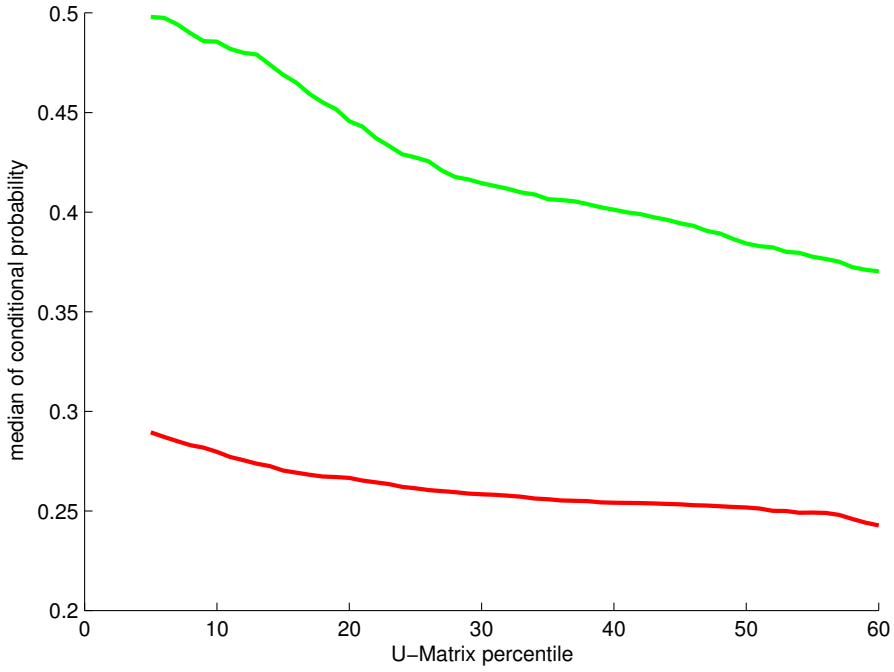
8.5 Conclusions

Fundamental analysis was performed by the help of Swarm-Organized Quantization (SOQ), which produces a low-dimensional projection from companies' fundamental features. The inherent distances were visualized by means of the U-Matrix method. Visual inspection reveals two large clusters. The resulting U-Matrix serves as an indicator for development of stocks' prices. The conditional probability for market-

conforming development of stocks' prices is significantly higher in output space regions with low U-Matrix heights.



(a)



(b)

Figure 8.6: Percentiles for U-Matrix segmentation: (a) Negative log-probability from two-sample KS test. (b) Median conditional probability for selected (green), unselected (red) regions.

Chapter 9

Discussion

The main concern of this work was to investigate how the Swarm-Intelligence paradigm can contribute to topographic mapping, i.e. the creation of low-dimensional images that depict regularities of high-dimensional or complex data. In literature, little can be found on successful applications of swarm-driven algorithms on the field of visual data exploration. The popular Ant-Based Clustering [LF94] method was often adapted but found to be not competitive with state-of-the-art methods [HKD05].

In this thesis, the Swarm-Organized Projection (SOP) was introduced. It is based on stochastic agents and fixed point iteration in order to overcome the parametrization problems of traditional topographic mapping methods. The power of SOP was demonstrated my means of empirical and theoretical evaluation and application on real-world data.

9.1 Novelty

On the Relation of ABC and SOM

In Section 5.1 the Ant-Based Clustering algorithm [DGF+90] [LF94] was characterized as a derivative of Self-Organizing Maps, whose inability to create correct topographic mappings derives from fixed and too small neighbourhoods, and the accounting for meaningless quantities such as the density in output space. However, these formal insights are new but not surprising since empirical evidence can be found in literature. Handl et al. [HKD05] have provided empirical evidence that topographic mapping is greatly improved by altering the normalization term of ABC, as proposed in Section 5.1. Tan et al. [TTT06] claim that the number of simulated ants does not affect the obtained mapping of data objects. However, a formal argument was not provided, but is found in Theorem 5.1 of this thesis.

Swarm-Organized Methods

The SOP method was inspired by Databots [Ult00a], at which data objects are identified with agents that move on a finite but unbound grid universe. Each SOP agent tends to minimize its personal topographic stress Φ. However, stress functions similar to Φ can be found in well-established methods. As shown in [HU08],

Φ is an upper limit of the quantization error used in Batch-SOM Ψ and, moreover, forms the topographic term of the attractiveness function used in traditional Ant-Based Clustering [LF94]. Furthermore, Φ resembles the generalized median of Dissimilarity-SOM [KS02] which is a derivative of conventional Batch-SOM that operates on pairwise dissimilarities instead of vectorial data. The Dissimilarity-SOM suffers from the *model collision* problem [Ros07], i.e. several neurons share a common codebook, which leads to misrepresentations of cluster structures. The SOP architecture avoids codebooks. Model collisions therefore do not occur when using SOP.

Although the traditional neighbourhood function F_σ of Batch-SOM, CCA and SOP is a Gaussian, it is not related with t-SNE neighbourhoods p_{ij}, q_{ij}. The F_σ represents a weight term for the interaction between mapped objects y_i, y_j. This interaction might be reconciliatory $(d_{\mathbb{D}}(x_i, x_i) - d_{\mathbb{O}}(y_i, y_j))^2$ as in CCA, or strictly repellent $d_{\mathbb{D}}(x_i, x_i)$ as in SOP and Batch-SOM. This is the main reason why the map space of SOP may be unbounded (i.e. toroid) but has to be limited. In contrast to that, t-SNE uses the terms p_{ij}, q_{ij} for reconciliatory purposes.

Annealing

If the structural features of a data set and the scaling of the structural features is not known, otherwise successful algorithms like CCA and SOM may eventually construct a faulty topographic mapping. The main reason for this effect stems from the difficulty to select an appropriate annealing scheme for the parameter which determines "neighbourhood" in the data. If the annealing scheme fits the structural scaling of the data, a sufficient topographic map is constructed when using sound and complete learning algorithms. Batch-SOM and its derivatives rely on exhaustive search on all grid nodes for bestmatching units. In contrast to that, SOP relies on few random samples from output space (see Algorithm 4.1). The computational effort of exhaustive bestmatch search is therefore avoided. Non-exhaustive search techniques enable the integration of batch learning criterions with stochastic search. The computational effort of the annealing scheme adapts itself to the intricacies of the data.

However, the use of fixed point iterations has been proposed in literature afore. The soft topographic vector quantization algorithm (STVQ) as proposed by Graepel and Obermayer [GBO98] offers the creation of SOM-like mappings. The STVQ relies on minimization of an objective function E by the help of the EM algorithm [DLR77]. This is achieved in two nested loops, similar to those used within SOP. The inner loop is a fixed point iteration in terms of the objective function, depending on the "free energy" parameter $\beta \in \mathbb{R}^+$. This parameter determines the smoothing that is done to the original objective function. Starting with low values, the second loop increases the value of β until the desired solution is obtained. Obviously, the β similarly acts as neighbourhood parameter, as known from radius σ in SOM-like systems. It is, however, not known how a reasonable choice for β and its increase factor is determined. In contrast to that, the neighbourhood radius σ of SOP is determined according to the architecture of the grid-shaped output space.

The SOP offers a much simpler architecture without the need for global objective functions, gradient descents, soft (fuzzy) assignments and EM like optimizations.

Acceleration

The computational complexity of SOM learning is dominated by the bestmatch search (cf. Formula 3.4). This is evident because all codebook vectors have to be checked, but the update affects few codebook vectors only, especially when using finite focus functions F_σ. In order to accelerate the learning of traditional Online-SOM and Batch-SOM (cf. Section 3.2) few techniques have been proposed in literature to constrain the set of searchable codebook vectors.

Most techniques exclude codebook vectors according to proximities in data space. Ra and Kim [RK91] proposed a weight-distance ordered partial codebook search (WPS) algorithm, which uses the mean value of data vectors to reject codebook vectors that cannot be bestmatches. Cuadros-Vargas et al. [CVRO03] and Kaski [Kas99] propose the use of tree-structures to split up the set of codebook vectors, e.g. by hyperplanes. These approaches are rather complex and, furthermore, can hardly be applied when working on dissimilarity data. The shortcut search from Kohonen [Koh99] [LKK04] restricts the bestmatch search for a given $x_i \in \mathbb{X}$ to the vicinity around the old bestmatch location $y_{i,old} \in \mathbb{O}$. However, there is no formal limit for the deviation of old and new bestmatch locations.

The Swarm-Organized Quantization (SOQ) may be regarded as a derivative of the Batch-SOM, that does not require an exhaustive search for the bestmatching neurons. The Swarm-Organized Projection (SOP) may be regarded as a derivative of the Dissimilarity-SOM, that does not rely on codebook vectors and does not require an exhaustive search for the bestmatching neurons. Both methods rely on a mechanism for approximate refinements of bestmatch locations, that is guided by stochastic agents instead of rigid criterions. To the best of our knowledge, this is a novel trait that distinguishes SOP and SOQ from other approaches. Due to its simplicity, the swarm-driven search and update of codebook vector might be combined with other acceleration techniques.

Applications

In Chapter 7 the SOP method depicts eight classes of miRNA-regulated genes. An expert confirmed the composition of classes to be so far unknown in literature, surprising and, therefore, valuable as a hypothesis for further research. Retrieving functional groups of genes is not a new analytical approach. However, functional grouping usually lacks some profound validation, cf. [SSZ05], e.g. by means of expert knowledge. Instead, we propose evaluation by means of an external cost function, i.e. the p-values obtained by over representation analysis methods such as GeneTrail [BKK$^+$07].

The influence of fundamental data on the development of stocks' prices is known from literature [LT93] [Deb98]. Thus, the connection between both classifications, development on stock market and classes derived from the U-Matrix, is not surprising. However, well-separated classes as retrieved by Deboek [Deb98] could not

be reproduced. This might be explained by differing feature selection and feature transformation.

9.2 Improvement

Objective Functions

Many methods, such as CCA or SNE, are characterized by an objective function that is optimized by the corresponding learning algorithm. Constant decrease of an objective function ensures convergence of the algorithm. This is not the case in SOP, at which the existence of an objective function could no be proven, yet. Even though the sum of Φ has been monitored to be monotonically decreasing. It remains unclear whether the existence of objective functions leads to superior mappings or proofs concerning these mappings. It was shown that the inner loop of the SOP algorithm will halt, since the probability for all agents to keep on moving is a monotonically decreasing function of time.

Since a SOP agent does not account for the stress Φ of other agents, it is likely that a decrease of its own stress will increase the stress of others, which is supposedly an unwanted effect but prevents the algorithm from premature convergence into local optima of an unknown objective function.

Benchmarks

The results obtained from the dispersion measure are non-ambiguous. The CCA method leads to less cohesive mappings than other methods, especially on the high-dimensional real world data sets SwissBanknotes, Wine and GPD194. This can be attributed to the mismatching default parametrization. The t-SNE is a promising technique that often leads to non-cohesive mappings due to the intentionally deficient (default) parametrization. This can be seen on many real world problems and the artificial Chainlink data for instance.

The SOP method produces mappings of little dispersion compared with the best obtained mappings. It even outperforms all other methods on Atom and SwissBanknotes data. However the sparse stochastic search of SOP agents prevents better results, because few agents are likely not to move on locations with less topographic stress Φ.

The accuracies of the nearest neighbour classifier lead to different conclusions. The non-linear embedding algorithms CCA and t-SNE clearly outperform the topology-preserving Batch-SOM and SOP on the real world data sets Iris, SwissBanknotes and Wine. On the other data sets, the differences are not that obvious. The separation of border classes greatly decreases in case of uniform distributions in output space, as obtained by SOP and Batch-SOM. See Figure 4.2 for illustration. Thus, CCA and t-SNE lead to more separation and less cohesion of classes in output space. However, Batch-SOM and SOP methods are intended as base for subsequent visualization techniques (cf. Section 3.7).

Applications

Chapter 8 shows that the probability for market-conforming stocks is roughly predictable by means of the resulting U-Matrix. Unfortunately, it is not a separate class on the U-Matrix that predicts the stocks' behaviour of interest. Instead, the unclassified outliers are less likely to show market-conforming behaviour. It is therefore hard to compare our results with traditional cluster algorithms, which leave no data vector unclassified.

Chapter 7 presents a novel approach for analysis of microRNA (miRNA) regulation patterns, i.e. which types of genes are actually regulated by miRNA. An expert confirmed the composition of gene classes to be so far unknown in literature, surprising and, therefore, valuable as a hypothesis for further research. The SOP method did outperform k-means and Ward clustering in terms of p-values obtained by Over representation Analysis (ORA). Even though the findings are statistically significant below the $\alpha = 0.001$ level, the expected improvements are hardly understood in terms of gained knowledge. However, preliminary analysis of additional re-runs showed even more promising results when evaluating with ORA.

9.3 Validity

The Batch-SOM method has been shown in Section 6.5 to produce topographic mappings of varying quality, depending on the pre-defined parametrization. These results cannot be generalized to other types of Self-Organizing Maps, e.g. Online-SOM or Heskes-SOM, because the update rules concerning the mapping greatly differ in these algorithms. Furthermore, our empirical evidence is limited to 2-dimensional borderless map space topologies.

Datasets

The data sets were carefully chosen in order to demonstrate several effects that frequently occur at the construction of topographic mappings. The "Chainlink" dataset is an artificial benchmark example consisting of more than a single manifold, at which global and local proximities contradict each other with respect to cluster memberships. For each ring there are points closer to the center of the other chain than its own. The Batch-SOM performs best on Chainlink data. However, Batch-SOM learning is prone to premature convergence. See [NMU06] for details.

The Atom data set consists of two classes having the same center point. This is a hard problem for methods that rely on computation of average vectors in data space. Thus the poor results of Batch-SOM become explainable.

The EngyTime data set consists of overlapping classes. Benchmarking with EngyTime evaluates the usability of projections in density-defined domains. For the Iris data, SOP performs slightly worse than CCA and Batch-SOM. This can be attributed to the stochastic nature of agents which does not guarantee convergence to a local optimum. In contrast to that, SOP does not show severe inabilities when trying to capture non-linear manifolds of Atom and Chainlink data.

Evaluation

The evaluation is based on the dispersion measure, which quantifies topographic mappings in a supervised fashion by means of Delaunay graphs in output space. This is based on formal analysis of the U-Matrix visualization technique. In comparison with other topographic quality measures, dispersion is a coarse indicator that is, however, invariant with respect to the topology of the output space. This is mandatory for comparing topographic mappings with greatly differing output space, e.g. periodic grids and non-periodic euclidean spaces.

The assumption for the dispersion measure to prefer the SOP method and penalize other methods does not hold in practice. The dispersion measure is the (normalized) sum of path lengths between non-connected subgraphs of a class (cf. Section 6.2). It does not account for the cardinality of these subgraphs. Thus, a single disconnected data object and a disrupted cluster are considered the same. In fact, SOP typically projects few (or single) data objects far away from its designated cluster, due to the stochastic nature of the algorithm. This means that the obtained topographic mappings are far more convenient than indicated in Section 6.5.

The periodic boundary condition supported in Batch-SOM and SOP do not imply an advantage over the non-periodic output spaces of CCA and t-SNE. CCA correctly unfolds, for instance, the periodic Atom data onto non-periodic output space $\mathbb{R}^2$ In contrast to that the Batch-SOM fails to reproduce the core cluster cohesively, despite its periodic output space.

9.4 Future Works

Experimental Evaluation

The benchmarking of SOP and SOQ, too, has to be continued in order to demonstrate strengths and weaknesses of the proposed self-organization paradigm. An investigation in sparse domains is particularly interesting, i.e. when mapping few data objects onto many grid nodes such that nearest neighbours are less likely to be found by stochastic agents.

Mining microRNA Data

The influence of (dis)similarity functions on the interestingness, novelty and validity of the obtained gene classes has not been investigated, yet. For example, Resnik's similarity based on Gene Ontology Terms may be applied directly to the self-organization process of the SOP algorithm and for visualization purposes by means of the Generalized U-Matrix.

Parameter Studies

The amount of samples each agent uses for sampling may have vital influence on its ability to localize a region in output space with less topographic stress Φ. Currently, the set of sensory samples has two elements. A larger set evidently leads to a higher probability of finding a node with less topographic stress, in comparison with the current location. However, large amounts of sensory samples may lead to premature

convergence into, i.e. the algorithm is prone to get stuck in a local minimum of the objective function. The main aim is therefore to derive an appropriate number of sensory samples as a function of output space size and number of data objects.

Chapter 10

Summary

Topographic mappings try to project high-dimensional or complex data onto a low-dimensional output space while sufficiently preserving the topography of the data. In this thesis we presented a novel algorithm for creation of topographic mappings from vectorial and relation data. The Swarm-Organized Projection (SOP) was inspired by Swarm-Intelligence methods. This means that the learning algorithm mimics the behaviour of social animals, e.g. flocking of birds or stigmergic cooperation of ants.

An introduction to topographic mapping techniques outlines the main characteristics of relevant learning algorithms. The quality of obtained mappings for such algorithms depends critically on a suitable concept of neighbourhood. Focusing methods rely on a (decreasing) neighbourhood parameter to capture global proximities first and more local proximities later on. When the annealing of these parameters is too fast or too slow, this frequently leads to misrepresentations of the inherent structures of the data. Popular focusing methods are Self-Organizing Batch Maps and Curvilinear Component Analysis. Non-focusing methods often rely on a fixed, but pre-defined concept of neighbourhood that is not changed during the learning of the topographic structure. Mismatching parametrization leads to severe misrepresentations of cluster structures. Particularly interesting non-focusing methods are Stochastic Neighbour Embedding and Ant-Based Clustering. Formal analysis of Ant-Based Clustering (ABC) reveals a connection with Self-Organizing Maps on the basis of shared signature terms. The structural inability to generate convenient projections becomes explainable, since ABC accounts for meaningless quantities that are prone to distort the formation of correct topographic mappings.

Swarm-Organized Projection (SOP) was proposed as a solution to the parametrization problem known from focusing algorithms, i.e. deriving an appropriate annealing scheme for the neighbourhood radius. SOP self-adapts its annealing scheme according to the behaviour of its swarm entities. On few selected benchmark data sets, the SOP method was compared against carefully parametrized competitor methods, namely Self-Organizing Batch Map, Curvilinear Component Analysis and t-distributed Stochastic Neighbour Embedding. SOP consistently projects the data topographically correct. In contrast to that, the competitor methods show crucial inabilities on few data sets. SOP shows only

marginal deviation from the best obtained results obtained, or even outperforms other methods.

The practical usefulness of our proposed method was demonstrated by means of two real-world applications. The Swarm-Organized Quantization (SOQ) is a fast derivative of SOP for vectorial data. SOQ projects fundamental data of american companies onto a low-dimensional output space for visual data analysis. The retrieved cluster structures are greatly related with the behaviour of stocks' prices, a quality that cannot be derived from fundamental data in a trivial way. Another real-world application applies SOP on data from molecular biology. The set of genes that are supposedly regulated by microRNA is classified into homogeneous subsets, in order to give a convenient answer to the question which types of genes are actually regulated by microRNA. The obtained classes were validated by means of Over representation Analysis, and show significantly better results than Ward's clustering and k-means. An expert confirmed the novelty and usefulness of the obtained knowledge.

Appendix A

Algorithms

Algorithm A.1 Self-Organizing Batch Map

function BATCHSOM($\mathbb{X}, \sigma_{min}, \sigma_{max}, t_{max}$)
  randomize $\{w_i : o_i \in \mathbb{O}\}$
  for $t \leftarrow 1, ..., t_{max}$ **do**
    $\sigma \leftarrow radiusannealing(\sigma_{min}, \sigma_{max}, \frac{t}{t_{max}})$
    for $j \leftarrow 1, ..., n$ **do**
      $y_j \leftarrow \arg\min_{o_i \in \mathbb{O}} \Psi(x_j, o_i)$
    end for
    for $o_i \in \mathbb{O}$ **do**
      $w_i \leftarrow \frac{\sum_{x_j \in \mathbb{X}} F_\sigma(d_\mathbb{O}(y_j, o_i)) \cdot x_j}{\sum_{x_j \in \mathbb{X}} F_\sigma(d_\mathbb{O}(y_j, o_i))}$
    end for
  end for
end function

Algorithm A.2 Curvilinear Component Analysis

function CCA($\mathbb{X}, \sigma_{min}, \sigma_{max}, \alpha_{min}, \alpha_{max}, t_{max}$)
  randomize $\{y_1, ..., y_n\}$
  for $t \leftarrow 1, ..., t_{max}$ **do**
    $\sigma \leftarrow radiusannealing(\sigma_{min}, \sigma_{max}, \frac{t}{t_{max}})$
    $\alpha \leftarrow rateannealing(\alpha_{min}, \alpha_{max}, \frac{t}{t_{max}})$
    for $i \leftarrow \pi(1), ..., \pi(n)$ **do**
      for $j \leftarrow 1, ..., i-1, i+1, ..., n$ **do**
        $y_j \leftarrow y_j + \alpha \cdot F_\sigma(d_\mathbb{O}(y_i, y_j)) \cdot (d_\mathbb{D}(x_i, x_j) - d_\mathbb{O}(y_i, y_j)) \cdot \frac{y_j - y_i}{d_\mathbb{O}(y_j, y_i)}$
      end for
    end for
  end for
end function

Algorithm A.3 Ant-Based Clustering

function ANTCLUSTER($\mathbb{X}, k_p, k_d$)
  randomize $\{y_1, ..., y_n\}$
  randomize $\{a_1, ..., a_{n'}\}$
  $c_k \leftarrow \perp$ for $k = 1, ..., n'$
  while true **do**
    $k \leftarrow$ choose ant
    $a_k \leftarrow randomwalk(a_k)$
    $\pi_p \leftarrow p_{pick}(x_{c_k}, a_k)$
    $\pi_d \leftarrow p_{drop}(x_{c_k}, a_k)$
    $\pi_0 \leftarrow randomnumber \in [0, 1]$
    if $c_k \neq \perp \wedge \pi_0 \leq \pi_d$ **then**
      $y_{c_k} \leftarrow a_k$
      $c_k \leftarrow \perp$
    end if
    if $c_k = \perp \wedge \pi_0 \leq \pi_p \wedge \exists y_i = a_k$ **then**
      $y_i \leftarrow \perp$
      $c_k \leftarrow i$
    end if
  end while
end function

Algorithm A.4 Stochastic Neighbour Embedding

function T-SNE($\mathbb{X}, \alpha$)
  randomize $\{y_1, ..., y_n\}$
  while $\neg$ converged **do**
    for $i \leftarrow 1, ..., n$ **do**
      for $j \leftarrow 1, ..., n$ **do**
        $q_{ij} \leftarrow \frac{e^{-d^2_{\mathbb{O}}(y_i, y_j)}}{\sum_{k \neq i} e^{-d^2_{\mathbb{O}}(y_i, y_k)}}$
      end for
    end for
    for $i \leftarrow 1, ..., n$ **do**
      $y_i \leftarrow y_i + \alpha \cdot \sum_{j=1}^{n} (p_{ij} - q_{ij})(y_i - y_j)$
    end for
  end while
end function

Appendix B

Benchmarks

B.1 Data

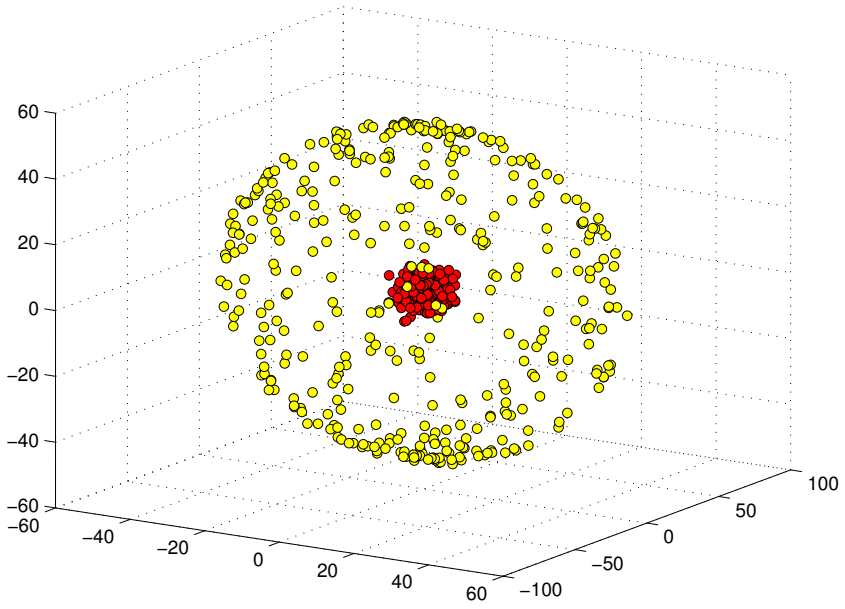


Figure B.1: Atom data in $\mathbb{R}^3$.

