Adaptive Multilevel Cluster Analysis by Self-Organizing Box Maps

Dissertation von

Tobias Galliat

eingereicht am
Fachbereich Mathematik und Informatik
der Freien Universität Berlin

im März 2002

Betreuer:

Prof. Dr. Dr. h.c. Peter Deuflhard
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustr. 7
14195 Berlin

Gutachter:

Prof. Dr. Dr. h.c. Peter Deuflhard
Prof. Dr. Peter Rentrop

Datum der Disputation:

10.07.2002

To my parents

Contents

Introduction **3**

1 Cluster Analysis in High-Dimensional Data **7**
1.1 Modeling 8
1.1.1 Geometric cluster problems 9
1.1.2 Dynamic cluster problems 12
1.2 Problem reduction via representative clustering 13
1.3 Efficient cluster description 16
1.4 How many clusters? 21

2 Decomposition **23**
2.1 General Definition 23
2.2 Approximate box decomposition 25
2.3 Decomposition based representative clustering 27
2.4 Efficient cluster description via approximate box decomposition . 34
2.4.1 Computation of membership rules 34
2.4.2 Discriminating attributes 36

3 Adaptive Decomposition by Self-Organized Neural Networks **41**
3.1 Self-Organizing Maps (SOM) 42
3.2 Self-Organizing Box Maps (SOBM) 44
3.3 Comparison SOM - SOBM 53
3.4 Computational complexity 56
3.5 Practical extensions 57
3.5.1 Pruning 58
3.5.2 Early stopping 58

4 Multilevel Representative Clustering **59**
4.1 General approach 59
4.2 Adaptive decomposition refinement 60
4.3 Approach based on Perron Cluster analysis 61

4.3.1 Theoretical background 62
4.3.2 Stochastic homogeneity functions 64

5 Applications **73**
5.1 Conformational analysis of biomolecules 73
5.1.1 Introduction 73
5.1.2 Adaptation of SOM and SOBM to cyclic data 76
5.1.3 Numerical results: HIV protease inhibitor 79
5.1.4 Prospect: Virtual screening 86
5.2 Cluster analysis of insurance customers 87
5.2.1 Modeling 87
5.2.2 Numerical results: Whiplash Injury Patients 87

Conclusion **91**

Appendix **93**

Symbols **95**

Bibliography **97**

Zusammenfassung **103**

Lebenslauf **105**

Introduction

One thing a child has to learn is to divide and to group objects based on their color, form or size, i.e. based on their attributes. Such an ability is very important for the improvement of abstract and logical human thinking. But it is also a very helpful ability in economics, industry, science or politics, where the identification and description of homogeneous groups — so called *clusters* — of customers, products, events or situations helps to structure information about these objects and therefore generates knowledge, which allows to make special, group depending offers or decisions. Unfortunately, the ad hoc identification and description of clusters by human beings usually gets impossible with increasing numbers of objects and attributes.

Clustering methods have been studied first in statistics[1], but nowadays, where the improvement of technology allows to store the data of millions of objects with hundreds of attributes in single databases, new techniques for *cluster analysis* are also suggested by researchers from the *machine learning/neural networks* area[2] and the *database* community[3]. Furthermore, a new direction of research called *Data Mining* or — according to the more general definition of Fayyad and Piatetsky-Shapiro [21] — *Knowledge Discovery in Databases (KDD)*, has been established, where algorithms are developed that are able to scan huge databases and to extract *knowledge patterns* within the data. Since clusters are important examples of such knowledge patterns, the development of fast and efficient clustering techniques is part of this fast growing research area.

The most popular clustering method is k*-means*, and most of the suggested algorithms in the literature are variants of this method. The basic idea of k-means is to determine k cluster representatives and to assign each object to the cluster with its representative closest to the object so that the sum of the squared distances between the objects and their corresponding representatives is minimized.

---

[1] See, e.g., the introductory textbooks by Duran and Odell [18] or Fukunaga [27].

[2] For an overview see the complementary textbooks by Bishop [7] and Ripley [53].

[3] Important research is not only done by database groups at university [64, 19, 32], but also from industrial groups like IBM'S Quest group [1].

An investigation of algorithms based on the k-means method or other frequently used clustering methods leads to the following observations:

- The computed clusters are *geometrically* based, i.e. the objects within the same cluster have the property that their distance is small if they are interpreted as points in a suitable metric space. For non-geometric cluster problems, the computed clusters are usually not satisfactory. An important example are *dynamic* cluster problems, where one is interested in the identification of *metastable* clusters. Here, the objects within the same cluster should exhibit a high probability for transitions between each other with regard to an underlying *dynamic system*.

- If the numbers of objects and attributes is high, heuristics are used to speed up the cluster identification process. Many of these heuristics are designed for special applications and therefore not generally usable. Further, a mathematical justification is very often missing.

- A correct number of clusters k has to be known a priori.