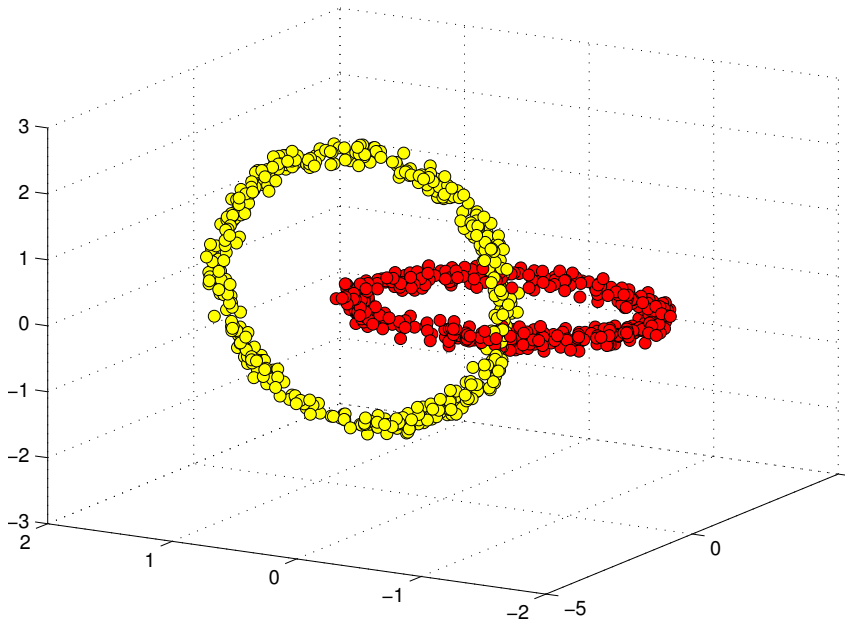


Figure B.2: Chainlink data in $\mathbb{R}^3$.

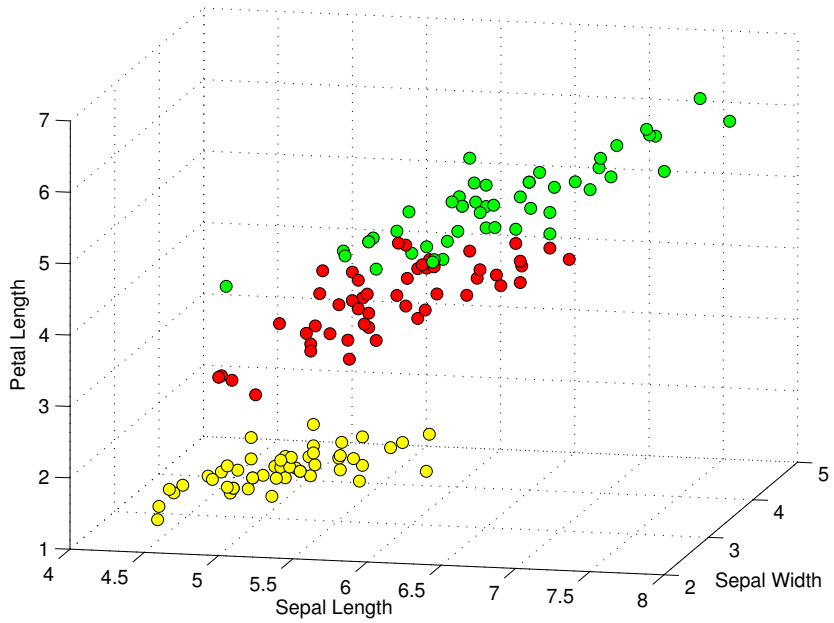


Figure B.3: Iris data restricted to $\mathbb{R}^3$.

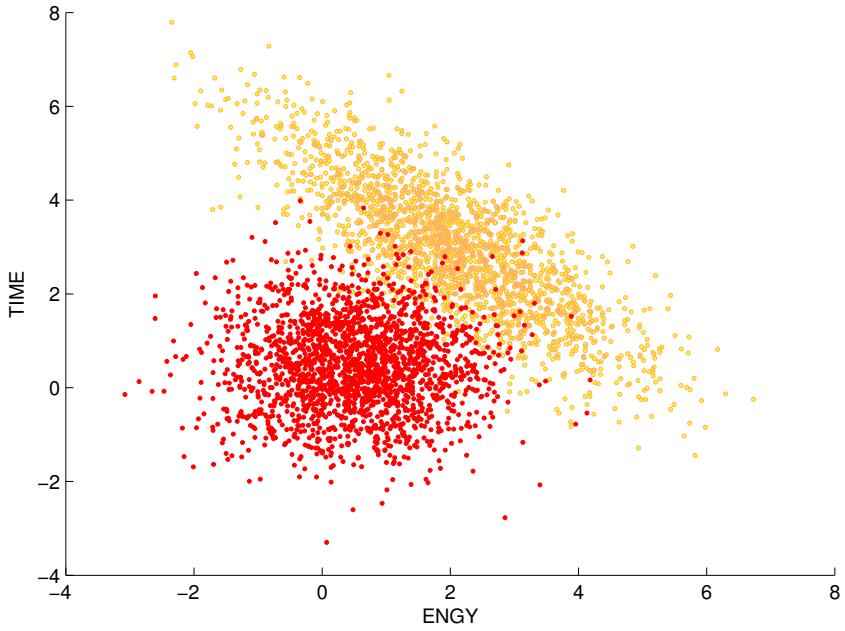


Figure B.4: EngyTime data in $\mathbb{R}^2$.

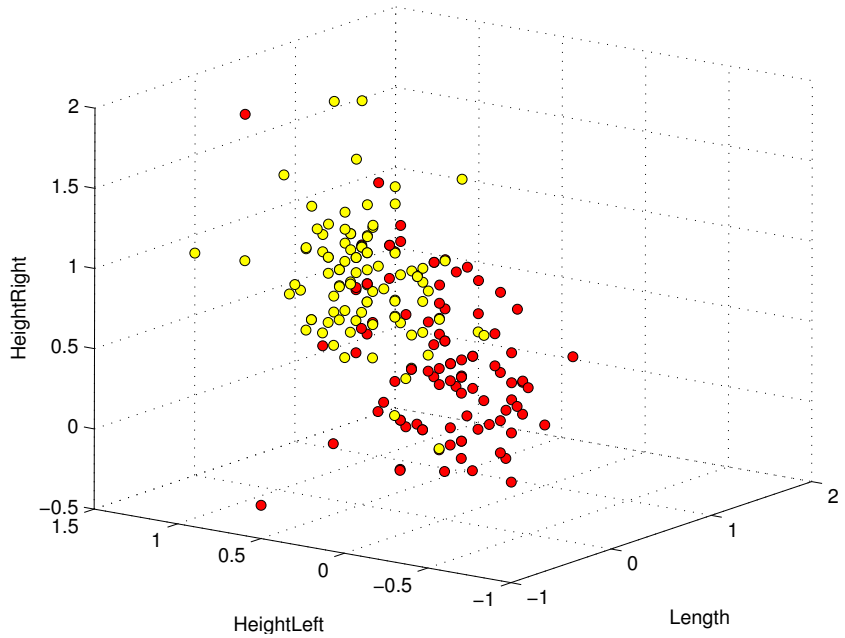


Figure B.5: First three (out of six) features of Swiss Bank Notes data depicted in $\mathbb{R}^3$.

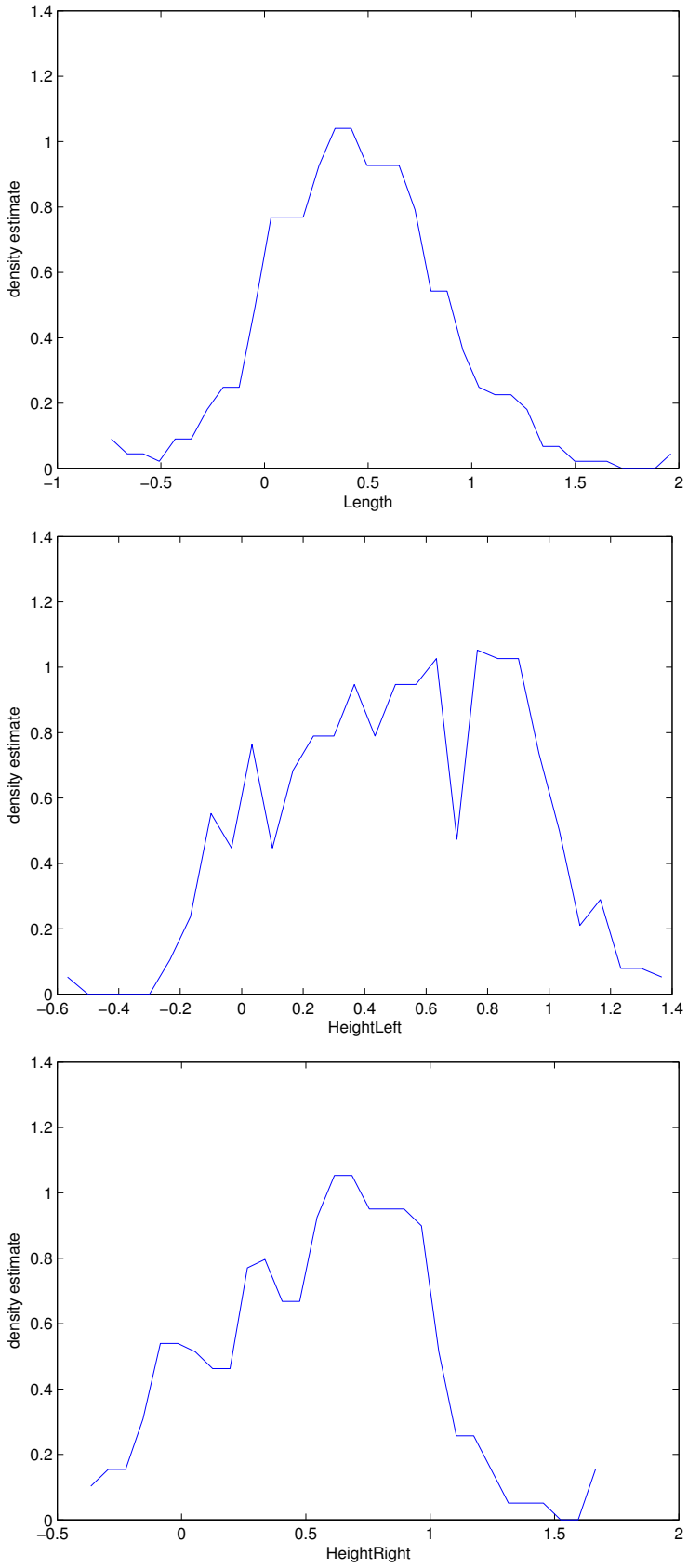


Figure B.6: Features' distributions of normalized Swiss Banknotes data.

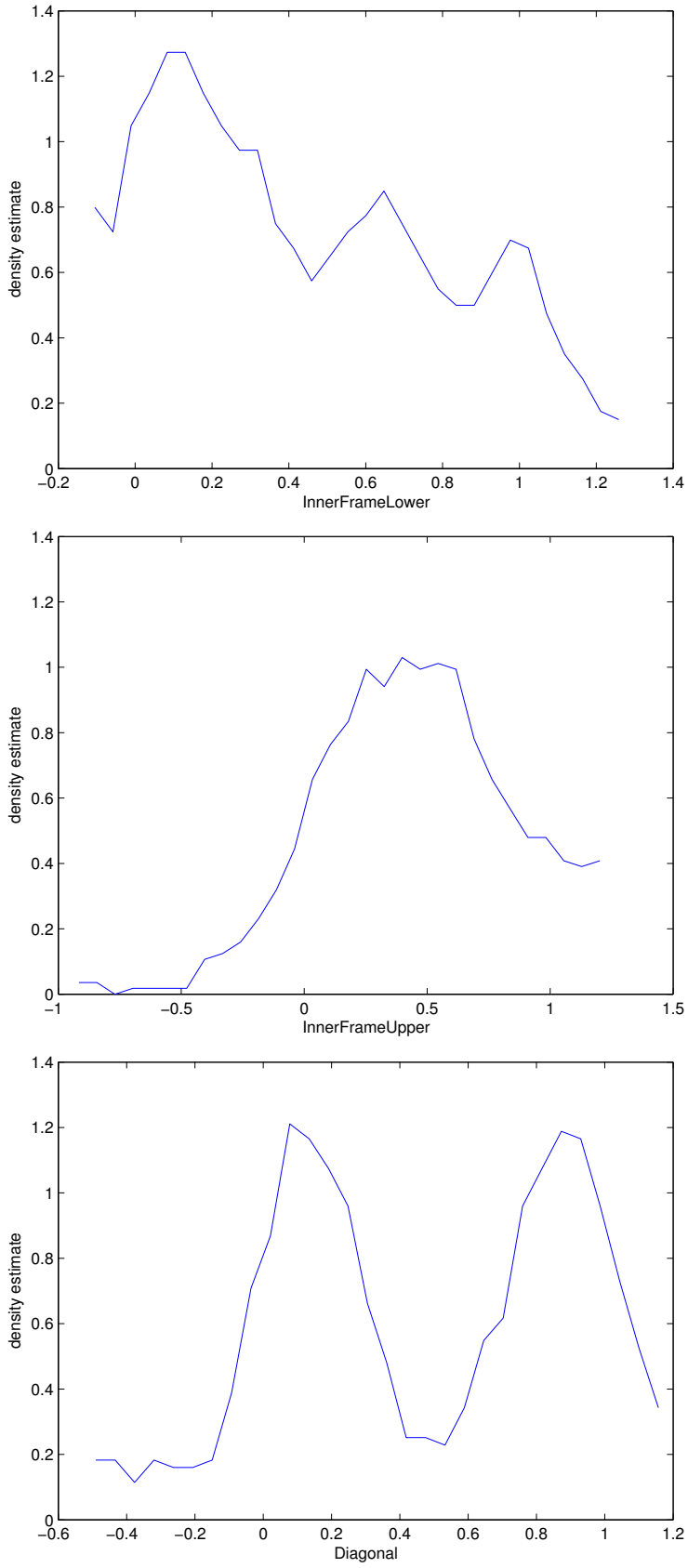
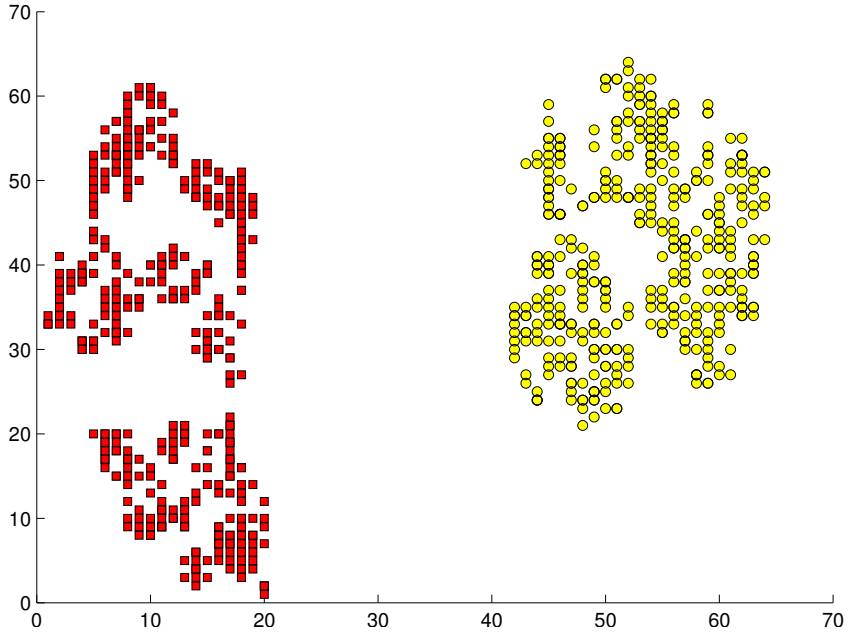


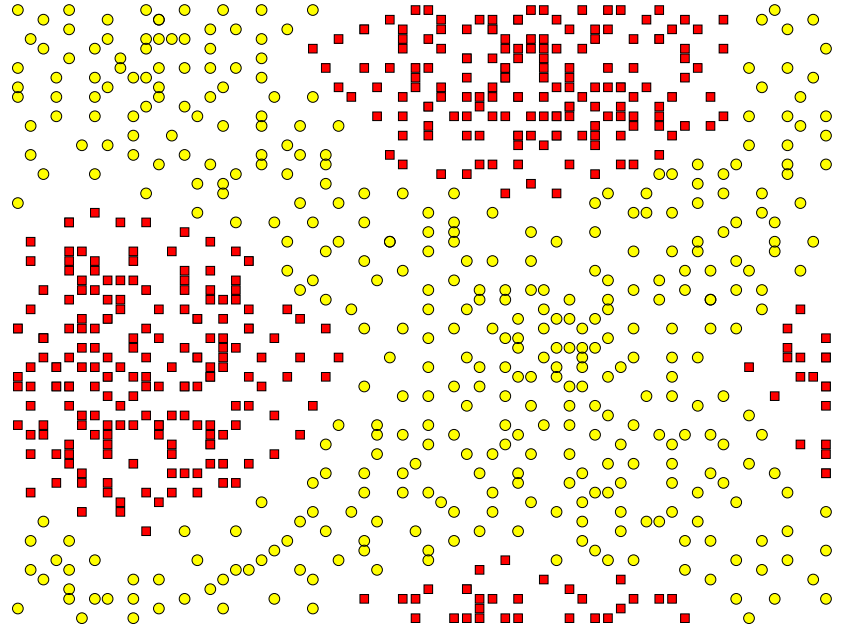
Figure B.6: Features' distributions of normalized Swiss Banknotes data.

B.2 Illustration

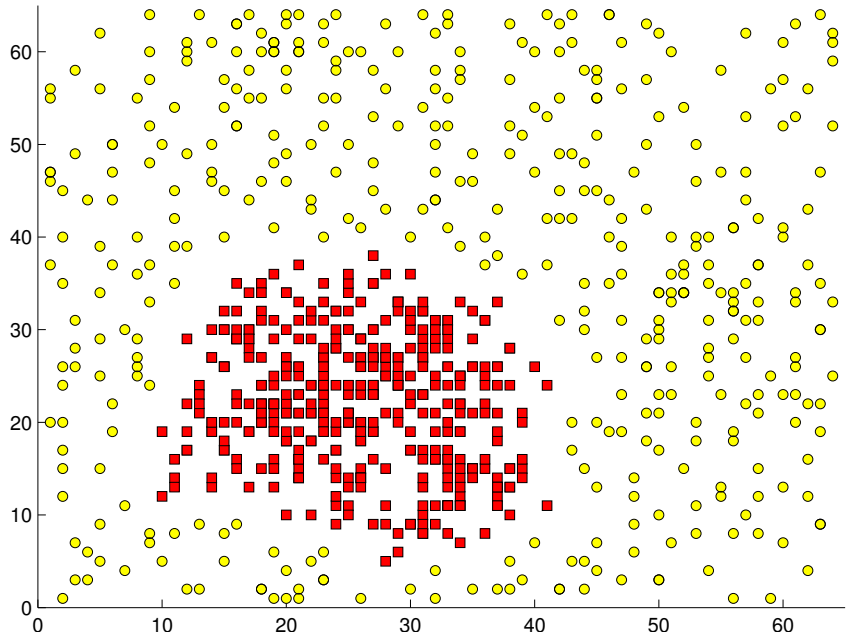


(a) t-SNE

Figure B.7: Atom data. t-SNE clearly separates the two classes, but disregards the proximity of the hull (yellow). Batch-SOM falsely depicts two cores on a toroid grid. SOP correctly assembles a single core on a toroid output space

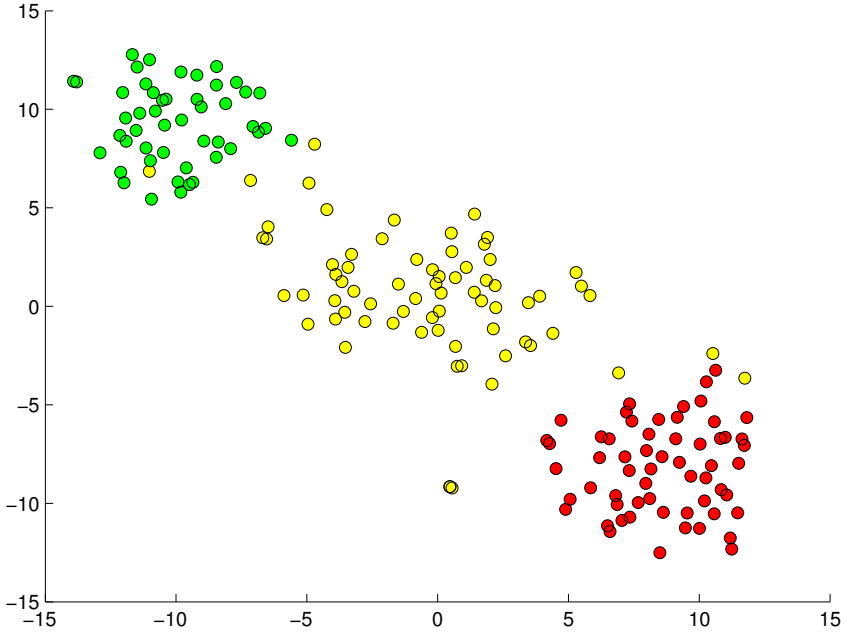


(a) Batch-SOM

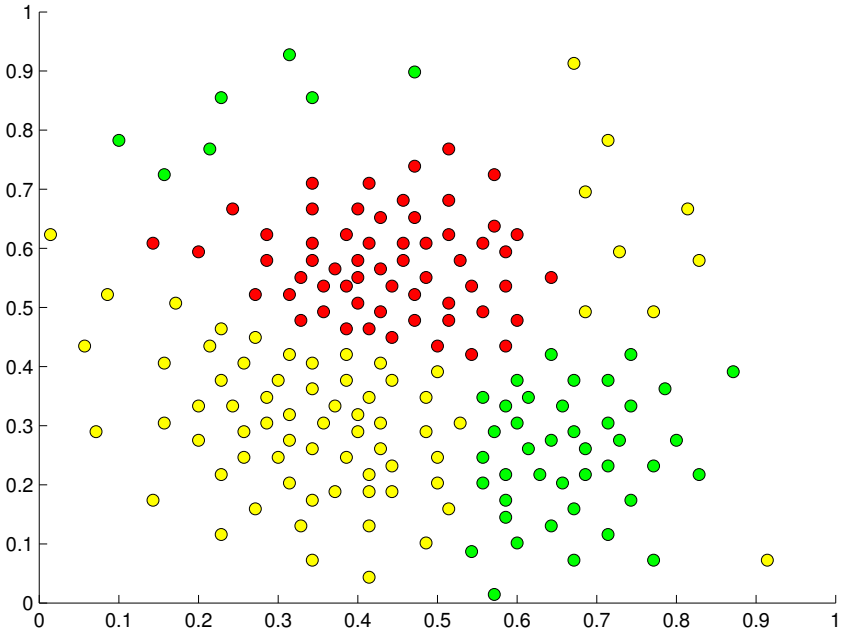


(b) SOP

Figure B.7: Atom data. t-SNE clearly separates the two classes, but disregards the proximity of the hull (yellow). Batch-SOM falsely depicts two cores on a toroid grid. SOP correctly assembles a single core on a toroid output space

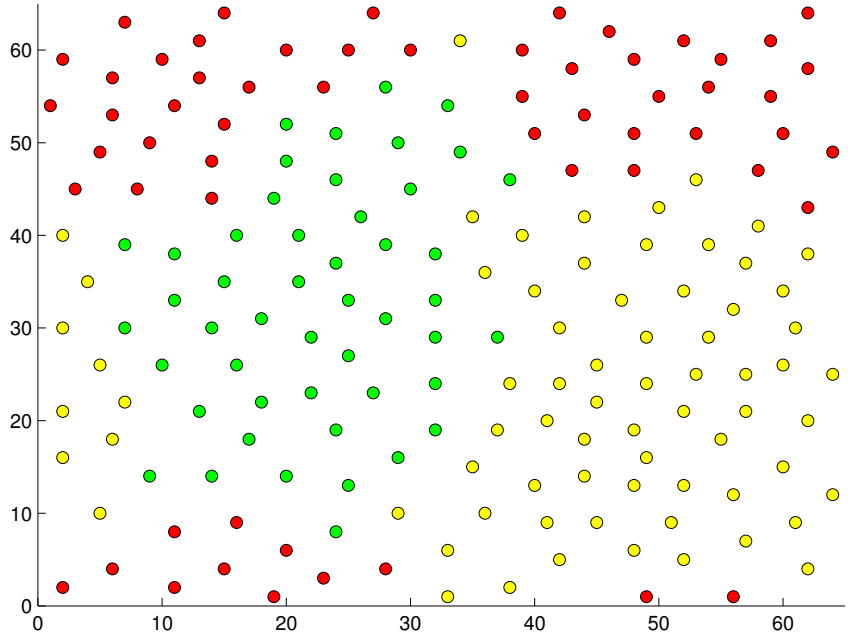


(a) t-SNE



(b) CCA

Figure B.8: Wine data. t-SNE clearly separates three classes, but produces minor inclusions. CCA disrupts classes.



(a) SOP

Figure B.8: Wine data. SOP correctly assembles the classes on a toroid output space.

B.3 Statistical Testing

	CCA	Batch-SOM	t-SNE	SOP
CCA	1	1	1	1
Batch-SOM	$7.7528e-046$	1	$7.7528e-046$	$7.7528e-046$
t-SNE	1	1	1	1
SOP	1	1	1	1

Table B.1: Atom data: asymptotic probabilities p_{ij} of two-sample KS test. Alternative hypothesis is: method$_i$ leads to larger dispersion values on an average than method$_j$.

	CCA	Batch-SOM	t-SNE	SOP
CCA	1	$1.3814e-040$	$2.9782e-006$	$1.3814e-040$
Batch-SOM	1	1	1	1
t-SNE	$6.0578e-008$	$5.7571e-017$	1	$1.9816e-016$
SOP	0.9897	0.9897	1	1

Table B.2: Chainlink data: asymptotic probabilities p_{ij} of two-sample KS test. Alternative hypothesis is: method$_i$ leads to larger dispersion values on an average than method$_j$.

	CCA	Batch-SOM	t-SNE	SOP
CCA	1	0.9108	1	1
Batch-SOM	0.5144	1	1	1
t-SNE	$1.4245e-006$	$2.9782e-006$	1	$2.9782e-006$
SOP	$5.2767e-012$	$1.1028e-008$	0.0966	1

Table B.3: Iris data: Asymptotic probabilities p_{ij} of two-sample KS test. Alternative hypothesis is: method$_i$ leads to larger dispersion values on an average than method$_j$.

	CCA	Batch-SOM	t-SNE	SOP
CCA	1	0.6011	1	0.9897
Batch-SOM	0.0157	1	1	1
t-SNE	$7.7528e-046$	$4.7406e-044$	1	$4.4051e-038$
SOP	$1.2231e-005$	0.00029	1	1

Table B.4: EngyTime data: Asymptotic probabilities p_{ij} of two-sample KS test. Alternative hypothesis is: method$_i$ leads to larger dispersion values on an average than method$_j$.

	CCA	Batch-SOM	t-SNE	SOP
CCA	1	$7.5783e-018$	1	$4.1181e-024$
Batch-SOM	1	1	1	0.009
t-SNE	$4.6498e-010$	$1.5196e-049$	1	$1.8606e-040$
SOP	1	0.9695	1	1

Table B.5: Swissbanknotes data: Asymptotic probabilities p_{ij} of two-sample KS test. Alternative hypothesis is: method$_i$ leads to larger dispersion values on an average than method$_j$.

	CCA	Batch-SOM	t-SNE	SOP
CCA	1	$1.1652e-035$	$1.4791e-032$	$7.0228e-029$
Batch-SOM	1	1	0.9897	1
t-SNE	1	$1.6382e-017$	1	$1.3762e-007$
SOP	1	0.0041	0.4311	1

Table B.6: Wine data: Asymptotic probabilities p_{ij} of two-sample KS test. Alternative hypothesis is: method$_i$ leads to larger dispersion values on an average than method$_j$.

B.4 Dispersion

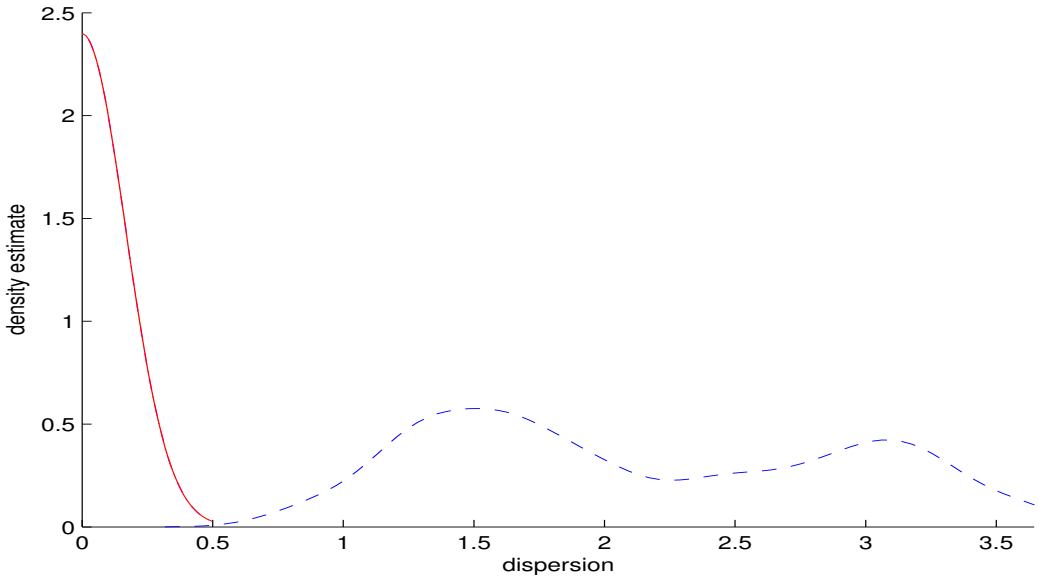


Figure B.9: Atom data, distribution of dispersions: SNE (dash-dot, magenta), CCA (dotted, green), Batch-SOM (dashed, blue) and SOP (solid, red).

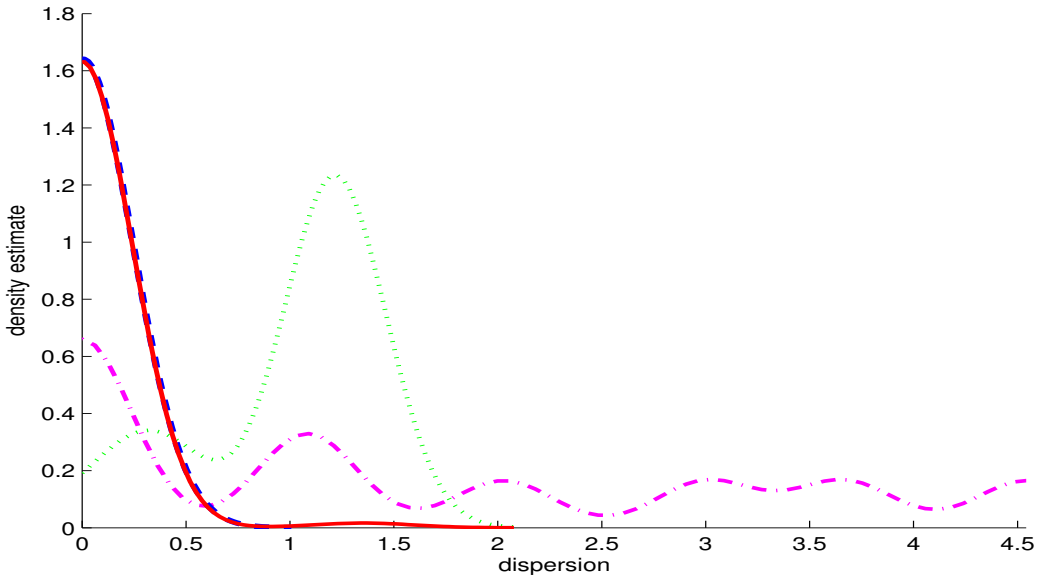


Figure B.10: Chainlink data, distribution of dispersions: SNE (dash-dot, magenta), CCA (dotted, green), Batch-SOM (dashed, blue) and SOP (solid, red).

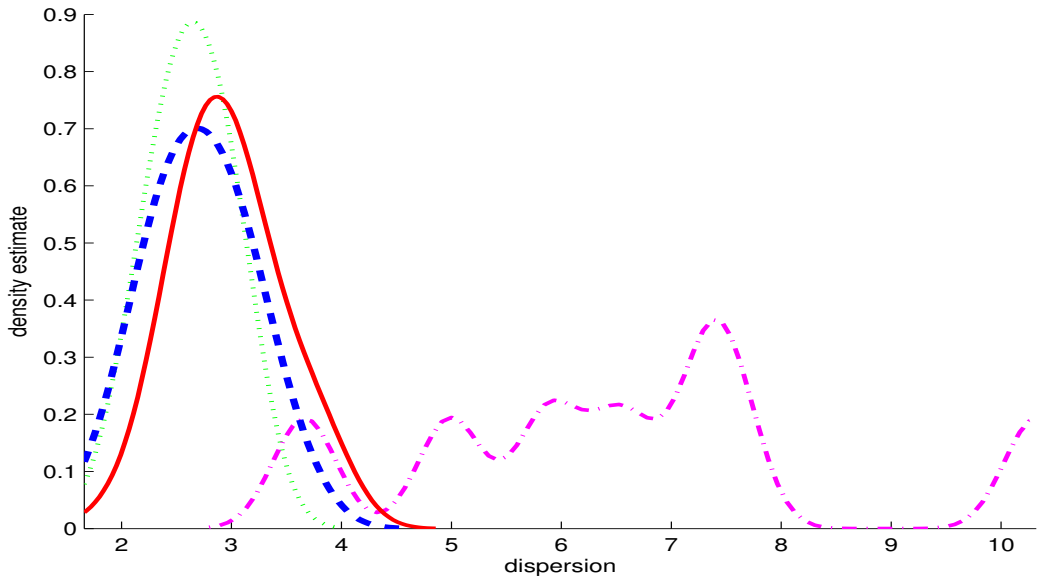


Figure B.11: EngyTime data, distribution of dispersions: SNE (dash-dot, magenta), CCA (dotted, green), Batch-SOM (dashed, blue) and SOP (solid, red).

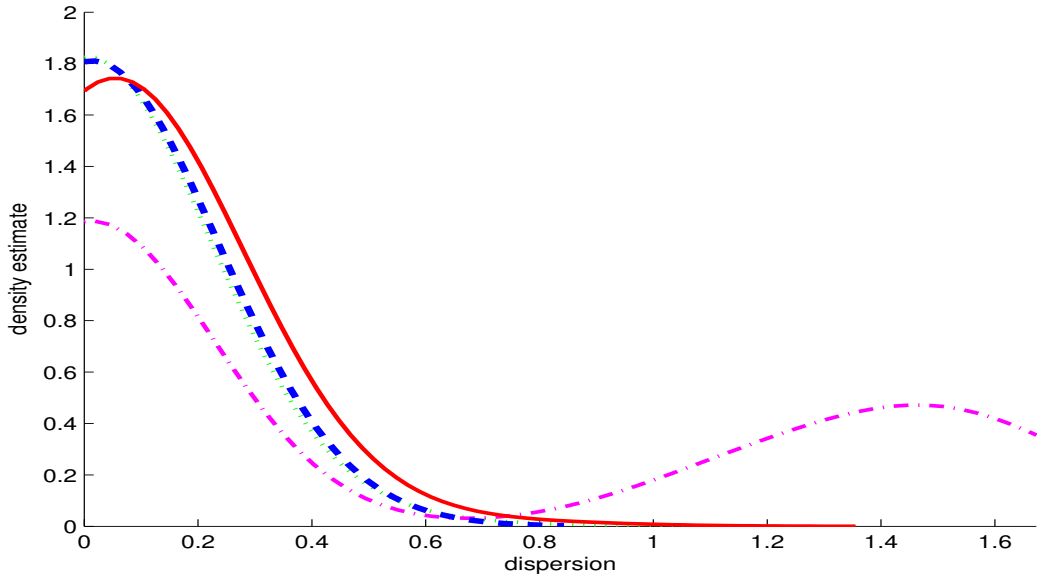


Figure B.12: Iris data, distribution of dispersions: SNE (dash-dot, magenta), CCA (dotted, green), Batch-SOM (dashed, blue) and SOP (solid, red).

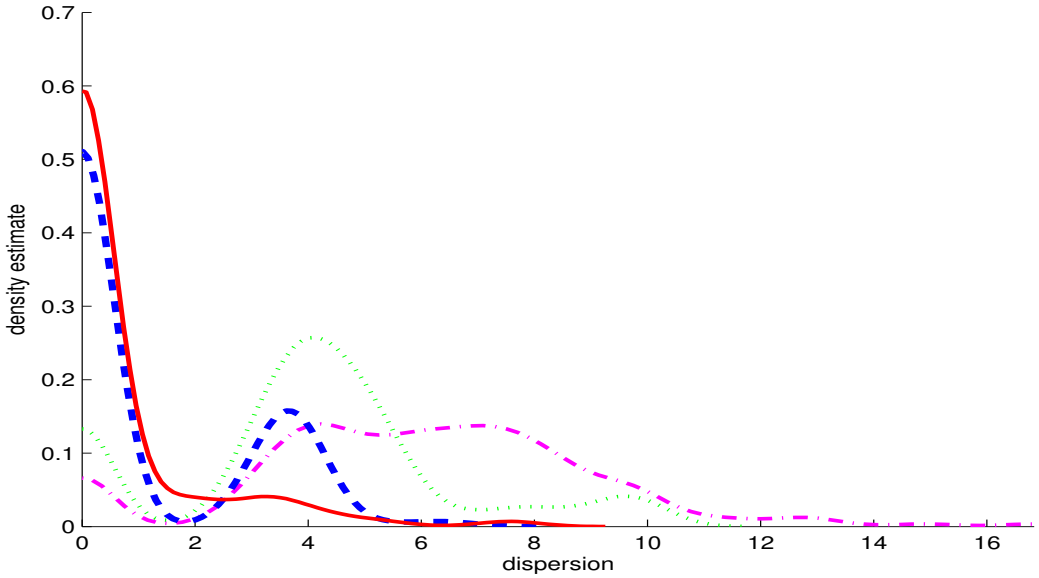


Figure B.13: SwissBanknotes data, distribution of dispersions: SNE (dash-dot, magenta), CCA (dotted, green), Batch-SOM (dashed, blue) and SOP (solid, red).

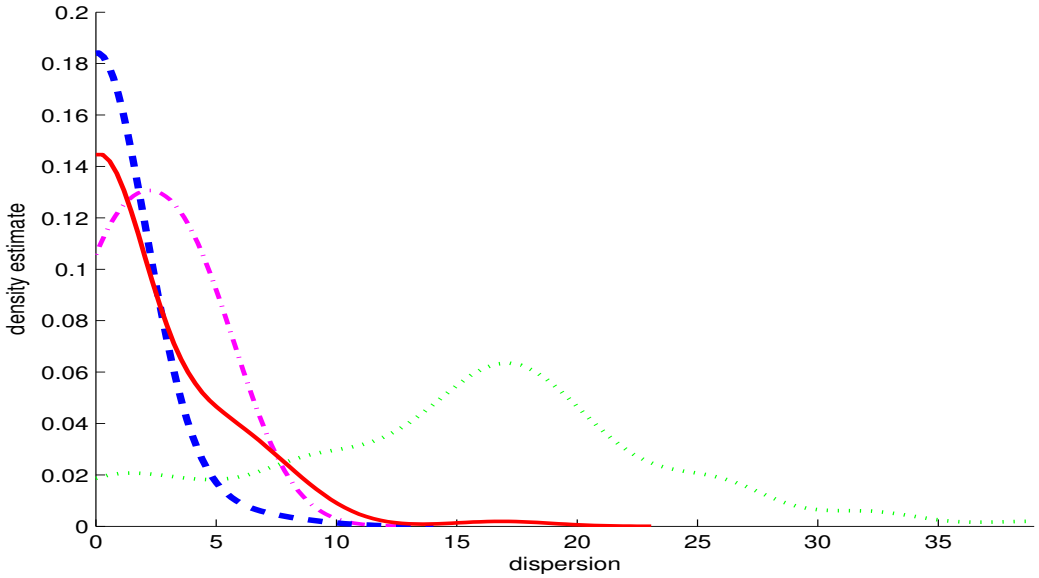


Figure B.14: Wine data, distribution of dispersions: SNE (dash-dot, magenta), CCA (dotted, green), Batch-SOM (dashed, blue) and SOP (solid, red).

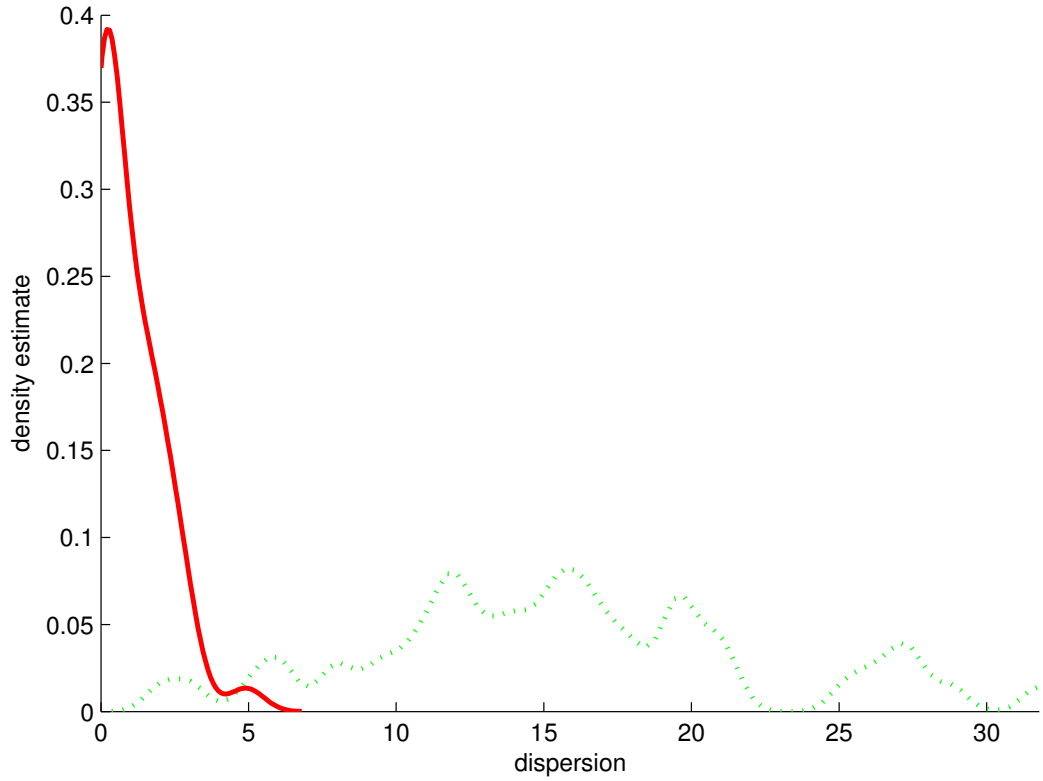
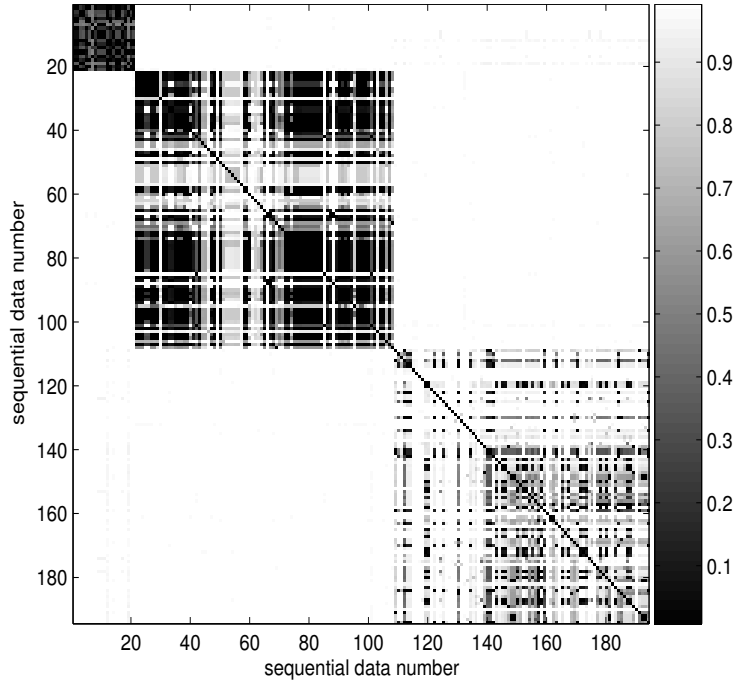


Figure B.15: GPD194 data, distribution of dispersions: CCA (dotted, green) and SOP (solid, red).

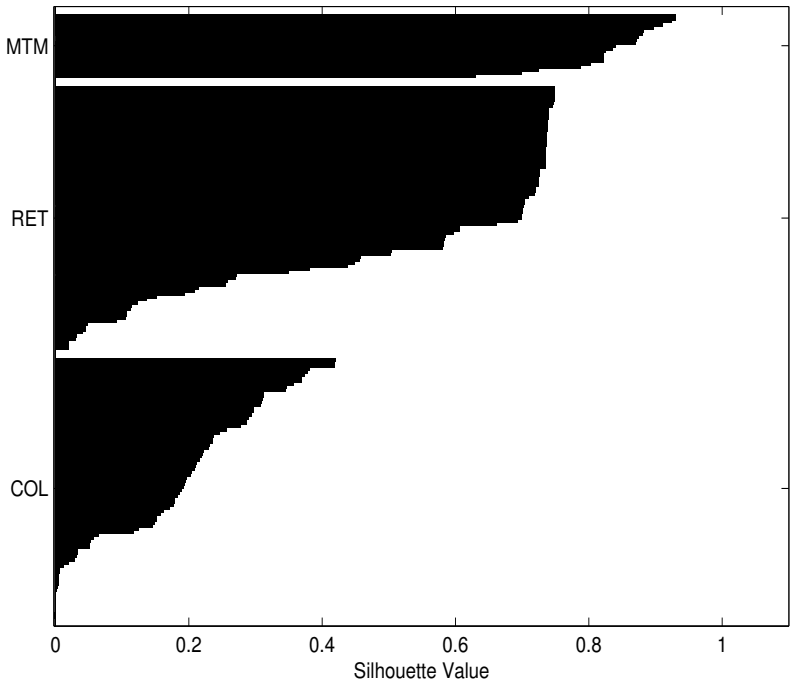
B.5 GPD194 Data

Protein family	# Genes	Gene symbols	# Protein sequences
myotubularin	7	MTMR1-4 MTRM6-8	21
receptor precursor	7	FGFR1-4 RET TEK TIE1	87
collagen alpha chain	13	COL1A2 COL21A1 COL24A1 COL27A1 COL2A1 COL3A1 COL4A1 COL4A2 COL4A3 COL4A6 COL5A3 COL9A1 COL9A2	86

Table B.7: Characteristics of the GPD194 dataset by Popescu et al. [PKM06]

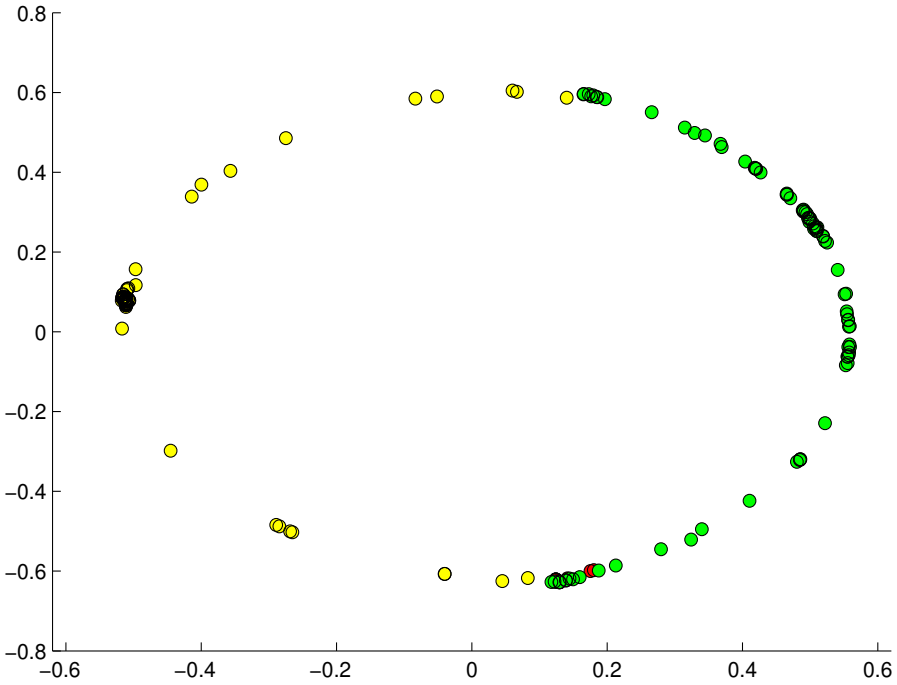


(a)

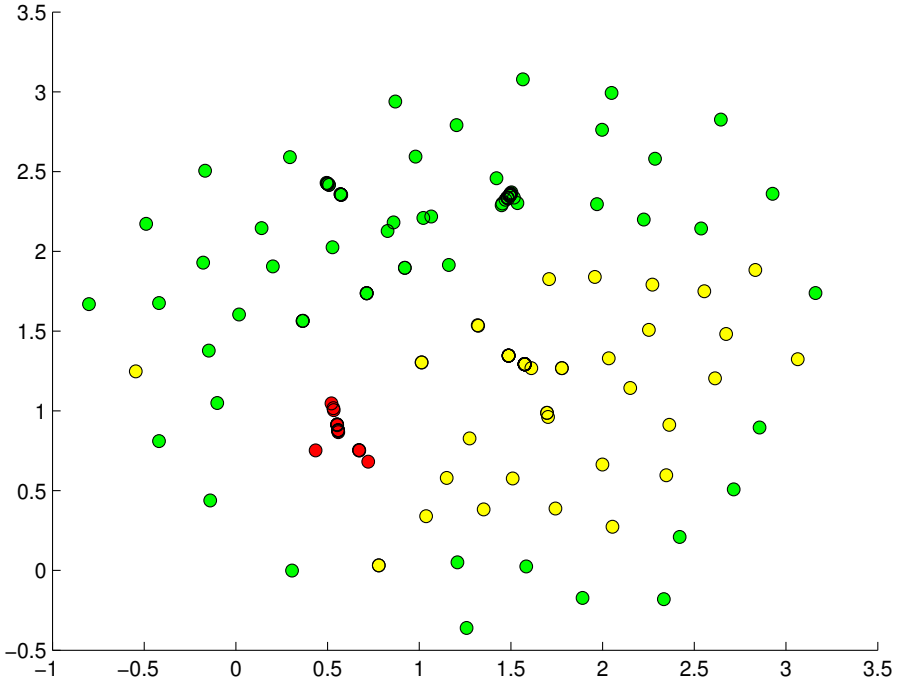


(b)

Figure B.16: GPD194 dataset [PKM06], proteins are sorted by family (myotubularins, receptor precursors, collagen alpha chains) and by gene. (a) Dissimilarity matrix. (b) Silhouette plot.

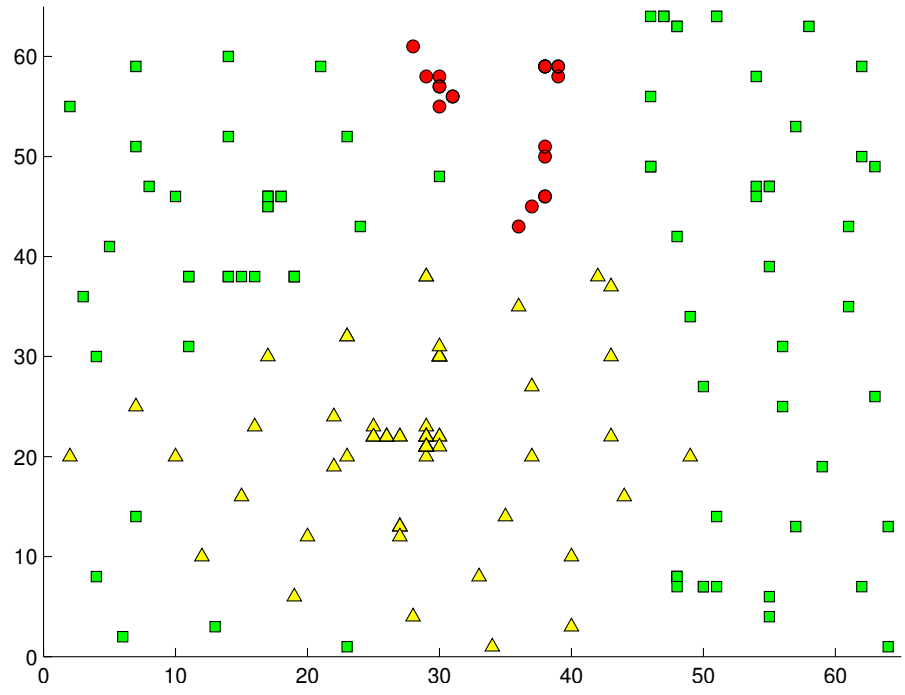


(a) MDS

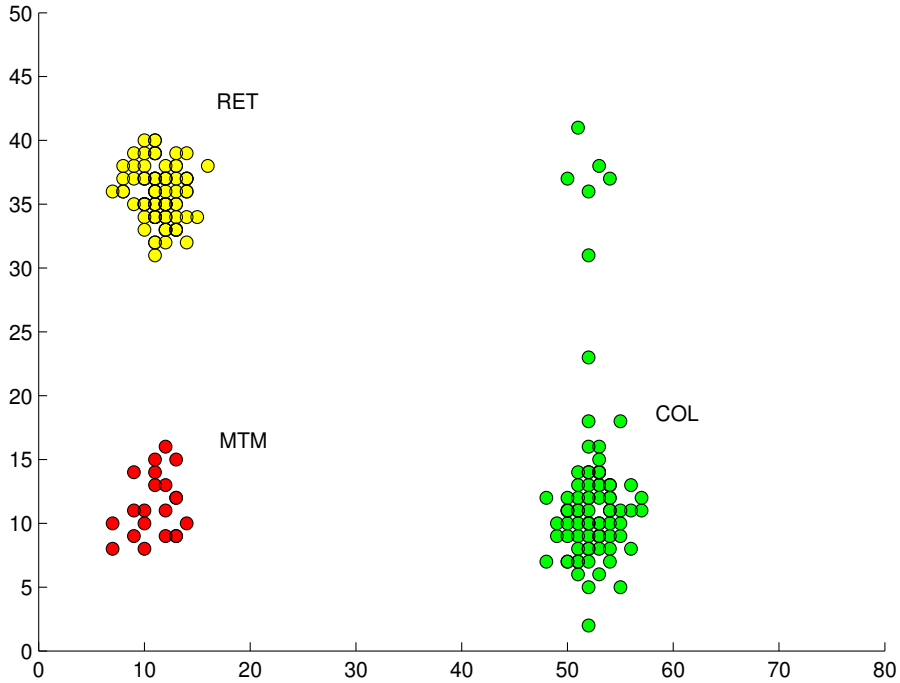


(b) CCA

Figure B.17: GPD194 data mapped by minimization of squared Kruskal stress (cf. [Kru64]). No well-separated clusters are depicted. Despite its distance-preserving intention, CCA depicts no well-separated clusters.



(a) SOP



(b) SOP'

Figure B.18: SOP cohesively maps GPD194 data onto toroid map. Large neighbourhood radius causes SOP' to form well-separated piles of myotubularins (MTM), receptor precursors (RET) and collagens (COL) Outliers of collagen class assemble near the COL pile.

Appendix C

Genes for microRNA

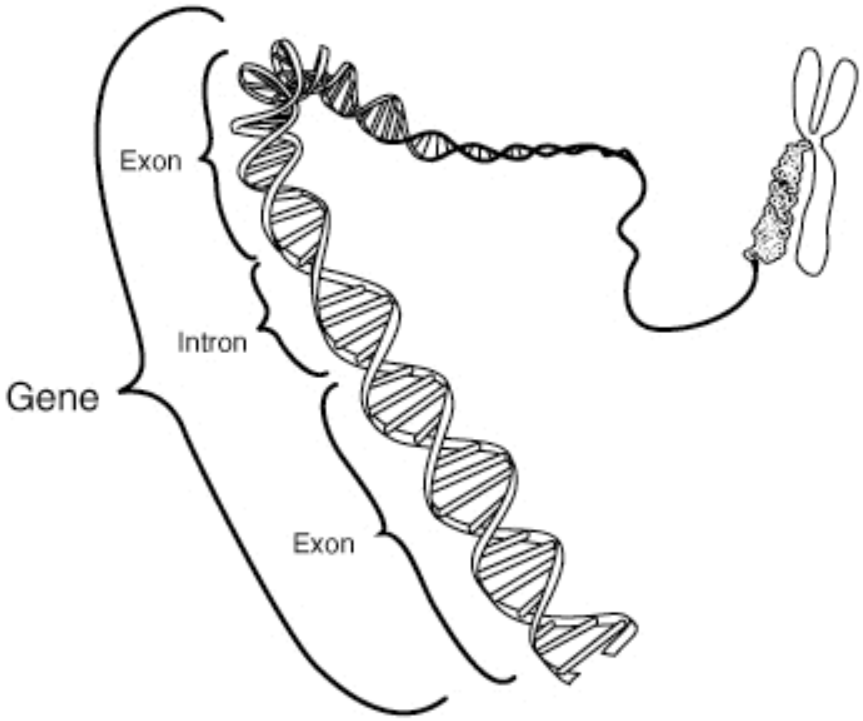


Figure C.1: The double helix structure of DNA in relation to a chromosome on the right. Image by National Human Genome Research Institute.

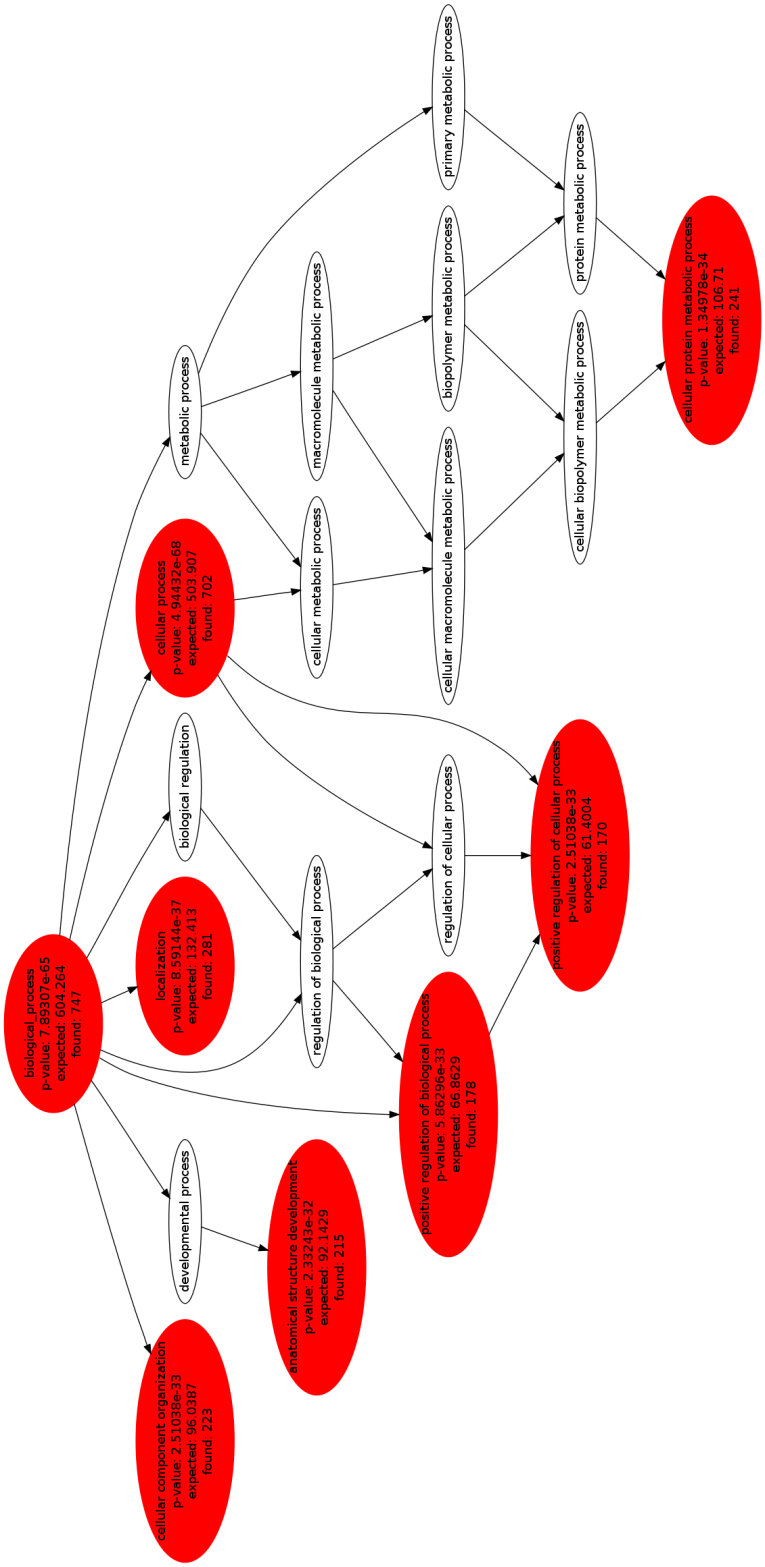


Figure C.2: Most significant terms from "biological process" ontology annotated to miRNA-regulated 749 genes. Image obtained by GeneTrail [BKK+07].

Appendix D

Fundamental Analysis

FndOwn

The percentage of common shares owned by mutual funds. It is derived by dividing the aggregate number of company shares owned by mutual funds by the total shares outstanding, and multiplying by 100. The Box-Cox tranformation suggests that the square root of FndOwn values is roughly normally distributed.

ShrOut

This is the most recent figure for the total number of common shares outstanding, denoted in millions. The logarithm of ShrOut is sufficiently normally distributed.

MktCap

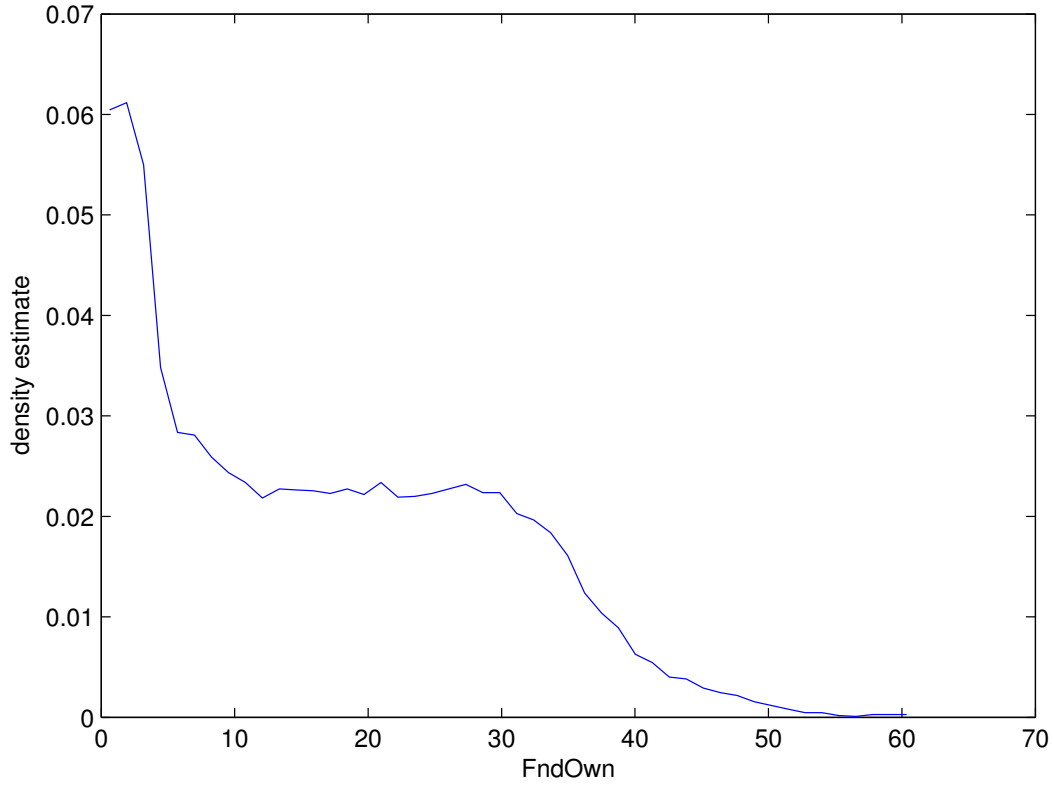
The current stock-market value of a company's equity, in millions. It is calculated by multiplying the current share price by the number of shares outstanding as of the most recently completed fiscal quarter. The logarithm of MktCap is sufficiently normally distributed.

DTotYe

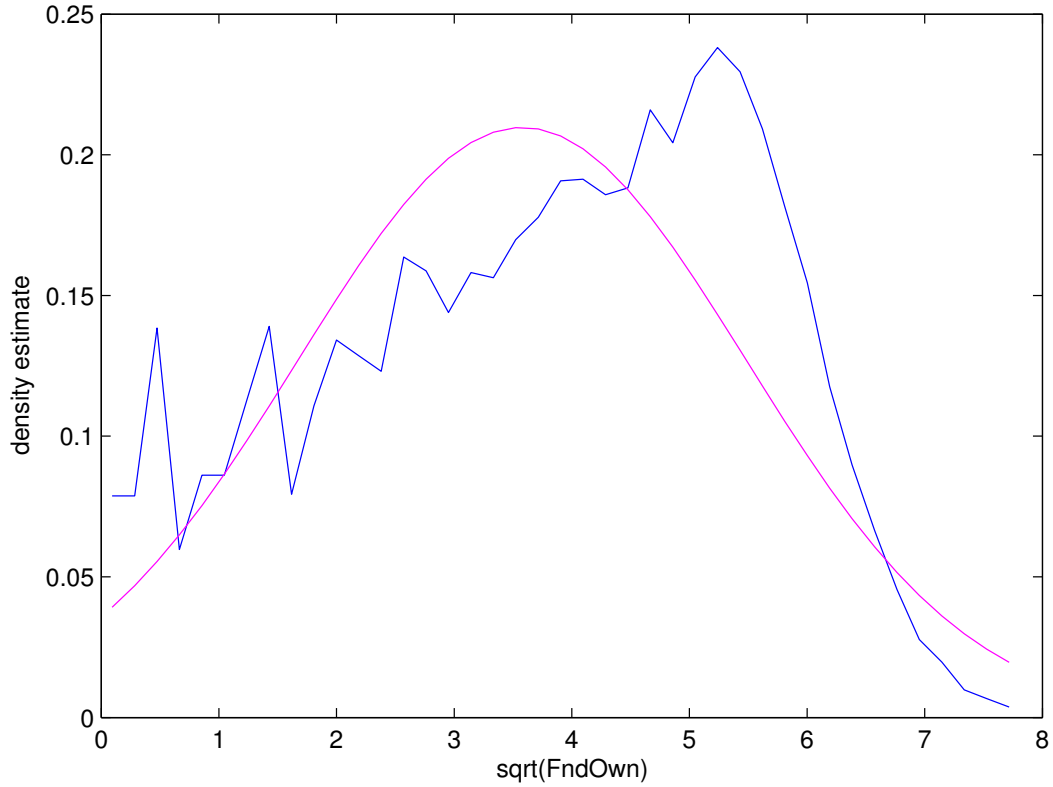
Listed for Year One and Two, this ratio is calculated by dividing long-term debt (excluding other liabilities) by total capitalization (the sum of common equity plus preferred equity plus long-term debt). This value is not provided for financial companies. The Box-Cox tranformation suggests that the square root of DTotYe values is roughly normally distributed.

ADV

The average daily trading volume of common shares during the trailing 12 months. This is expressed in unit shares. The logarithm of ADV is sufficiently normally distributed.

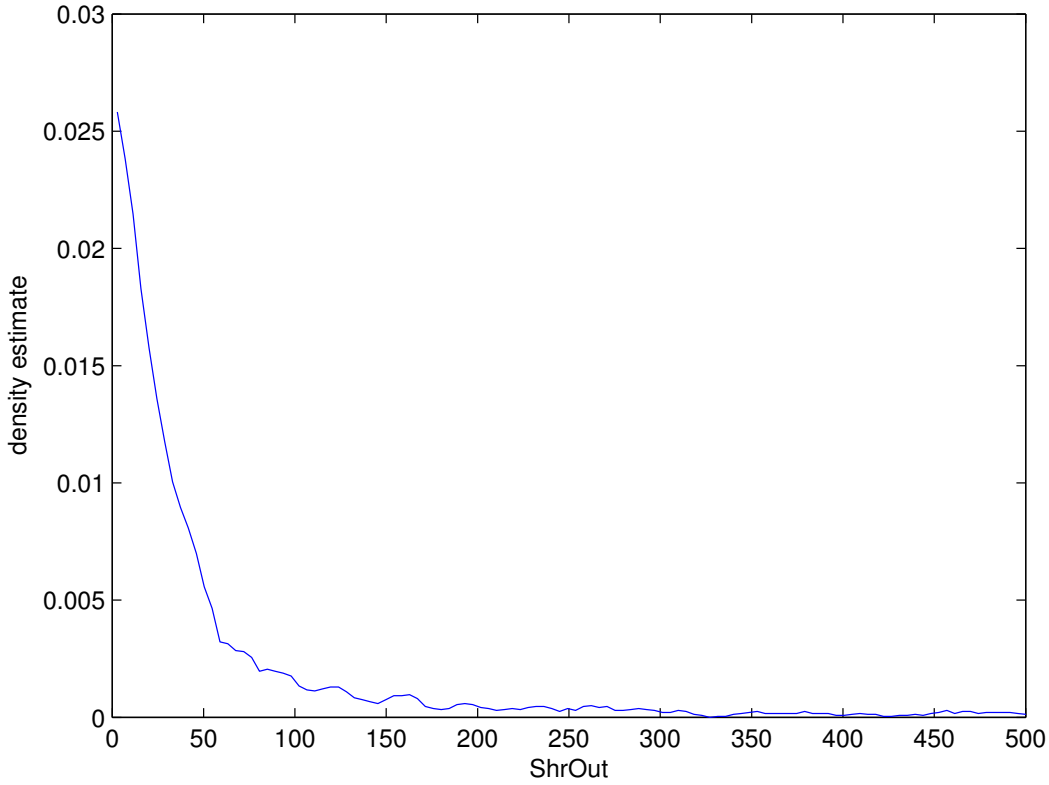


(a) FndOwn

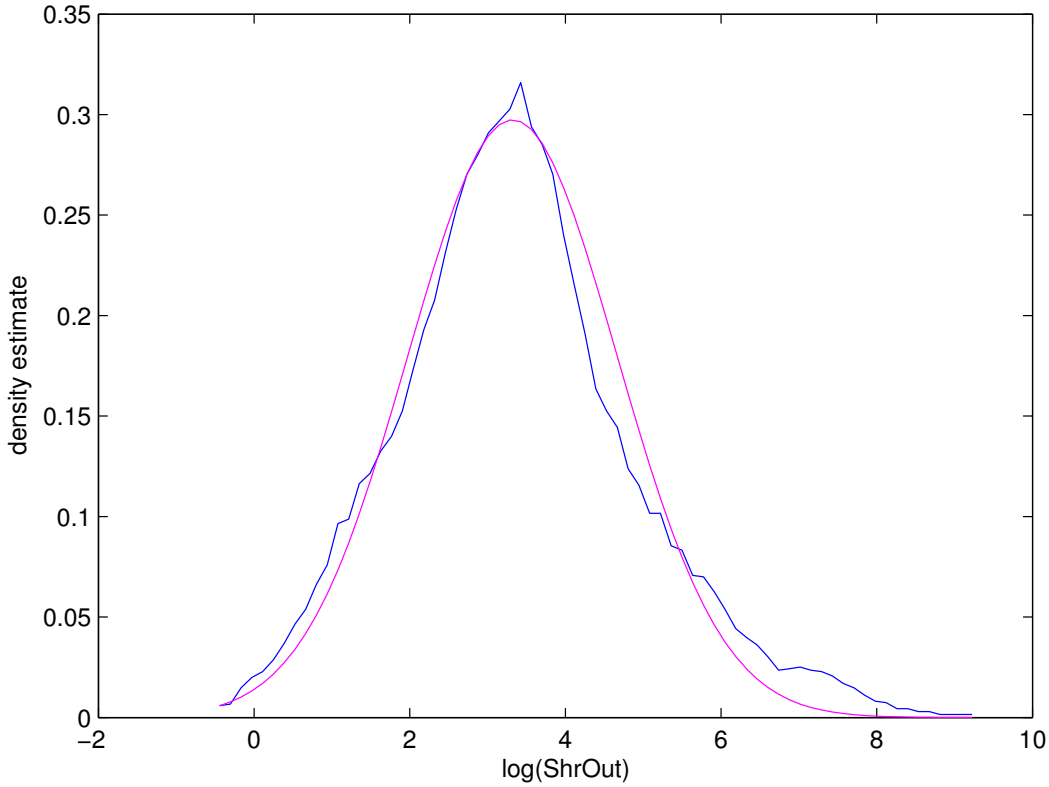


(b) square root of FndOwn

Figure D.1: Feature Distribution: percentage of common shares owned by mutual funds

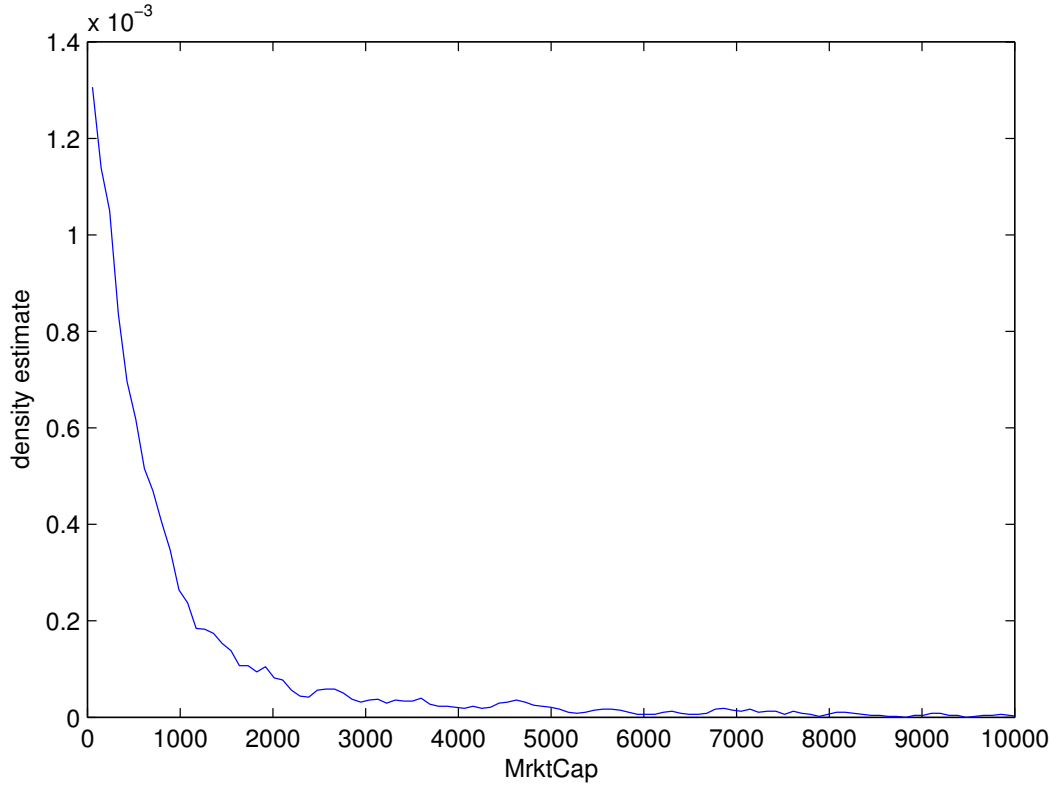


(a) ShrOut

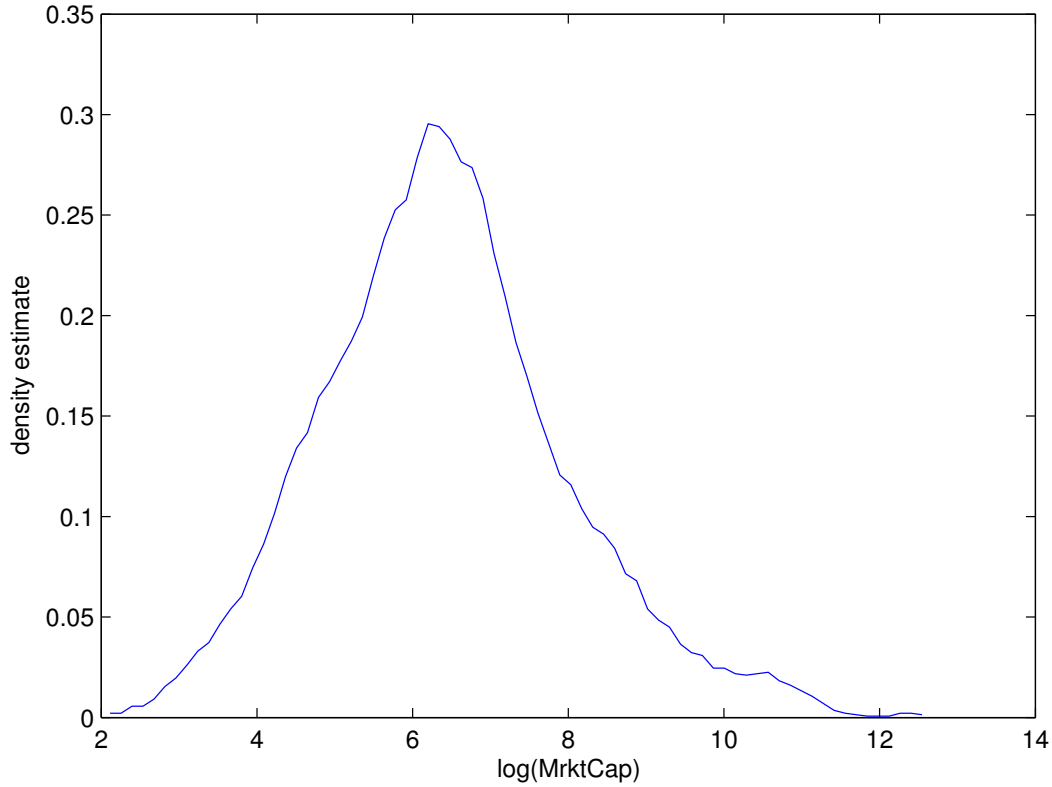


(b) logarithmized ShrOut

Figure D.2: Feature Distribution: common shares outstanding

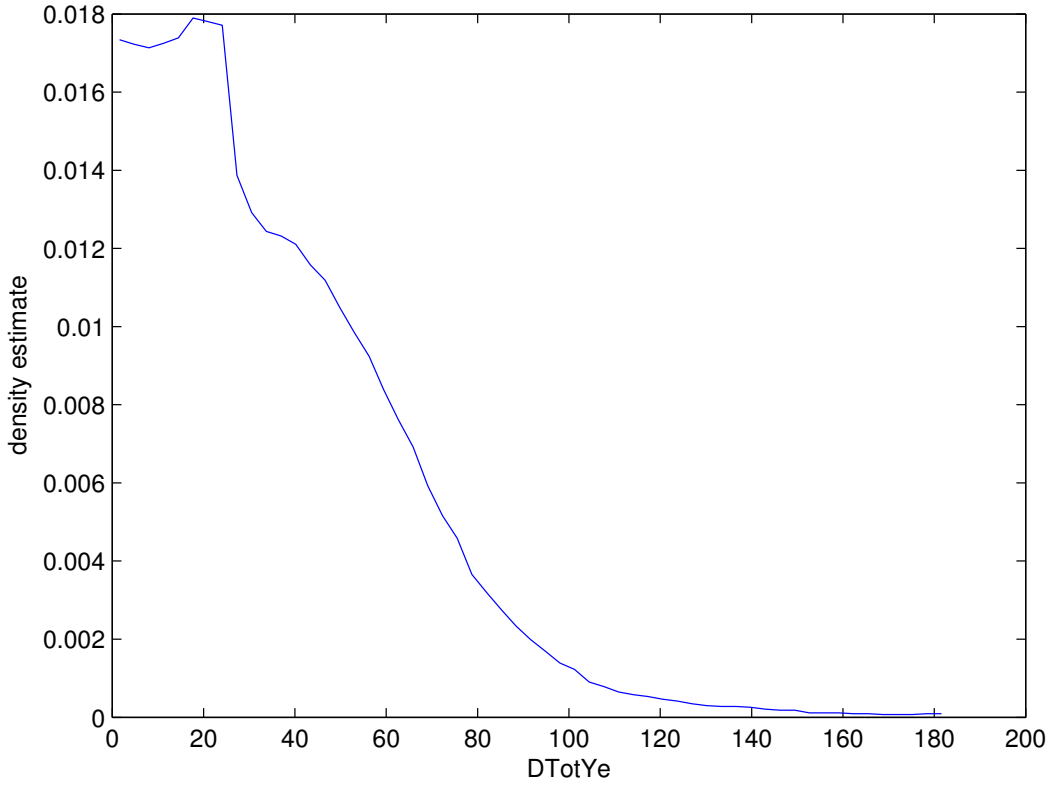


(a) MrktCap

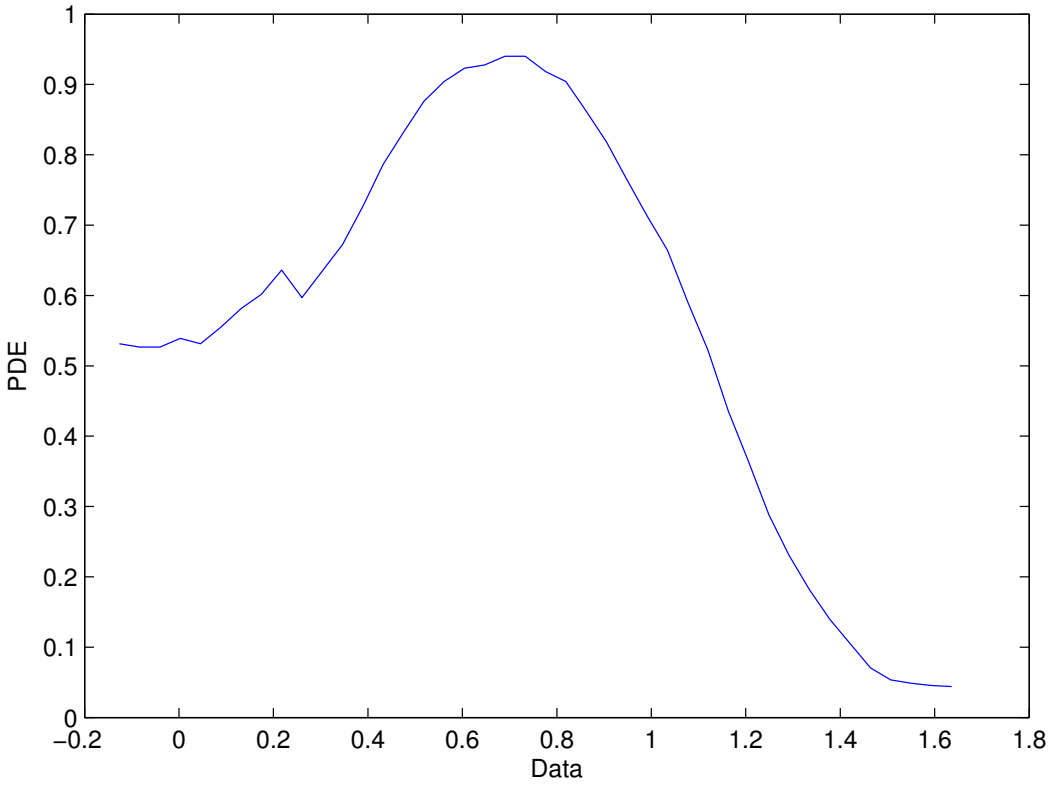


(b) logarithmized MrktCap

Figure D.3: Feature distribution: current stock-market value of a company's equity

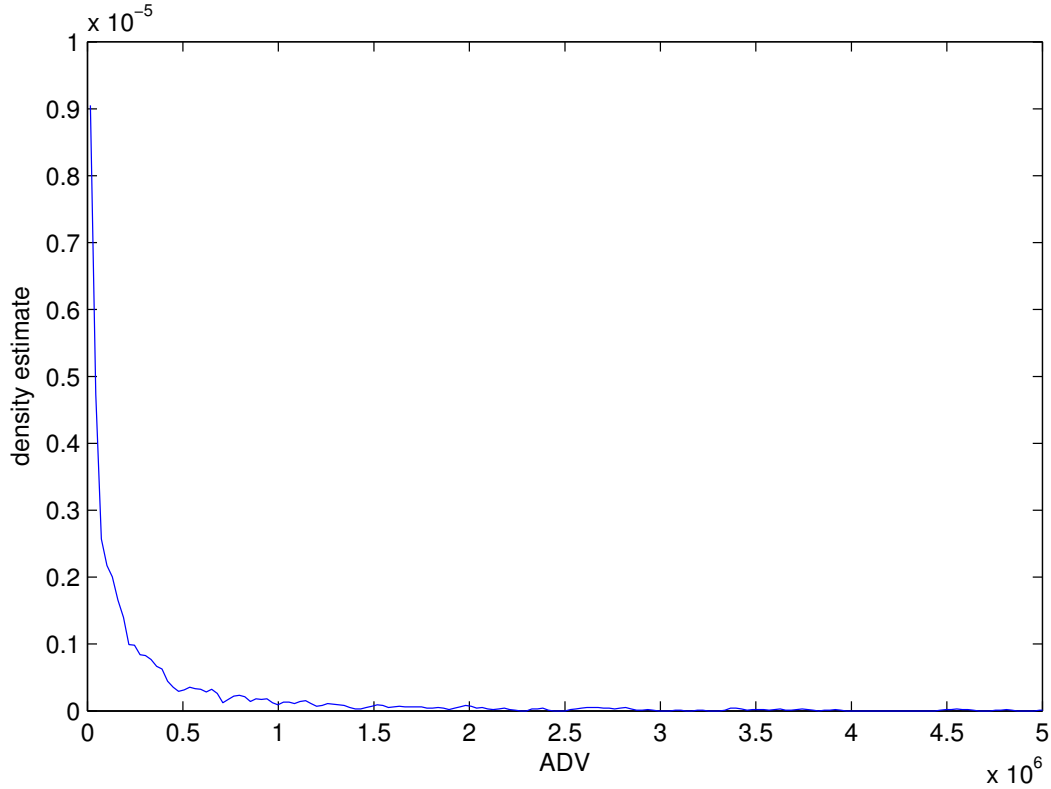


(a) DTotYe

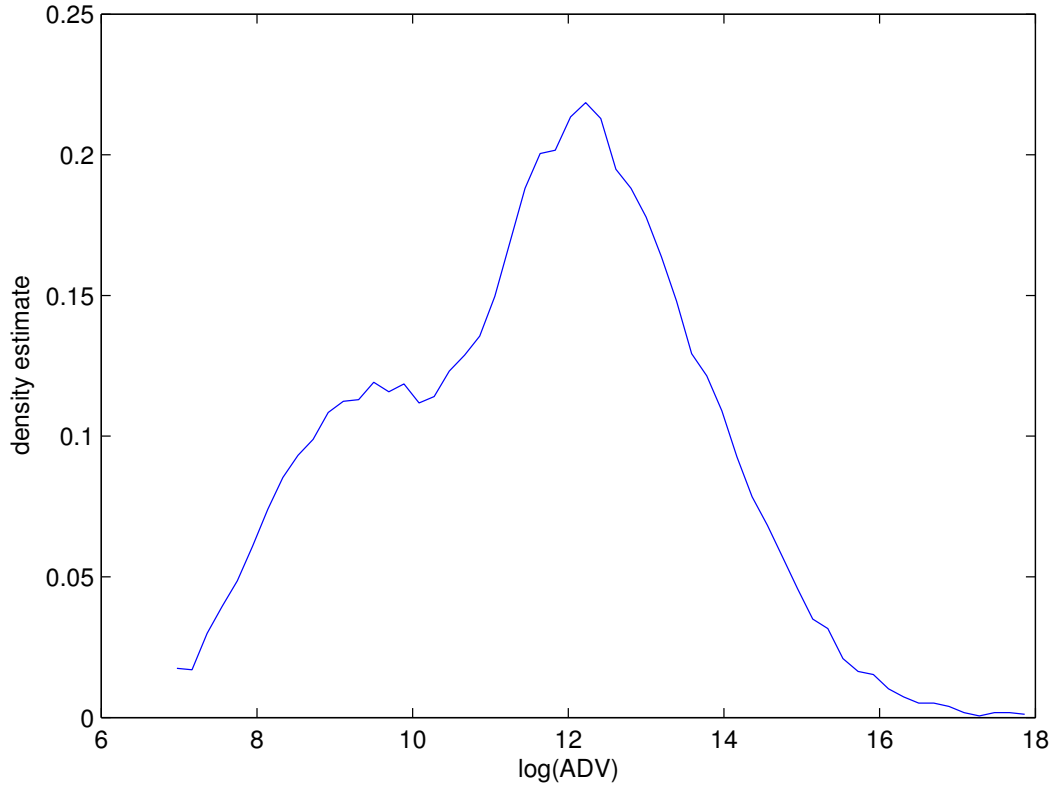


(b) transformed DTotYe

Figure D.4: Feature distribution: long-term debt divided by total capitalization



(a) ADV



(b) logarithmized ADV

Figure D.5: Feature distribution: average daily volume

PriCurr

The closing price at the end of the trading day on the relevant exchange as of the release date of Principia Stocks. The logarithm of PriCurr is sufficiently normally distributed.

Y1Y2

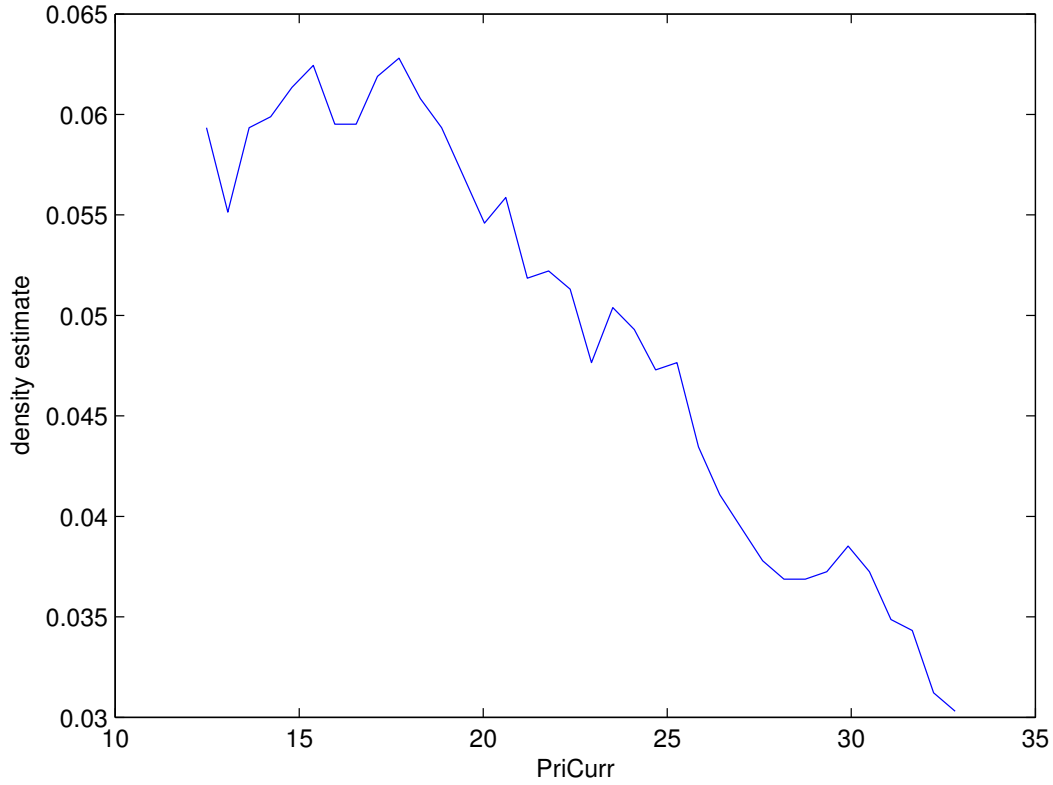
The annual percentage change in a company's revenues. The calculation is a given year's revenues minus the prior year's revenues, divided by the prior year's revenues. The resulting value is then multiplied by 100. Y1Y2 can be thought of as log-normally distributed because the percental change of revenues is the product of many small independent factors, calculated over periods of a year or more. The relative difference of revenue changes is used for further analysis.

Y2Y3

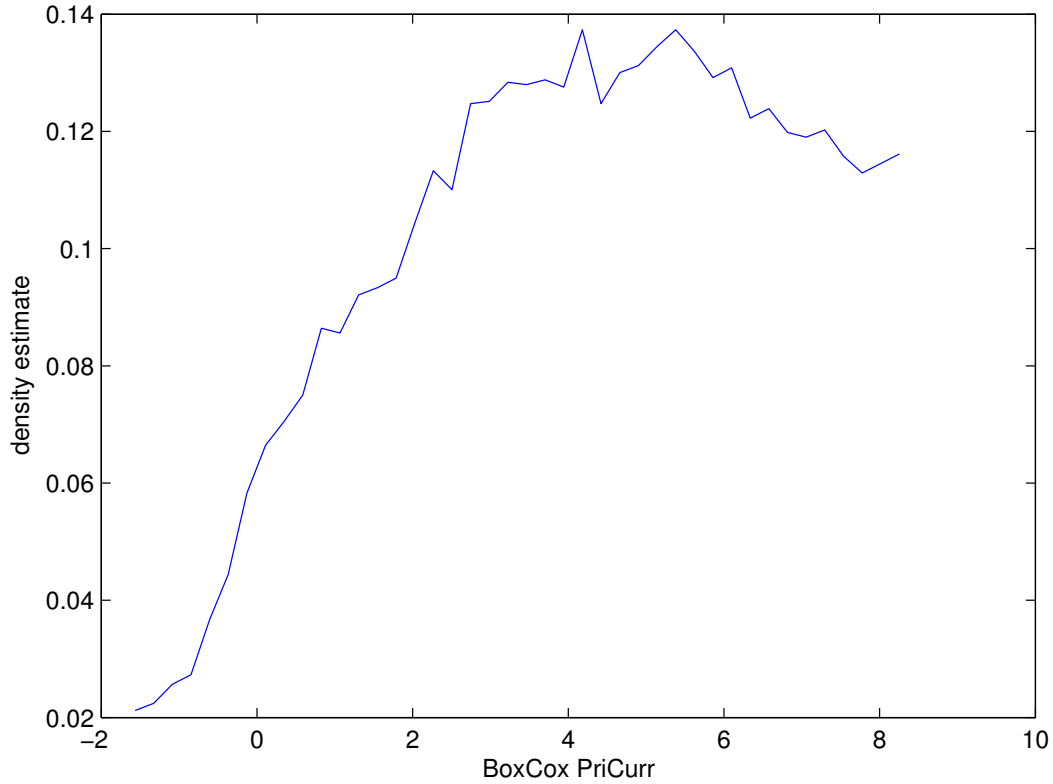
The annual percentage change in a company's revenues. The calculation is a given year's revenues minus the prior year's revenues, divided by the prior year's revenues. The resulting value is then multiplied by 100. Y2Y3 can be thought of as log-normally distributed because the percental change of revenues is the product of many small independent factors, calculated over periods of a year or more. The relative difference of revenue changes is used for further analysis.

Y3Y4

The annual percentage change in a company's revenues. The calculation is a given year's revenues minus the prior year's revenues, divided by the prior year's revenues. The resulting value is then multiplied by 100. Y3Y4 can be thought of as log-normally distributed because the percental change of revenues is the product of many small independent factors, calculated over periods of a year or more. The relative difference of revenue changes is used for further analysis.

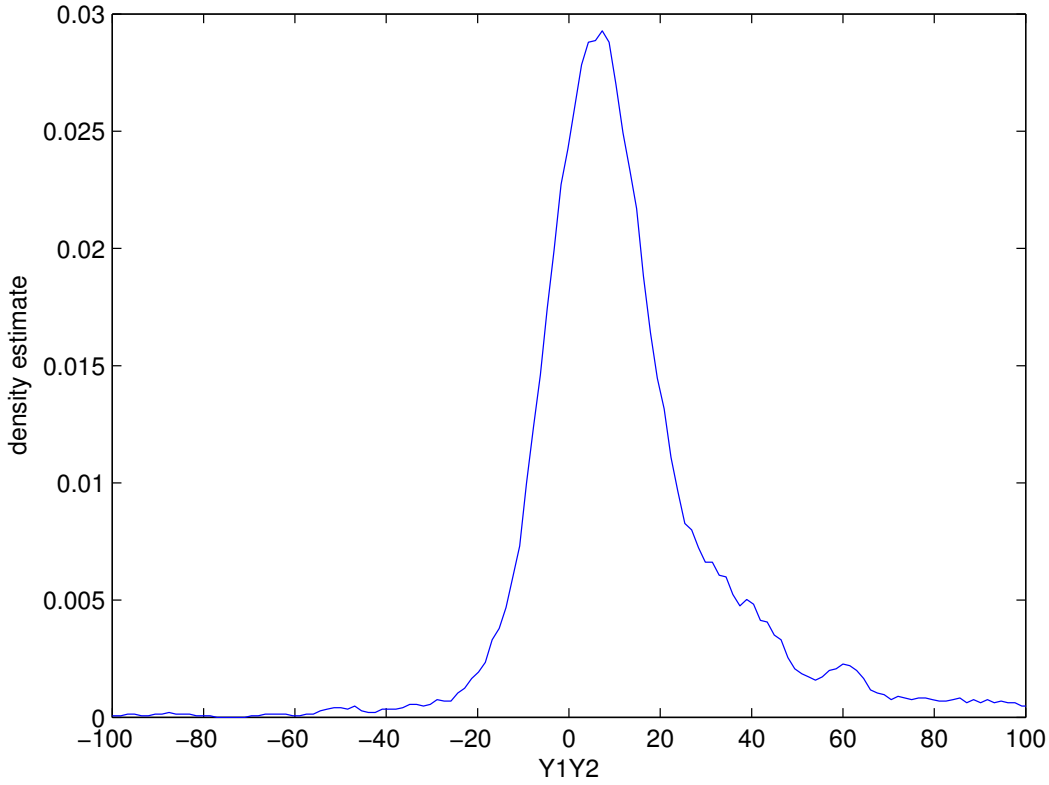


(a) PriCurr

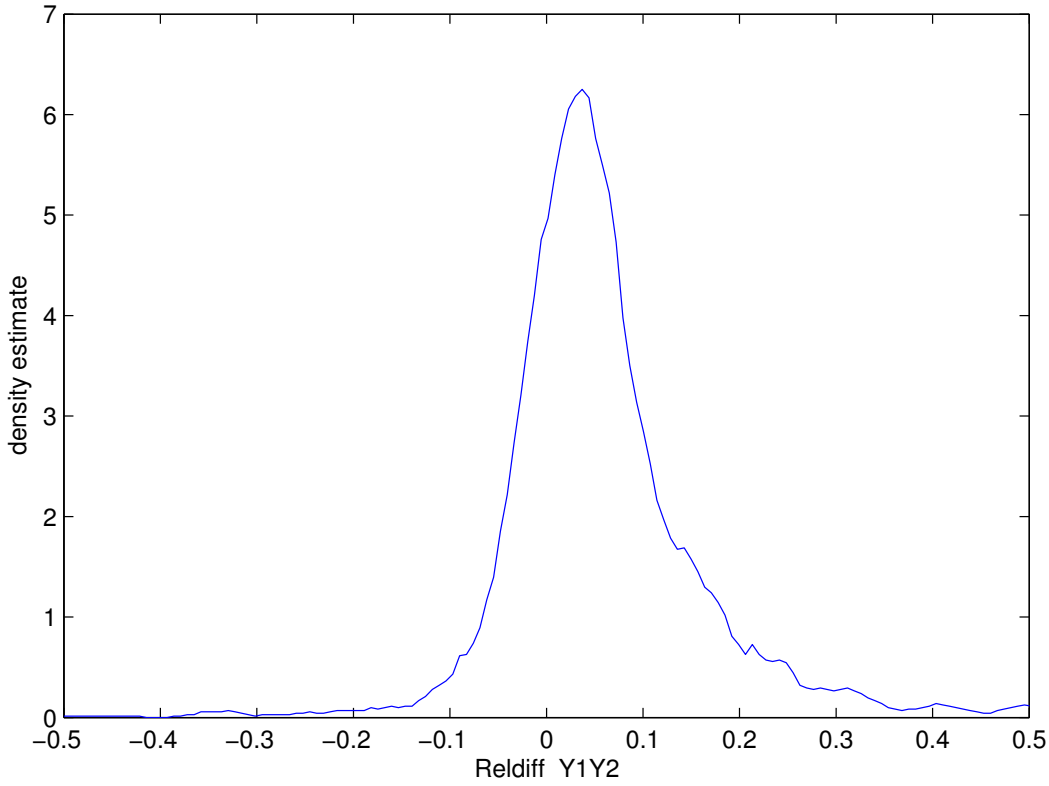


(b) transformed PriCurr

Figure D.6: Feature distribution: closing price at the end of the trading day

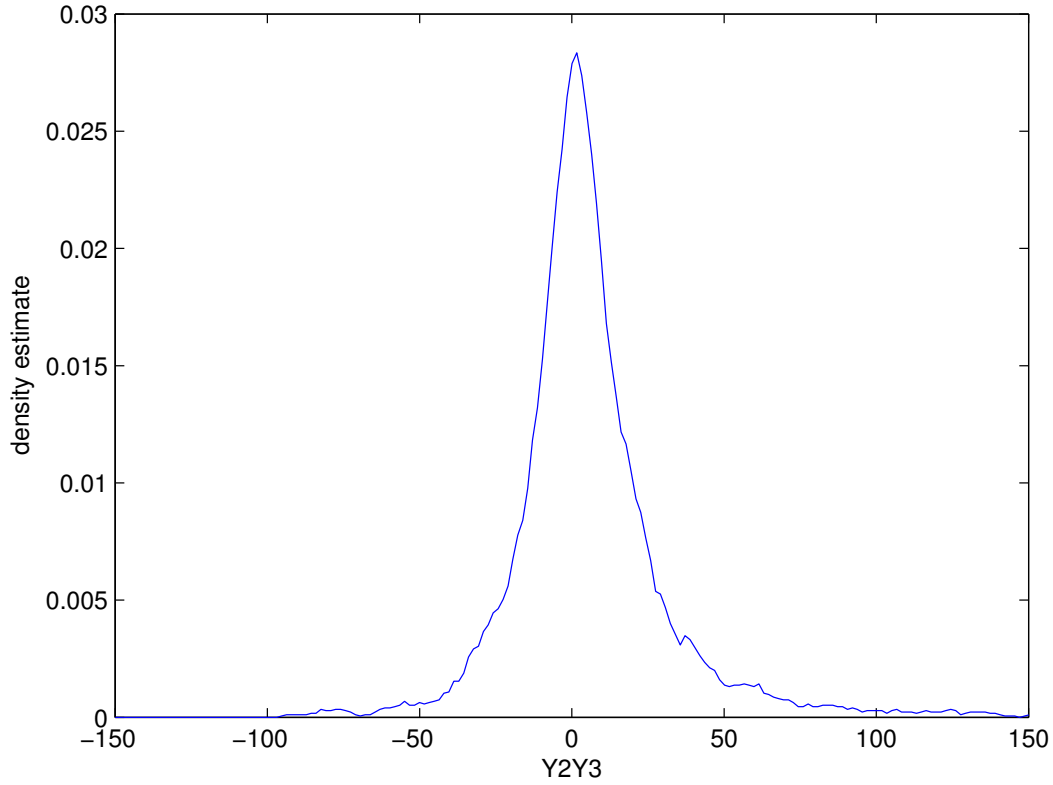


(a) Y1Y2

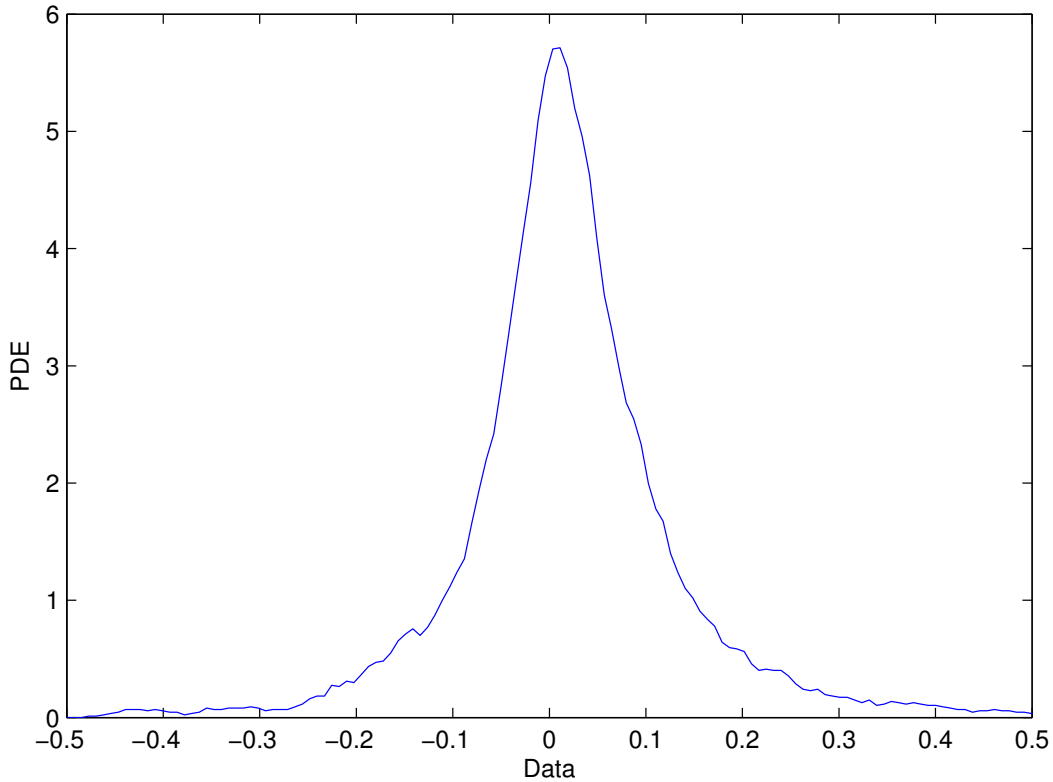


(b) RelDiff Y1Y2

Figure D.7: Feature distribution: annual percental change in a company's revenues

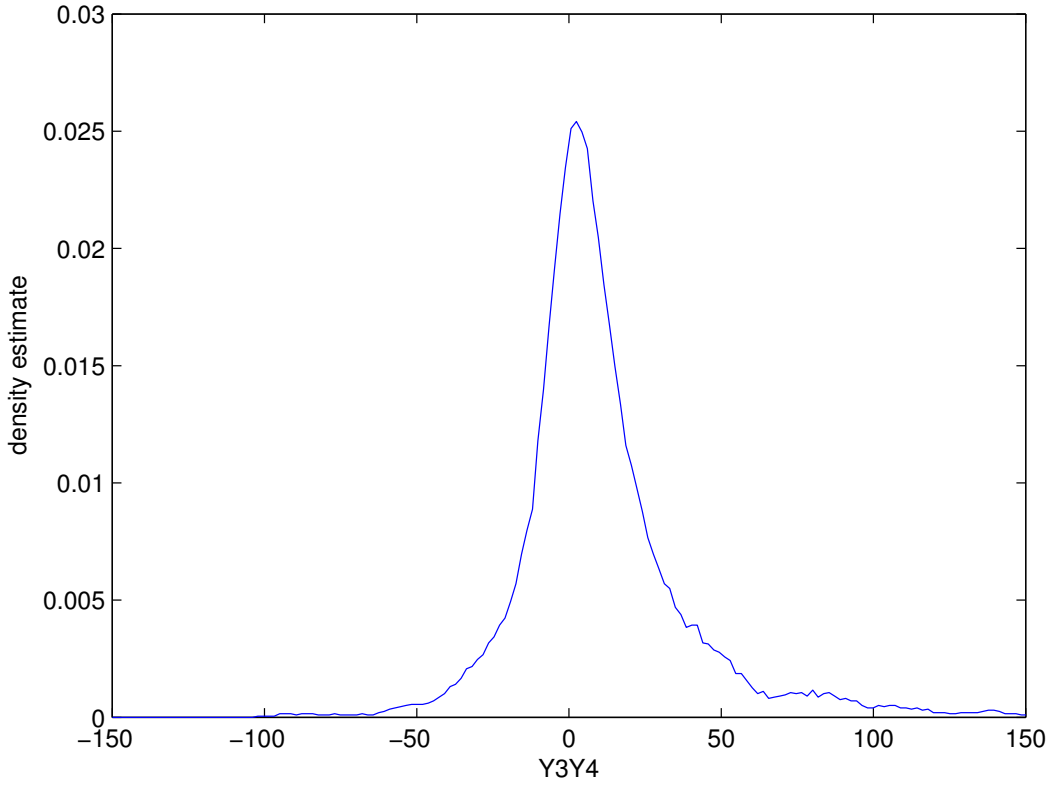


(a) Y2Y3

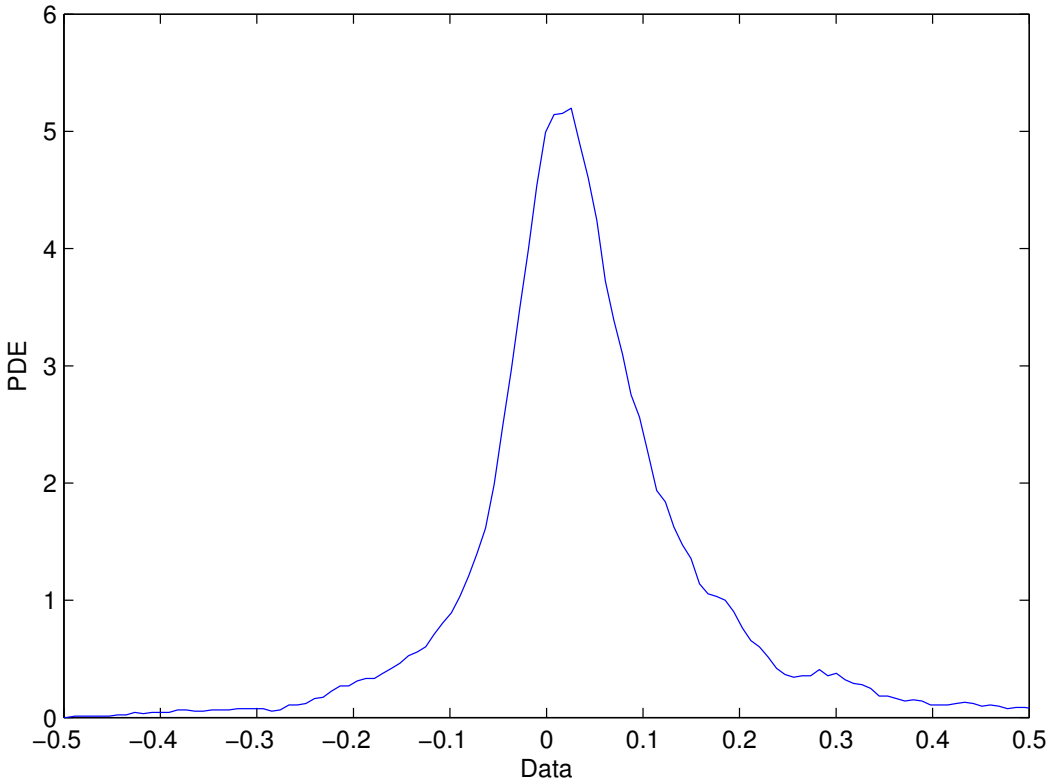


(b) RelDiff Y2Y3

Figure D.8: Feature distribution: prior annual percentage change in a company's revenues



(a) Y3Y4



(b) RelDiff Y3Y4

Figure D.9: Feature distribution: annual percentage change in a company's revenues two years ago

Q1Q5

The percentage year-on-year change in quarterly revenues. The calculation of quarterly year-on-year change is the most recent quarter's revenues minus the year-ago quarter's revenues, divided by the year-ago quarter's revenues; the resulting value is then multiplied by 100. Q1Q5 can be thought of as log-normally distributed because the percental change of revenues is the product of many small independent factors, calculated over periods of a year or more. The relative difference of revenue changes is used for further analysis.

TRRK1

This feature shows how a stock's total returns compare with those of its industry over the given time periods. The stocks in each industry are ranked on a scale from 1 to 100, where 1 represents the highest-returning 1% of stocks in that industry for the given time period. TRRK1 is a rank-based feature that sufficiently follows a uniform distribution. The feature's distribution is therefore not transformed. The range is standardized by linear scaling.

PECur

A stock's current price divided by the company's trailing 12-month earnings per share. Just like returns, these values are produced by multiplication of small, normally distributed changes. The logarithm of PECur is sufficiently normally distributed.

PtBC

The most recent stock price divided by the most recent book value per share. The logarithm of PtBC is sufficiently normally distributed.

PtSC

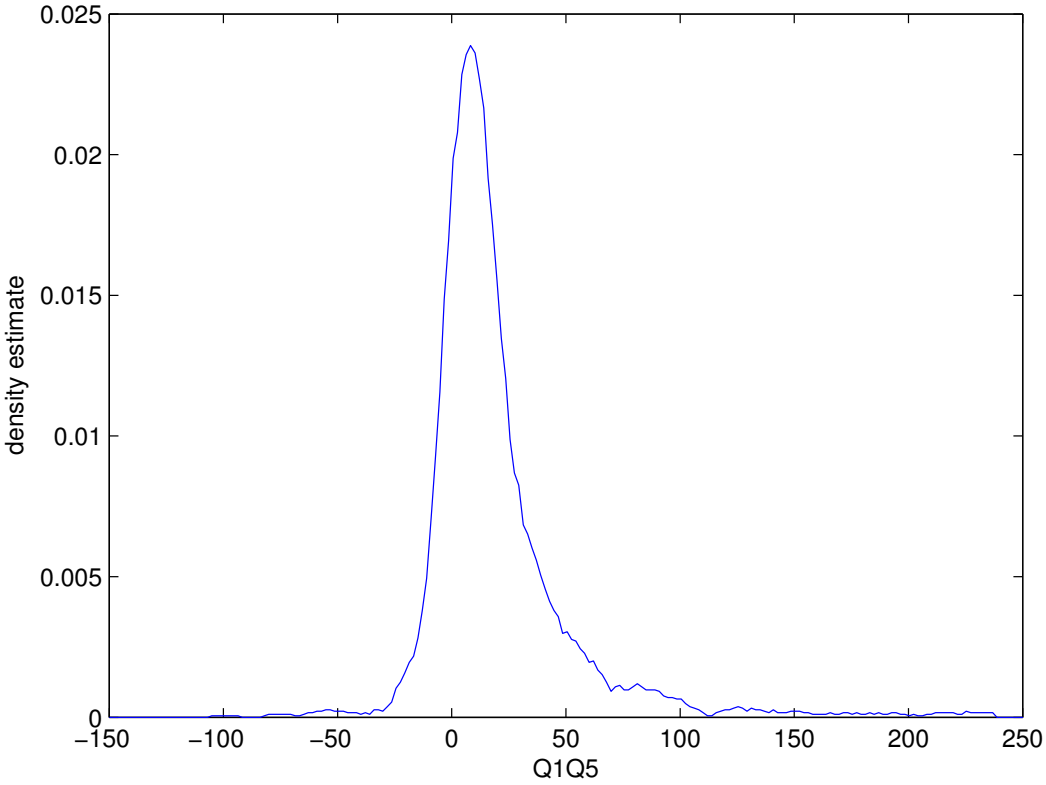
A stock's current price divided by the company's sales per share over the trailing 12 months. The logarithm of PtSC is sufficiently normally distributed.

PtCFC

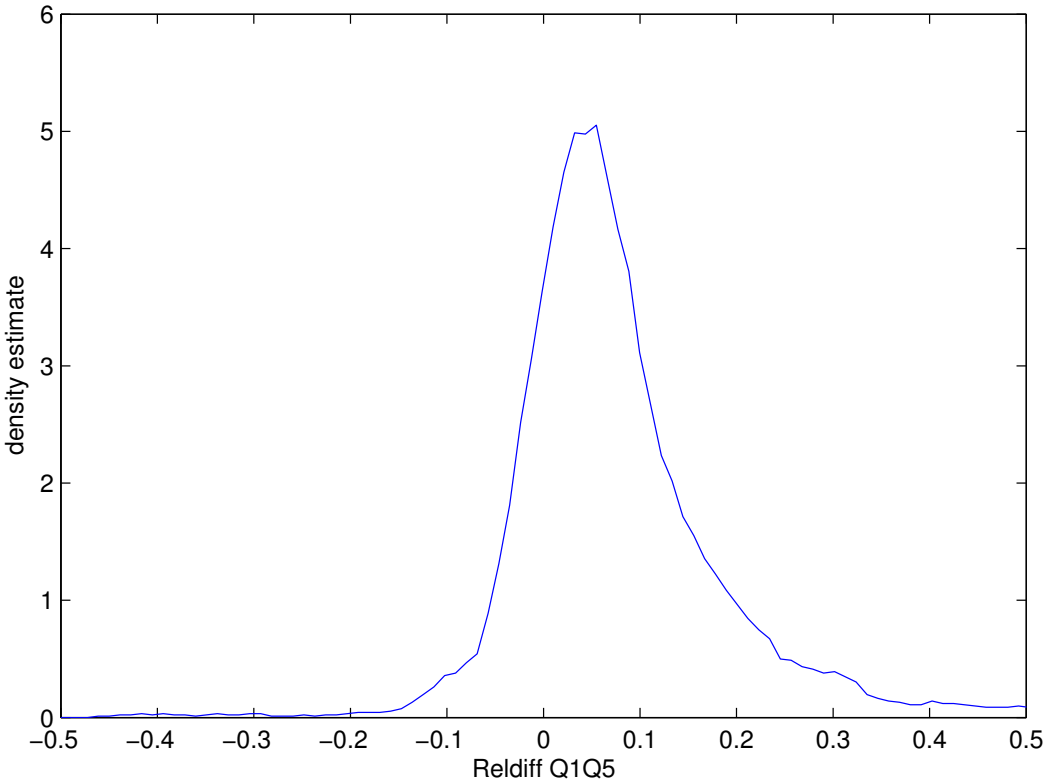
A stock's most recent price divided by the cash-flow per share of the latest fiscal year.
The logarithm of PtSC is sufficiently normally distributed.

RS1Y

The stock's relative strength versus the S&P 500 index in the last year as a return $\in [-100, \infty)$. For further analysis the relative difference of RS1Y values is used in order to limit the values' range.

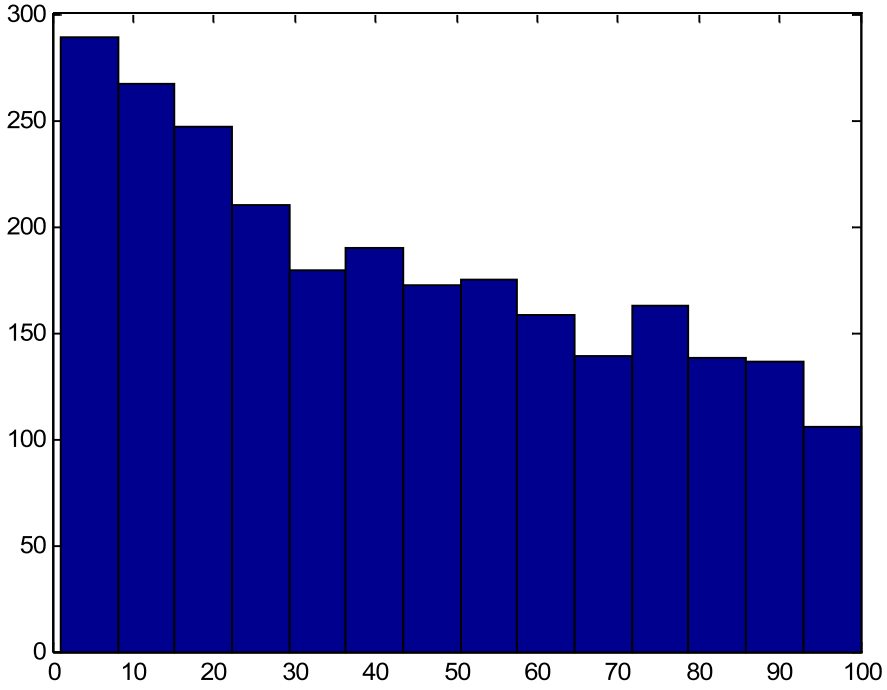


(a) Q1Q5

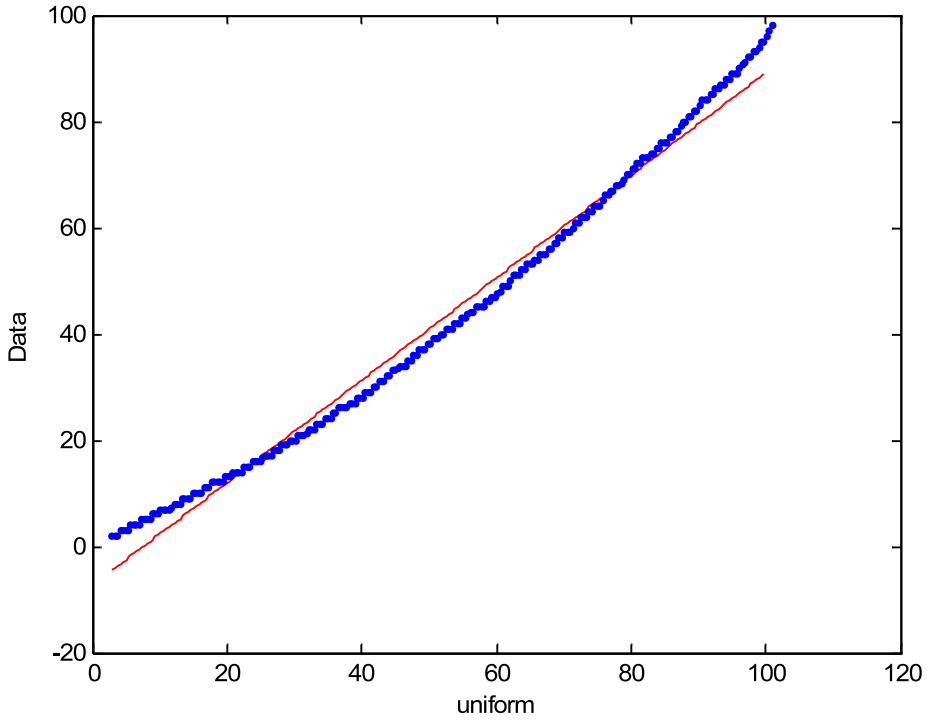


(b) Reldiff Q1Q5

Figure D.10: Feature distribution: percental year-on-year change in quarterly revenues

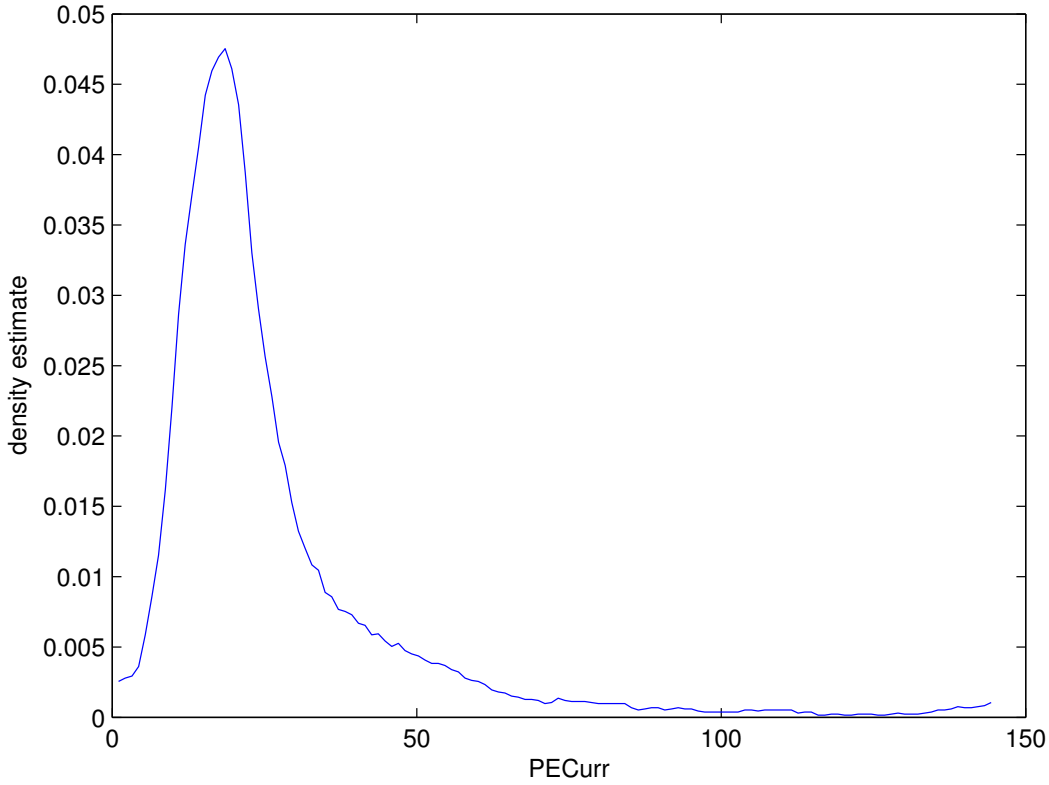


(a) TRRK1

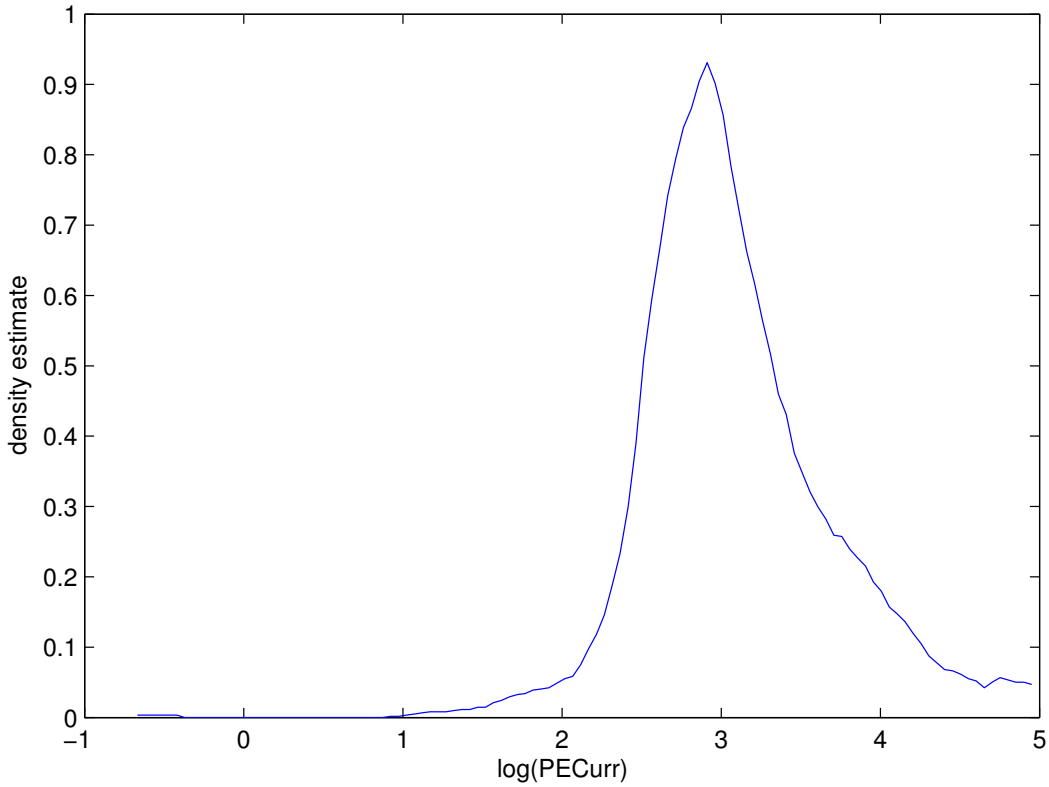


(b) TRRK1 vs. Uniform

Figure D.11: Feature Distribution: total returns in comparison with those of its industry

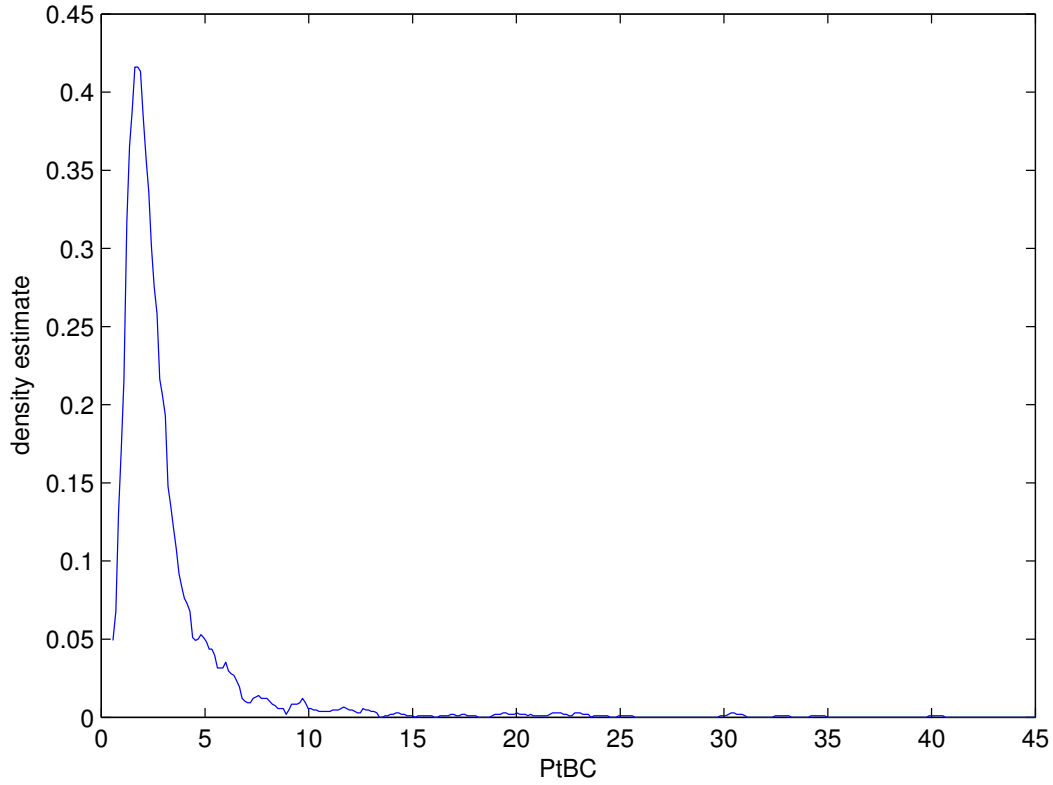


(a) PECur

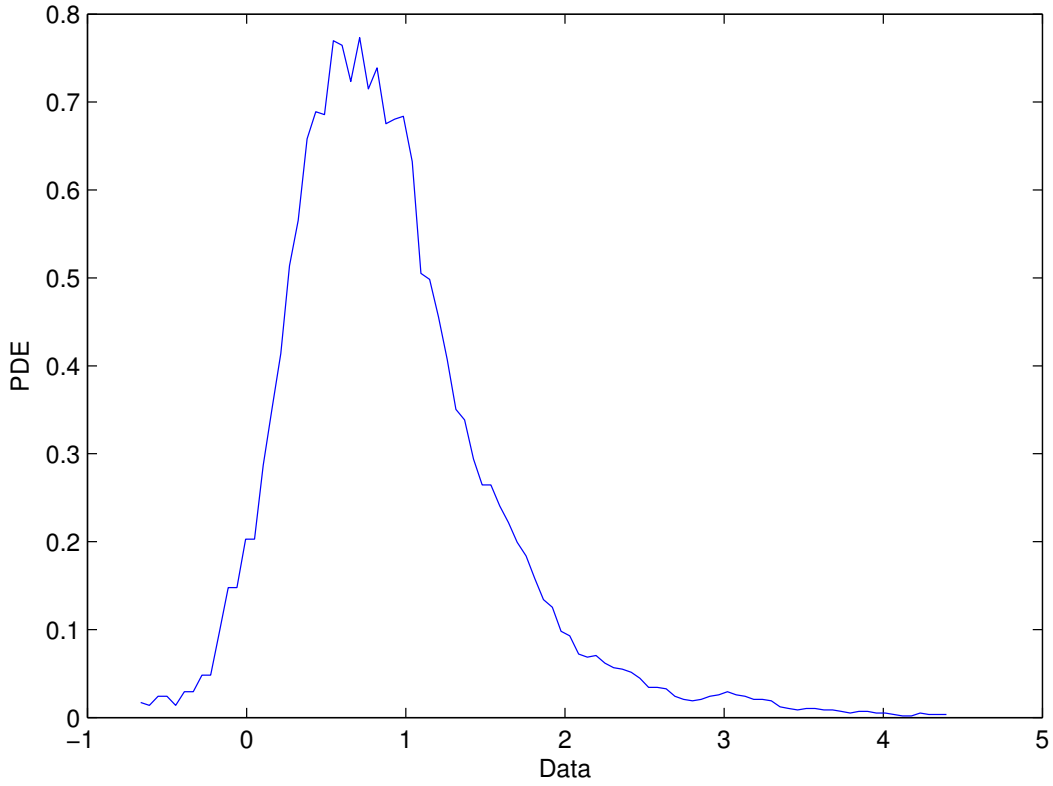


(b) logarithmized PECur

Figure D.12: Feature distribution: current price divided by the company's earnings per share

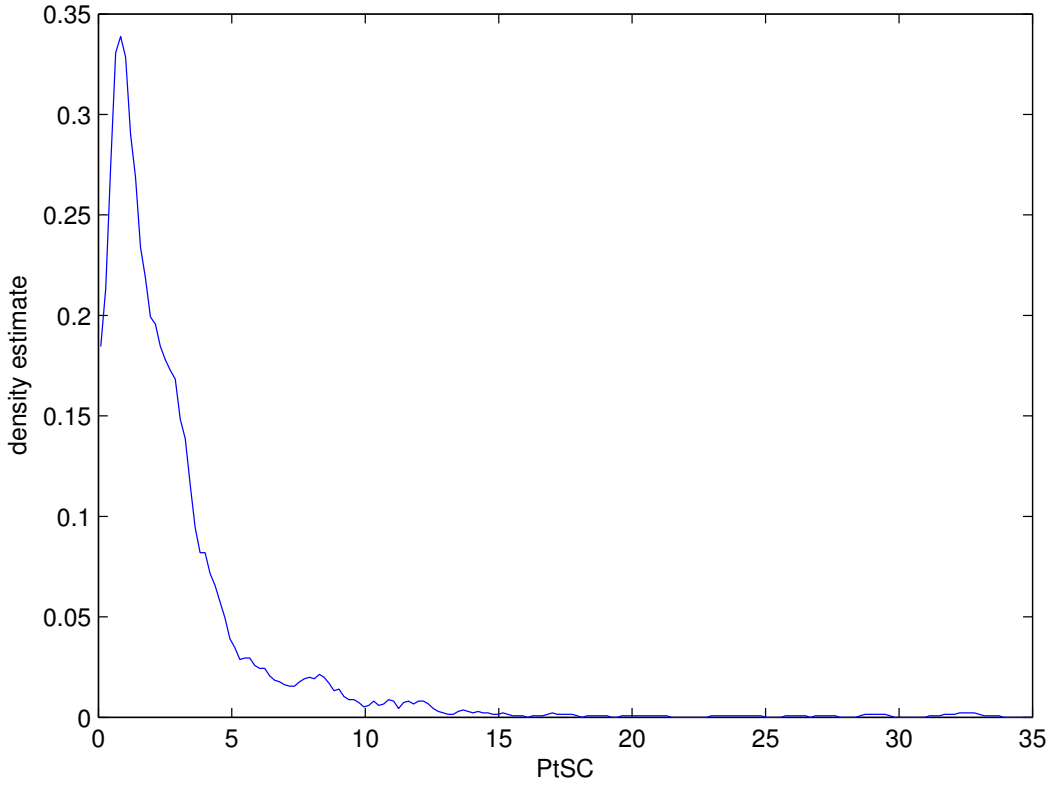


(a) PtBC

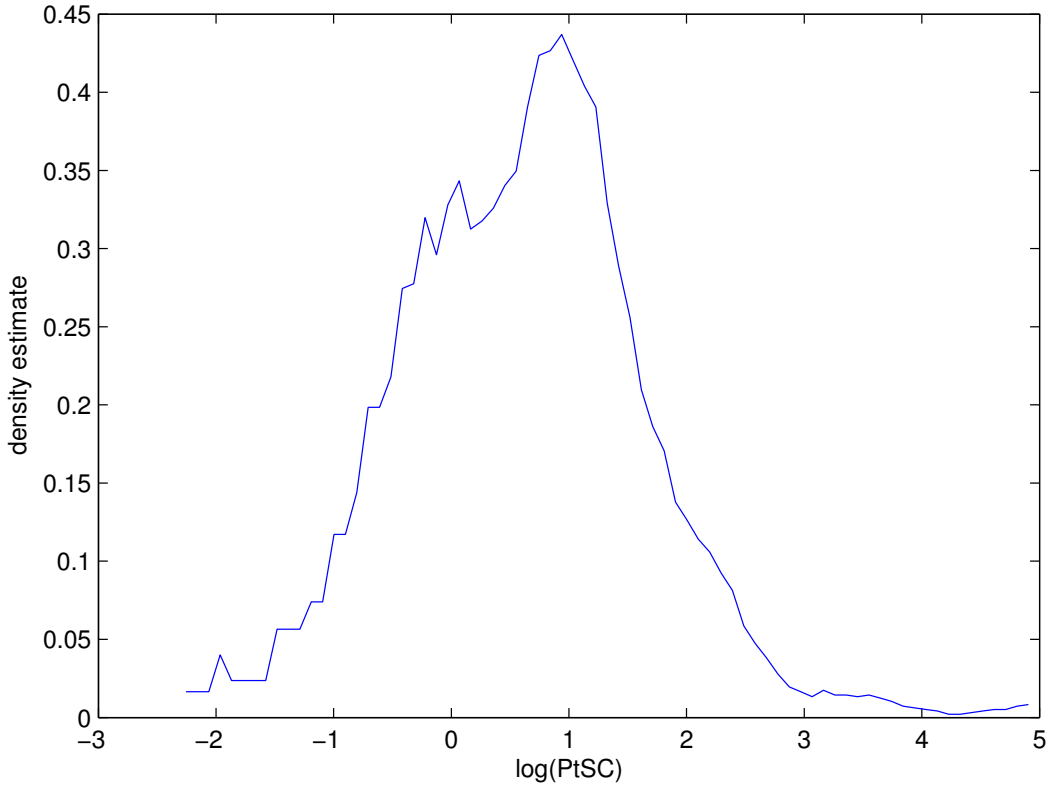


(b) logarithmized PtBC

Feature distribution: recent stock price divided by recent book value per share

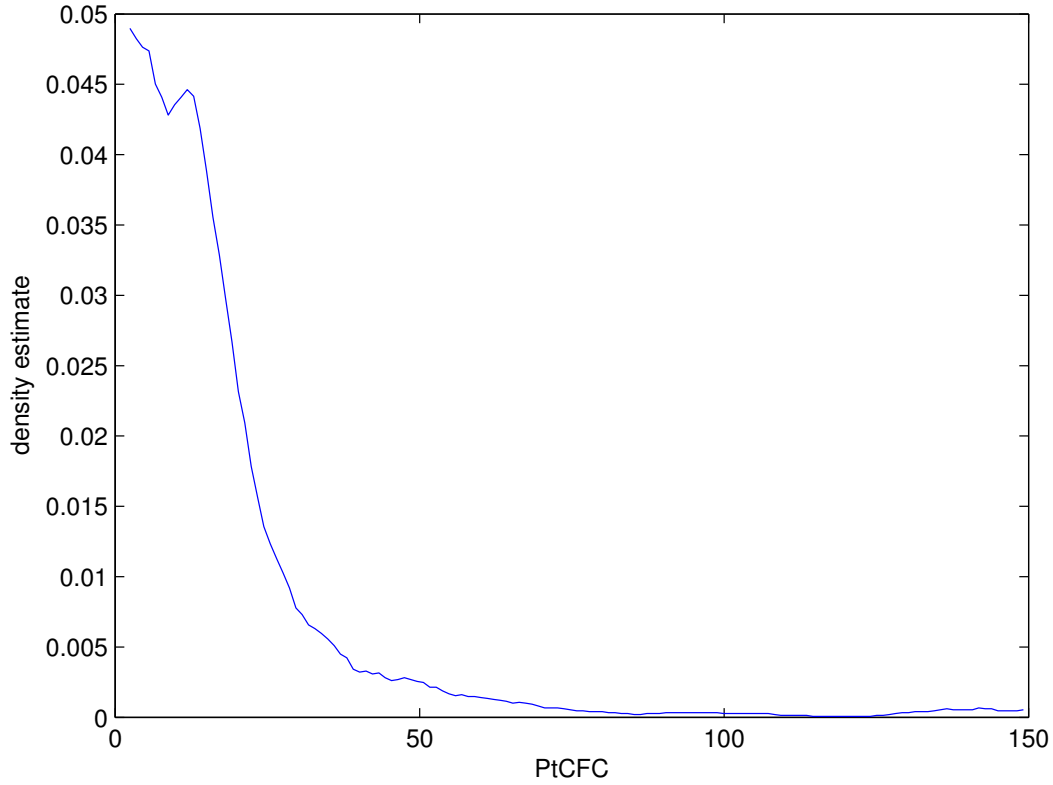


(c) PtSC

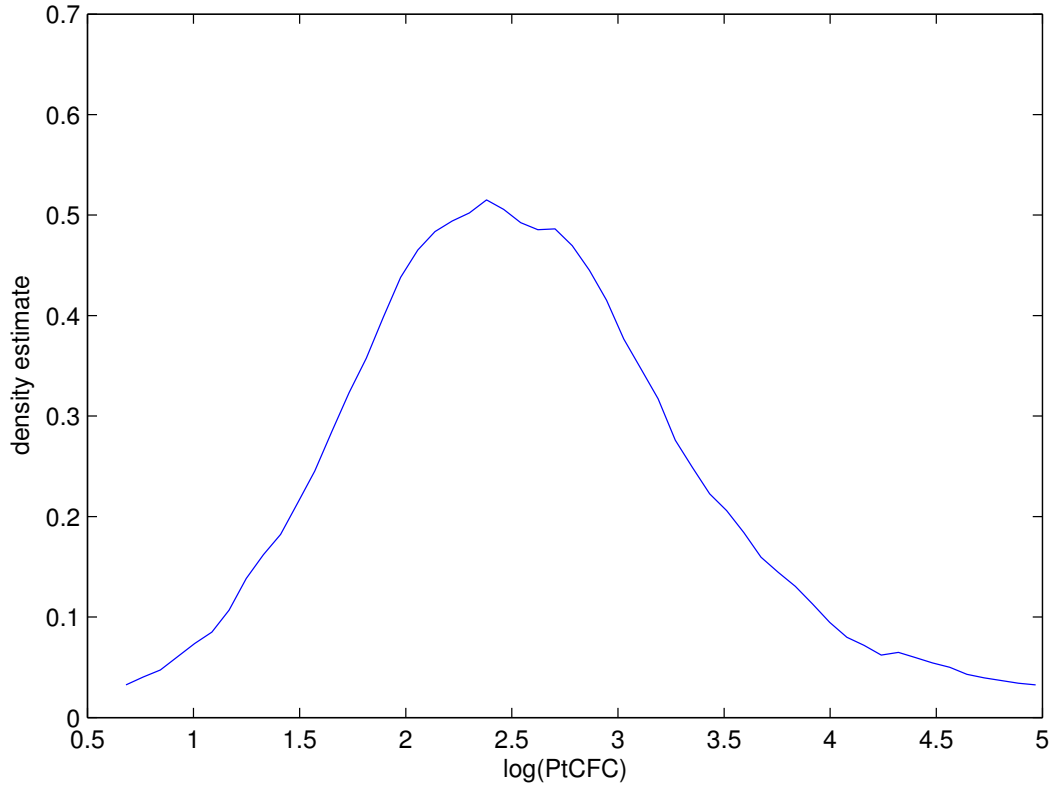


(d) logarithmized PtSC

Figure D.13: Feature distribution: stock's current price divided by the company's sales per share

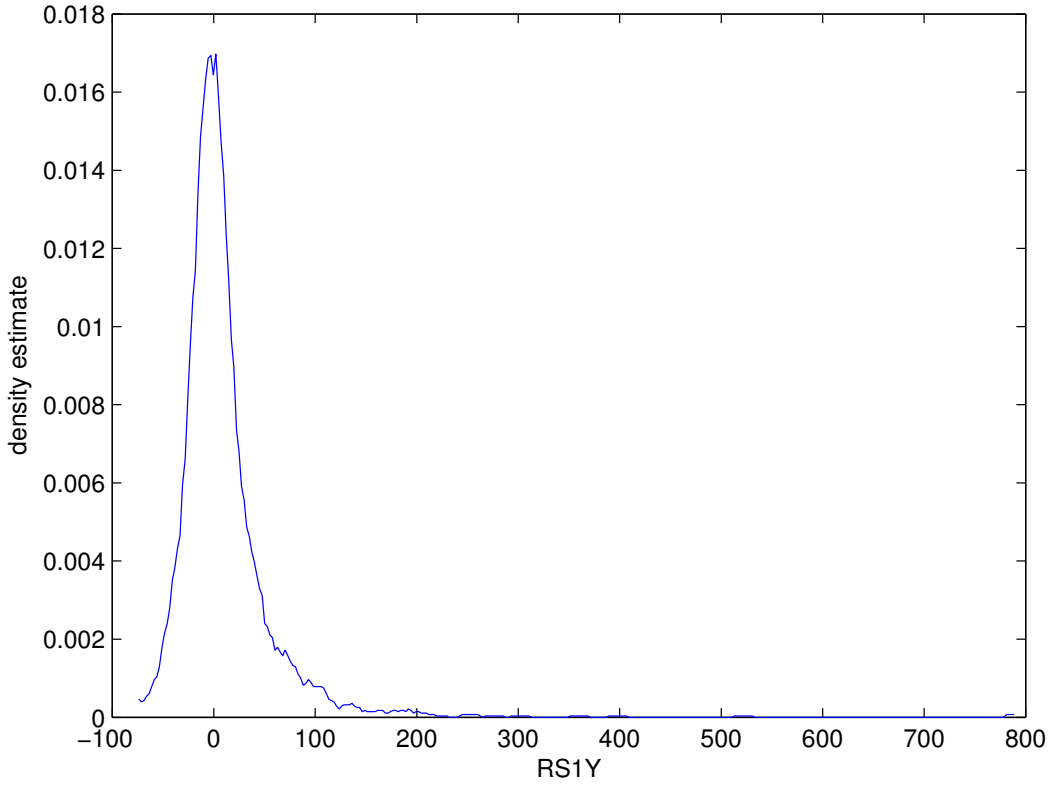


(a) PtCFC

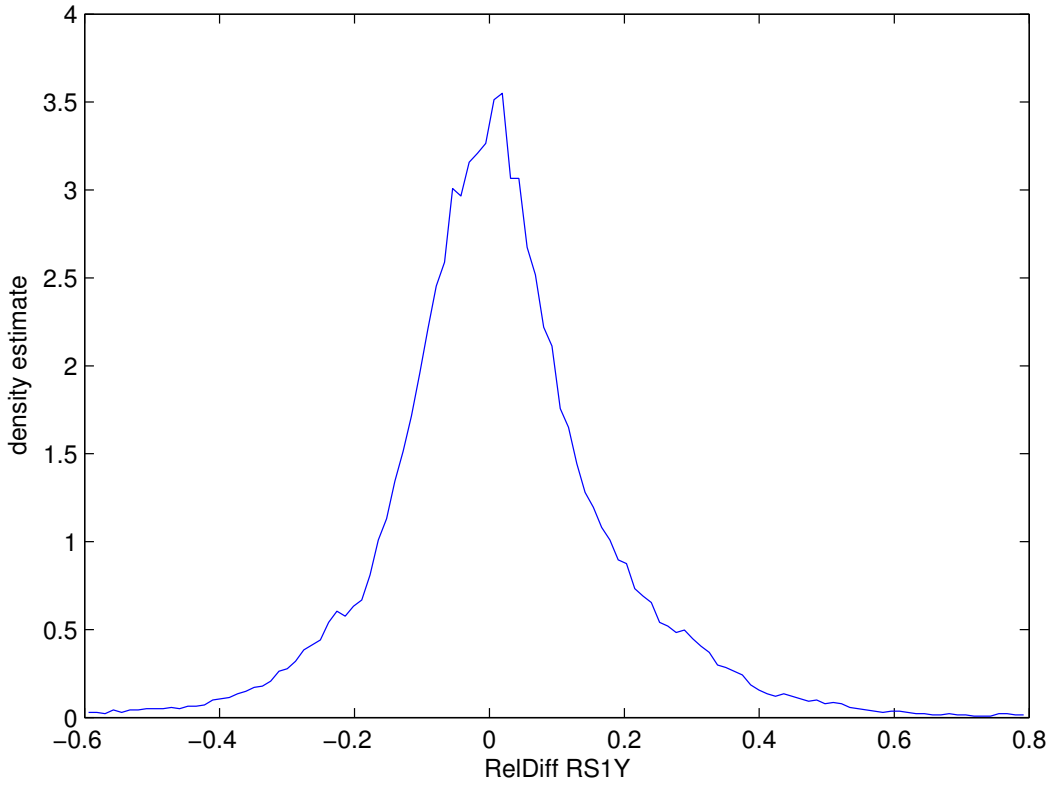


(b) logarithmized PtCFC

Figure D.14: Feature distribution: recent price divided by the cash-flow per share



(a) RS1Y



(b) Reldiff RS1Y

Feature distribution: relative strength versus the S&P 500 index

Bibliography

[AB57] J. Aitchison and J. A. C. Brown. *The Lognormal Distribution: With Special Reference to Its Uses in Economics*. Cambridge University Press, Cambridge, 1957.

[ABB+00] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 25(1):25–29, May 2000.

[ABR64] M. A. Aizerman, E. A. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control,*, volume 25, pages 821–837, 1964.

[ACdV92] S. Aeberhard, D. Coomans, and O. de Vel. Comparison of classifiers in high dimensional settings. Technical Report 92-02, Dept. of Computer Science, James Cook University, North Queensland, 1992.

[AI06] Claus Aranha and Hitoshi Iba. The effect of using evolutionary algorithms on ant clustering techniques. In *Proceedings of the 2006 Asia Pacific Workshop on Genetic Programming (ASPGP06)*, pages 24–34, 2006. Hanoi, Vietnam.

[Alt97] S.F. Altschul. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[Ash47] W. Ross Ashby. Principles of the self-organizing dynamic system. *Journal of General Psychology*, 37:125–128, 1947.

[Bag02] Paul Baggenstoss. Statistical modeling using gaussian mixtures and hmms with matlab. Technical Report TM 03-128, Naval Undersea Warfare Center, USA, 2002.

[Bay63] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.

[BC64] G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964.

[BDT99] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford University Press, Inc., New York, NY, USA, 1999.

[Ber03] Michael Berry. *Survey of Text Mining : Clustering, Classification, and Retrieval*. Springer, September 2003.

[BHV99] H.-U. Bauer, M. Herrmann, and T. Villmann. Neural maps and topographic vector quantization. *Neural Networks*, 12(4-5):659–676, 1999.

[BKK+07] C. Backes, A. Keller, J. Kuentzer, B. Kneissl, N. Comtesse, Y. A. Elnakady, R. Müller, E. Meese, and H. P. Lenhof. Genetrail - advanced gene set enrichment analysis. *Nucleic Acids Res*, 35(Web Server issue), July 2007.

[BM01] Eric Bonabeau and Christopher Meyer. Swarm intelligence: A whole new way to think about business. *Harvard Business Review*, pages 106–114, May 2001.

[BN03] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[BP93] J. C. Bezdek and N. R. Pal. An index of topological preservation and its application to self-organizing feature maps. In *Proc. Int. Joint Conference on Neural Networks*, volume III, pages 2435–2440. IEEE Service Center, Piscataway, NJ, 1993.

[BS04] T. Beissbarth and T. P. Speed. Gostat: find statistically overrepresented gene ontologies within a group of genes. *Bioinformatics*, 20(9):1464–1465, June 2004.

[BW89] G. Beni and J. Wang. Swarm intelligence in cellular robotic systems. In *Proceed. NATO Advanced Workshop on Robots and Biological Systems*, 1989.

[CCM+04] Jill Cheng, Melissa Cline, John Martin, Kulp, and Michael Siani-Rose. A knowledge-based clustering algorithm driven by gene ontology. *Journal of Biopharmaceutical Statistics*, 14(3):687–700, 2004.

[CDF+01] Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulaz, and Eric Bonabeau. *Self-organization in biological systems*, chapter 11. Princeton University Press, Princeton, NJ, USA, 2001.

[CDWG+03] M. Chli, P. De Wilde, J. Goossenaerts, V. Abramov, N. Szirbik, L. Correia, P. Mariano, and R. Ribeiro. Stability of multi-agent systems. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 1, pages 551–556, 2003.

[CLS+04] George Adrian Calin, Chang-Gong Liu, Cinzia Sevignani, Manuela Ferracin, Nadia Felli, Calin Dan Dumitru, Masayoshi Shimizu, Amelia Cimmino, Simona Zupo, Mariella Dono, Marie Dell'Aquila, Hansjuerg Alder, Laura Rassenti, Thomas J. Kipps, Florencia Bullrich, Massimo Negrini, and Carlo M. Croce. MicroRNA profiling reveals distinct signatures in B cell chronic lymphocytic leukemias. *Proceedings of the National Academy of Sciences*, 101(32):11755–11760, 2004.

[CMB+04] Evelyn Camon, Michele Magrane, Daniel Barrell, Vivian Lee, Emily Dimmer, John Maslen, David Binns, Nicola Harte, Rodrigo Lopez, and Rolf Apweiler. The gene ontology annotation (GOA) database: sharing knowledge in Uniprot with gene ontology. *Nucleic Acids Research*, 32:D262, 2004.

[CMB07] J. Chabalier, J. Mosser, and A. Burgun. A transversal approach to predict gene product networks from ontology-based similarity. *BMC Bioinformatic*, 8(235), 2007.

[Cri70] Francis Crick. Central dogma of molecular biology. *Nature*, 227:561–563, August 1970.

[CSC07] Francisco M. Couto, Mario J. Silva, and Pedro M. Coutinho. Measuring semantic similarity between gene ontology terms. *Data & Knowledge Engineering*, 61:137–152, 2007.

[CSTK02] Nello Cristianini, John Shawe-Taylor, and Jaz Kandola. Spectral kernel methods for clustering. In *Advances in Neural Information Processing Systems*. MIT Press, 2002.

[CVRO03] Ernesto Cuadros-Vargas, Roseli Francelin Romero, and Klaus Obermayer. Speeding up algorithms of SOM family for large and high dimensional databases. In *Proc. of Workshop on Self-Organizing Maps (WSOM 2003)*, Kitakyushu, Japan, 2003.

[CW09] Maria Chli and Philippe De Wilde. *Convergence and Knowledge Processing in Multi-Agent Systems*. Springer, 2009.

[DAGP89] Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2):159–168, March 1989.

[Dar59] Charles Darwin. *The Origin of species*. New York, Boston: H.M. Caldwell Co., 1859.

[Deb98] Guido Deboek. *Picking Mutual Funds with Self-Organizing Maps*. Springer-Verlag, 1998.

[DGF+90] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 356–363, Cambridge, MA, USA, 1990. MIT Press.

[DH97] Pierre Demartines and Jeanny Hérault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8:148–154, 1997.

[DHS01] Richard. O. Duda, Peter. E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2001. pp. 517ff.

[Dic45] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via theEM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[DM90] Richard Durbin and Graeme Mitchison. A dimension reduction framework for understanding cortical maps. *Nature*, 343, 1990.

[Dor92] Marco Dorigo. *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.

[Dor01] Marco Dorigo. Swarm intelligence, ant algorithms and ant colony optimization. Talk, 2001. Adaptive Computation and Simulation: Swarm Intelligence.

[Dry78] Hilmar Drygas. Über multidimensionale Skalierung. *Statistical Papers*, 19(1), 1978.

[DU00] Guido J. Deboeck and Alfred Ultsch. Picking stocks with emergent self-organizing value maps. *Neural Networks World*, 10:203–216, 2000.

[Dun74] J. C. Dunn. Well separated clusters and optimal fuzzy-partitions. *Journal of Cybernetics*, 4:95–104, 1974.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[ELL01] Brian Everitt, Sabine Landau, and Morven Leese. *Cluster Analysis*. Arnold Publishers, 2001. Chapter 4.

[EM69] Robert D. Edwards and John Magee. *Technical analysis of stock trends*. J. Magee, Springfield, Mass., 5th edition edition, 1969.

[Fam70] Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2), 1970.

[Fis35] R. A. Fisher. *The Design of Experiments*. Hafner Press, New York, 1935.

[Fis36] Ronald Aylmer Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:197–188, 1936.

[FPSS96] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *Ai Magazine*, 17:37–54, 1996.

[FR88] B. Flury and H. Riedwyl. *Multivariate Statistics, A Practical Approach*. Chapman and Hall, London, 1988.

[FSPB07] Holger Fröhlich, Nora Speer, Annemarie Poustka, and Tim Beißbarth. GOSim - an R-package for computation of information theoretic GO similarities between terms and gene products. *BMC Bioinformatics*, 8, 2007.

[FSSZ06] Holger Fröhlich, Nora Speer, Christian Spieth, and Andreas Zell. Kernel based functional gene grouping. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2006*, pages 3580–3585, 2006.

[FT74] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 23(9):881 – 890, 1974.

[GBO98] Thore Graepel, Matthias Burger, and Klaus Obermayer. Self-organizing maps: Generalizations and new optimization techniques. *Neurocomputing*, 21(1-3):173–190, November 1998.

[GBRV07] Steffen Grossmann, Sebastian Bauer, Peter Robinson, and Martin Vingron. Improved detection of overrepresentation of gene-ontology annotations with parent-child analysis. *Bioinformatics*, 2007.

[GFJ+07] Andrew Grimson, Kyle Kai-How Farh, Wendy K. Johnston, Philip Garrett-Engele, Lee P. Lim, and David P. Bartel. MicroRNA targeting specificity in mammals: Determinants beyond seed pairing. *Mol Cell*, 27(1):91–105, July 2007.

[GFS95] Geoffrey J. Goodhill, Steven Finch, and Terrence J. Sejnowski. A unifying measure for neighbourhood preservation in topographic mappings. In *Proceedings of the 2nd Joint Symposium on Neural Computation*, pages 191–202, University of California, San Diego and California Institute of Technology, 5, Institute for Neural Computation, La Jolla, CA, 1995.

[GJGvD+06] Sam Griffiths-Jones, Russell J. Grocock, Stijn van Dongen, Alex Bateman, and Anton J. Enright. mirbase: microRNA sequences, targets and gene nomenclature. *Nucleic acids research*, 34(Database issue):D140–144, January 2006.

[GKCS08] Hamid Ghous, Paul J. Kennedy, Daniel R. Catchpoole, and Simeon J. Simoff. Kernel-based visualisation of genes with the gene ontology. In *Proc. Seventh Australasian Data Mining Conference*, pages 133–140, 2008.

[Goe06] Andreas Goebels. *Agent Coordination Mechanisms for Solving a Partitioning Task*. PhD thesis, University of Paderborn, 2006.

[Gra59] Plerre-P Grassé. La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6(1):41–80, March 1959.

[Hat08] John N. Hatzopoulos. *Topographic Mapping: Covering the Wider Field of Geospatial Information & Technology*. Universal Publishers, Boca Raton, Florida, USA, 2008.

[HH07] Alexander Hasenfuss and Barbara Hammer. Relational topographic maps. In *Advances in Intelligent Data Analysis VII*, LNCS, pages 93–105. Springer, 2007.

[HKD05] Julia Handl, Joshua Knowles, and Marco Dorigo. Ant-based clustering and topographic mapping. *Artificial Life*, 12:35–61, 2005.

[HR02] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *Neural Information Processing Systems (NIPS 2002)*, volume 15, pages 857–864, 2002.

[HTH+05] Lin He, Michael M. Thomson, Michael T. Hemann, Eva Hernando-Monge, David Mu, Summer Goodson, Scott Powers, Carlos Cordon-Cardo, Scott W. Lowe, Gregory J. Hannon, and Scott M. Hammond. A microRNA polycistron as a potential human oncogene. *Nature*, 435(7043):828–833, 2005.

[HU08] Lutz Herrmann and Alfred Ultsch. The architecture of ant-based clustering to improve topographic mapping. In Marco Dorigo, Mauro Birattari, Christian Blum, Maurice Clerc, Thomas Stützle, and Alan Winfield, editors, *Ant Colony Optimization and Swarm Intelligence - Proceedings 6th Int. Conf. (ANTS2008)*, pages 379–386, Brussels, Belgium, 2008.

[Jac01] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[JC97] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)*, pages 9008+, September 1997.

[JHK98] Bärbel Elpelt Joachim Hartung and Karl-Heinz Klösener. *Statistik*. R. Oldenbourg, Munich, 1998.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.

[Kas97] S. Kaski. *Data exploration using self-organizing maps*. PhD thesis, Helsinki University of Technology, 1997.

[Kas99] Samuel Kaski. Fast winner search for SOM-based monitoring and retrieval of high-dimensional data. In *Proc. of Ninth International Conference on Artificial Neural Networks*, volume 2, pages 940–945, 1999.

[KE95] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. of the IEEE Int. Conf. on Neural Networks*, pages 1942–1948. IEEE Press, 1995.

[Kei02] Daniel A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visiual Computer Graphics*, 8(1):1–8, 2002.

[Kir78] Arnold Kirsch. Bemerkung zu H. Drygas, über multidimensionale Skalierung. *Statistical Papers*, 19(3), 1978.

[KL51] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[KNO+03] Samuel Kaski, Janne Nikkilä, Merja Oja, Jarkko Venna, Petri Törönen, , and Eero Castren. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4(48), 2003.

[Koh89] T. Kohonen. *Self-organization and associative memory: 3rd edition*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[Koh97] T. Kohonen. *Self-organizing Maps: 2nd edition*. Springer-Verlag, Berlin, Germany, 1997.

[Koh99] T. Kohonen. Speedup of SOM computation. Technical report, Helsinki University of Technology, 1999.

[KR90] L. Kaufman and P. J. Rousseeuw. *Finding groups in data. an introduction to cluster analysis*. Wiley Series in Probability and Mathematical Statistics.˜Applied Probability and Statistics, New York: Wiley, 1990, 1990.

[Kru64] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, March 1964.

[KS02] Teuvo Kohonen and Panu Somervuo. How to make large self-organizing maps for nonvectorial data. *Neural Networks*, 15(8-9):945–952, 2002.

[Leh98] E. L. Lehmann. *Testing Statistical Hypotheses*. Springer, 1998. 2nd edition.

[Lev66] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[LF94] Erik D. Lumer and Baldo Faieta. Diversity and adaptation in populations of clustering ants. In *SAB94: Proceedings of the third international conference on Simulation of adaptive behavior : from animals to animats 3*, pages 501–508, Cambridge, MA, USA, 1994. MIT Press.

[LFA93] R. C. Lee, R. L. Feinbaum, and V. Ambros. The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *Cell*, 75(5):843–854, December 1993.

[LGM+05] Jun Lu, Gad Getz, Eric A. Miska, Ezequiel Alvarez-Saavedra, Justin Lamb, David Peck, Alejandro Sweet-Cordero, Benjamin L. Ebert, Raymond H. Mak, Adolfo A. Ferrando, James R. Downing, Tyler Jacks, Robert R. Horvitz, and Todd R. Golub. MicroRNA expression profiles classify human cancers. *Nature*, 435(7043):834–838, 2005.

[Lin98] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.

[LKK04] Krista Lagus, Samuel Kaski, and Teuvo Kohonen. Mining massive document collections by the websom method. *Information Sciences*, 163:135–156, 2004.

[Llo03] S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, January 2003.

[LSA01] E. Limpert, W. A. Stahel, and M. Abbt. Log-normal distributions across the sciences: Keys and clues. *BioScience*, 51(5):341–352, May 2001.

[LSBG03a] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–83, 2003.

[LSBG03b] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Semantic similarity measures as tools for exploring the gene ontology. *Pacific Symposium on Biocomputation*, pages 601–612, 2003.

[LT93] Baruch Lev and S. Ramu Thiagarajan. Fundamental information analysis. *Journal of Accounting Research*, 31(2):190–215, 1993.

[Mac67] J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[MC88] Glenn W. Milligan and Martha C. Cooper. A study of standardization of variables in cluster analysis. *Journal of Classification*, 5(2):181–204, 1988.

[Mit97] T. Mitchell. *Machine Learning*. McGraw-Hill Education (ISE Editions), 1st edition, October 1997.

[MP08] Meeta Mistry and Paul Pavlidis. Gene ontology term overlap as a measure of gene functional similarity. *BMC Bioinformatics*, 9, 2008.

[MW47] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60, 1947.

[NMU06] Mario Nöcker, Fabian Mörchen, and Alfred Ultsch. An algorithm for fast and reliable ESOM learning. In Michel Verleysen, editor, *Proceedings of 14th European Symposium on Artificial Neural Networks (ESANN)*, pages 131–136, Bruges, Belgium, 2006.

[NVK07] K. Nybo, J. Venna, and S. Kaski. The self-organizing map as a visual neighbor retrieval method. In *Proc. 6th International Workshop on Self-Organizing Maps (WSOM), 2007*. University Library of Bielefeld, 2007.

[OLH08] Kristian Ovaska, Marko Laakso, and Sampsa Hautaniemi. Fast gene ontology based clustering for microarray experiments. *BioData Mining*, 1(1):11+, November 2008.

[O'N95] W. O'Neil. *How to make money in stocks: A winning system in good times and bad*. McGraw Hill, New York, 1995.

[OWZ+05] Kathryn A. O'Donnell, Erik A. Wentzel, Karen I. Zeller, Chi V. Dang, and Joshua T. Mendell. c-Myc-regulated microRNAs modulate E2F1 expression. *Nature*, 435(7043):839–843, 2005.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[PFB+08] Catia Pesquita, Daniel Faria, Hugo Bastos, Antonio Ferreira, Andre Falcao, and Francisco Couto. Metrics for go based protein semantic similarity: a systematic evaluation. *BMC Bioinformatics*, 9(S4), 2008.

[PKM06] Mihail Popescu, James Keller, and Joyce Mitchell. Fuzzy measures on the gene ontology for gene product similarity. *IEEE Transactions on Computational Biology and Bioinformatics*, 3(3), 2006.

[Pri57] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[PRM02] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 2002.

[Pyl99] Dorian Pyle. *Data Preparation for Data Mining (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, March 1999.

[Qua01] John Quackenbush. Computational genetics: Computational analysis of microarray data. *Nature Reviews Genetics*, 2:418–427, 2001.

[RA04] Vitorino Ramos and Ajith Abraham. Evolving a stigmergic self-organized datamining. In *International Conference on Web Based Communities*, 2004.

[Ran71] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[Res95] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[RFR06] Vitorino Ramos, Carlos Fernandes, and Agostinho C. Rosa. Social Cognitive Maps, Swarm Collective Perception and Distributed Search on Dynamic Landscapes. *Brains, Minds & Media, Journal of New Media in Neural and Cognitive Science, NRW, Germany*, 2006.

[RK91] Sung-Woong Ra and J.-K. Kim. Fast weight-ordered search algorithm for image vector quantization. *IEEE Electronic Letters*, 27:2081–2083, 1991.

[RM01] Jos Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2001.

[RMS92] Helge Ritter, Thomas Martinetz, and Klaus Schulten. *Neural computation and self-organizing maps: an introduction*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1992.

[Ros07] Fabrice Rossi. Model collisions in the dissimilarity SOM. In *Proceedings of XVth European Symposium on Artificial Neural Networks (ESANN 2007)*, pages 25–30, Bruges, Belgium, 2007.

[RS00] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[Ruv01] Gary Ruvkun. Molecular biology: Glimpses of a tiny RNA world. *Science*, 294(5543):797–799, October 2001.

[Sch69] Thomas C. Schelling. Models of segregation. *The American Economic Review*, 59(2):488–493, 1969.

[Sch80] Friedrich Schmid. Über ein Problem der mehrdimensionalen Skalierunq. *Statistical Papers*, 21(2), 1980.

[SFSZ05] Nora Speer, Holger Fröhlich, Christian Spieth, and Andreas Zell. Functional grouping of genes using spectral clustering and gene ontology. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 298–303. IEEE Press, 2005.

[SMC08] Stefanie Sassen, Eric A. Miska, and Carlos Caldas. MicroRNA: implications for cancer. *Virchows Archiv : an international journal of pathology*, 452(1):1–10, January 2008.

[Smi48] N. Smirnov. Table for estimating the goodness of fit of empirical distributions. *The Annals of Mathematical Statistics*, 19(2):279–281, 1948.

[SSDB95] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, October 1995.

[SSM98] B. Schölkopf, A. J. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[SSP+05] Jose L. Sevilla, Victor Segura, Adam Podhorski, Elizabeth Guruceaga, and Jose M. Mato. Correlation between gene expression and go semantic similarity. *IEEE Transactions on Computational Biology and Bioinformatics*, 2(4), 2005.

[SSZ05] Nora Speer, Christian Spieth, and Andreas Zell. Spectral clustering gene ontology terms to group genes by function. In *5th Workshop on Algorithms in Bioinformatics (WABI 2005)*, pages 1–12, 2005.

[Stu08] Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908.

[Sve98] Johan Fredrik Markus Svensen. *The Generative Topographic Mapping*. PhD thesis, Ashton University, 1998.

[TdSL00] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[Tho98] Michael C. Thomsett. *Mastering Fundamental Analysis*. Dearborn Financial Publishing, Inc., Chicago, 1998.

[Tor52] Warren Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, December 1952.

[TTT06] S.C. Tan, K.M. Ting, and S.W. Teng. Reproducing the results of ant-based clustering without using ants. In *Proceedings IEEE Congress on Evolutionary Computation (CEC 2006)*, Vancouver, BC, Canada, July 16-21, 2006. IEEE Press.

[UGKL93] Alfred Ultsch, Gabriela Guimaraes, Dieter Korus, and Heng Li. Knowledge extraction from artificial neural networks and applications. In *Proceedings of Transputer-Anwender-Treffen / World-Transputer-Congress (TAT/WTC)*, Aachen, 1993.

[UH05] Alfred Ultsch and Lutz Herrmann. The architecture of emergent self-organizing maps to reduce projection errors. In *European Symposium on Artificial Neural Networks*, pages 1–6, 2005.

[Ult] Alfred Ultsch. Fundamental clustering problem suite. http://www.uni-marburg.de/fb12/datenbionik.

[Ult99] A. Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In *Kohonen Maps*, pages 33–46. Elsevier, 1999.

[Ult00a] Alfred Ultsch. Clustering with databots. In *Proc. Int. Conf. Advances in Intelligent Systems Theory and Applications (AISTA)*, 2000. Canbera.

[Ult00b] Alfred Ultsch. The neuronal data mine. In *Proceedings 2nd Int. ICSC Symposium on Neural Computation*, Berlin, 2000.

[Ult03] Alfred Ultsch. Maps for the visualization of high-dimensional data spaces. In *Proceedings Workshop on Self-Organizing Maps (WSOM 2003)*, pages 225–230, Kyushu, Japan, 2003.

[Ult08] Alfred Ultsch. Is log ratio a good value for measuring return in stock investments? In *Proc. of the 32nd Annual Conference of the German Classification Society*. Springer, 2008.

[UM06] Alfred Ultsch and Fabian Mörchen. U-maps: topographic visualization techniques for projections of high dimensional data. In *Proceedings 30th Annual Conference of the German Classification Society*, 2006. Berlin.

[US90] Alfred Ultsch and H.P. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings Intern. Neural Networks*, Paris, 1990. Kluwer Academic Press.

[UV94] A. Ultsch and C. Vetter. Self-organizing-feature-maps versus statistical clustering methods: A benchmark. Technical Report 9, Dept. of Mathematics and Computer Science, University of Marburg, Germany, 1994.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, November 2008.

[Ven07] Jarkko Venna. *Dimensionality Reduction for Visual Exploration of Structures*. PhD thesis, Helsinki University of Technology, 2007.

[Ves02] Juha Vesanto. *Data Exploration Process Based on the Self-Organizing Map*. PhD thesis, Helsinki University of Technology, 2002.

[VHAP00] Juha Vesanto, Johan Himberg, Esa Alhoniemi, and Juha Parhankangas. Self-organizing map in matlab: the SOM toolbox. In *Proceedings of the Matlab DSP Conference*, pages 35–40, 2000. Espoo, Finland.

[VK06] Dejan Vinkovic and Alan Kirman. A physical analogue of the schelling model. *Proceedings of the National Academy of Sciences of the United States of America*, 103(51):19261–19265, 2006.

[Vor07] Georgy Voronoi. Nouvelles applications des parametres continus à la theorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik*, 133:97–178, 1907.

[vR79] C. J. van Rijsbergen. *Information retrieval*. Butterworths, London, 2nd edition, 1979.

[VS02] Juha Vesanto and Mika Sulkava. Distance matrix based clustering of the self-organizing map. In *ICANN '02: Proceedings of the International Conference on Artificial Neural Networks*, pages 951–956, London, UK, 2002. Springer-Verlag.

[War63] Joe H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[WHKK07] Nils Weskamp, Eyke Hüllermeier, Daniel Kuhn, and Gerhard Klebe. Multiple graph alignment for the structural analysis of protein active sites. *IEEE Transactions on Computational Biology and Bioinformatics*, 4(2), 2007.

[Wis08] Axel Wismüller. *Exploratory Morphogenesis - A Novel Computational Framework for Self-Organization*. PhD thesis, Technical University of Munich, 2008.

[WNL99] Philippe De Wilde, Hyacinth S. Nwana, and Lyndon C. Lee. Stability, fairness and scalability of multi-agent systems. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 3(2):84–91, 1999.

[Zre93] Stephane Zrehen. Analyzing kohonen maps with geometry. In St. Gielen and B. Kappen, editors, *Proceedings of the International Conference an Artificial Neural Networks*. Springer, 1993.

Glossary

Agent

Agents are software entities located in a low-dimensional output space. Each agent is identified with an object from a data space of interest. Agents move on their own based on other agents' location in order to position themselfs nearby similar agents and, respectively, data objects. 35, 46, 54, 65

Class

For a given set of data objects $X = \{x_1, ..., x_n\}$ any subset $C \subset X$ is simply refered to as class. It is assumed that X decomposes into several disjoint, exhaustive and non-empty classes. A class is a subset of objects sharing a certain property, e.g. sets of genes responsible for certain tasks in molecular biology. 21, 130

Cluster

Clusters are a refinement of the concept of classes. Classes whose elements share a certain geometric property are refered to as clusters. Typically, cluster can be thought of as dense point clouds in high-dimensional vectorial spaces. 24

Cluster Analysis

Cluster analysis refers to the process of finding intrinsic subsets of similar objects in a given set $x_1, ..., x_n$ of data objects. By doing so the data set is segmented into several clusters. 24, 77, 118, 132

Cohesiveness

A mapping $m : \{x_1, ...x_n\} \to \{y_1, ..., y_n\}$ is refered to as cohesive with respect to class $C \subset \{x_1, ...x_n\}$ if the intersection of $C \times C$ and the Delaunay graph of images $\{y_1, ..., y_n\}$ is connected. Thus Voronoi cells of class C are adjacent, which is a major requirement for the U-Matrix visualization to correctly depict the inner class dissimilarities of class C in low-dimensional output spaces. For non-cohesive mappings inter class dissimilarities are falsely depicted between images of the same class C. 60, 96

Data

Data refers to a set of objects $\{x_1, ..., x_n\}$. Each object represents a unique entity from the given domain of interest, e.g. in molecular biology data deals

with genes that are to be analyzed in terms of interactions and behaviour. 19, 21

Data Object

A data object is the most basic element of any data space. The comparision of two objects is evaluated by means of a (dis)similarity function d. 21

Delaunay graph

The Delaunay graph (V, E) of a given set $V = \{y_1, ..., y_n\}$ contains edges $(y_i, y_j) \in E$ iff the Voronoi cells $V(i), V(j)$ are adjacent. Delaunay graphs are useful formalisms for exploration of neighbourhood relations. 23, 60, 96

Density

The probability of a random sample falling within a given set is given by the integral of its probability density function over the set. Densities are usually associated with continuous univariate distributions. 25, 58, 75, 77

Dissimilarity

A dissimilarity function d is a two-ary, symmetric, non-negative real-valued operation that compares data objects. Values of zero indicate identical objects, whereas large values indicate greatly differing objects. As a special case, metric distance functions furthermore fulfil the triangle inequality. 20, 115

Graph

A pair of sets (V, E) representing nodes (vertices) and edges. The set V of nodes represents the underlying data objects. The set of edges $E \subset V \times V$ represents a connection or similarity relation among nodes. 22

Topographic Mapping

A mapping $m : \{x_1, ...x_n\} \rightarrow \{y_1, ..., y_n\}$ that projects a given set of data objects from a high-dimensional vectorial or even non-vectorial data space onto a low-dimensional output space is refered to as *topographic* if it preserves the data objects' topography by its images, i.e. nearby objects are assigned onto nearby locations in output space. 37

Topography

For a given set $x_1, ..., x_n$ of data objects its topography is the set of all pairwise dissimilarity relations. Dissimilarities may be expressed by means of several formalisms: (metric) distance functions, ranks of distances, (geometric) neighbourhood relations. 37

Topology

The concept of topology is a restriction of topography such that pairwise dissimilarities are expressed by means of geometric neighbourhood relations that are invariant to scaling and transition. Identical topographies are caused

by clusters of identical geometrical shape. For example, adjacency of data vectors' Voronoi cells is a topology based on a simple geometrical relation. 38, 92

Voronoi cell

For a given set $\{y_1, ..., y_n\} \subset D$ the data space D decomposes into n subsets by assigning each element $y \in D$ to its nearest neighbour among the $y_1, ..., y_n$. These subsets $V(i) = \{y \in D : d(y, y_i) \leq d(y, y_k) \forall k = 1, ..., n\}$ are refered to as Voronoi cells. 23, 58, 96