In the case of reversible dynamic cluster problems, the theory of *Perron Cluster* analysis that has been recently developed by DEUFLHARD ET AL. offers a new access. The key concept of Perron Cluster analysis is the identification of metastable clusters by computing *almost invariant aggregates* of a suitable stochastic matrix $\mathcal{S}$. Via an investigation of the eigenvalues and the eigenvectors of the matrix $\mathcal{S}$, not only a correct number of clusters k can be determined, but also the metastable clusters themselves. Without a problem reduction, the size of the matrix $\mathcal{S}$ depends on the number of objects that have to be clustered. Therefore Perron Cluster analysis is directly usable only for very small reversible dynamic cluster problems[4].

Self-organized neural networks, especially KOHONEN'S *Self-Organizing Maps*, can be used to replace groups of similar objects by single representatives. The representatives are related to each other in a way that tries to preserve the original cluster structure, i.e. a fitting clustering of the representatives should correspond to a fitting clustering of the original objects. In contrast to the k-means method and its variants, the number of representatives is usually much larger than any correct number of clusters. Therefore, self-organized neural networks can be used as a kind of pre-clustering process to reduce the complexity of a cluster problem.

The aim of this thesis is a fruitful combination of Perron Cluster analysis and self-organized neural networks within an *adaptive multilevel clustering approach*

[4] As a first remedy, the use of essential degrees of freedom in the spirit of [4] made it possible to identify metastable clusters of a small molecule via Perron Cluster analysis [59].

that allows a fast and robust identification and an efficient description of clusters in *high-dimensional* data. In a general variant that needs a correct number of clusters k as an input, this new approach is relevant for a great number of cluster problems since it uses a cluster model that covers geometrically, but also dynamically based clusters. Its essential part is a method called *representative clustering* that guarantees the applicability to large cluster problems: Based on an *adaptive decomposition* of the object space via self-organized neural networks, the original problem is reduced to a smaller cluster problem. The general clustering approach can be extended by Perron Cluster analysis so that it can be used for large reversible dynamic cluster problems, even if a correct number of clusters k is unknown a priori. The basic application of the extended clustering approach is the *conformational analysis* of biomolecules, with great impact in the field of *Drug Design*. Here, for the first time the analysis of practically relevant and large molecules like an *HIV protease inhibitor* becomes possible.

This thesis is divided into five chapters. It starts with a general mathematical definition of cluster analysis in high-dimensional data. The scalability problem of the identification step will be addressed and the idea of representative clustering will be presented. In the section following, a rigorous definition of efficient cluster description will be given. The first chapter closes with a survey of the difficulties that arise, if a correct number of clusters is not known a priori.

The second chapter establishes a concept of decomposition within cluster analysis. Based on a general definition we will present a special variant called approximate box decomposition. It will be shown that the concept of decomposition gives way to a significant cluster problem reduction via representative clustering without destroying the original cluster structure. In addition, the usefulness of approximate box decompositions for the computation of efficient cluster descriptions will be demonstrated.

In the following chapter, KOHONEN'S Self-Organizing Maps are used for the computation of adaptive decompositions. Further, a powerful extension called *Self-Organizing Box Maps* will be suggested that computes approximate box decompositions.

In the fourth chapter, we are going to present a multilevel clustering approach using representative clustering based on successively refined adaptive decompositions. After an introduction to the basic theory, we combine Perron Cluster analysis with our clustering approach so that it includes an automatic computation of a correct number of cluster for cluster problems with a stochastic homogeneity function.

The final chapter gives a comprehensive presentation of applications. Especially the conformational analysis of biomolecules will be described in detail and illustrated with numerical results.

Acknowledgment

Foremost, I would like to thank P. Deuflhard for guiding me into a fascinating research area and giving me the chance to work in his group. His constant encouragement and confidence were extremely helpful for the progress of this thesis.

Furthermore, I am indebted to J. Weyer, who has taught me to become a real applied mathematician. During the last years he spent much of his rare time, on giving me advice and support.

Finally, I have to thank my parents and my girl-friend Simone for showing me that live is a wonderful present.

Chapter 1

Cluster Analysis in High-Dimensional Data

Clustering can be loosely defined as partitioning a set of objects into a given number k of disjoint subsets, so called clusters, so that the homogeneity between objects within each cluster is strong. Instead of homogeneity, the terms relationship or similarity are used synonymously in the literature.

Obviously, the definition given above does only make sense together with a measure for the homogeneity between objects. In this case any possible set of k clusters has a certain quality, depending on the measured homogeneity between all objects within each cluster.

One easily checks that the number of ways to partition a set of n objects in k disjoint non-void subsets is given by [18]:

$$\mathcal{K}(n,k) := \frac{1}{k!}\sum_{i=0}^{k}\binom{k}{i}(-1)^i(k-i)^n. \tag{1.1}$$

The function $\mathcal{K}(n,k)$ grows exponentially fast in n. Already in a very small set of objects the number of possible partitionings in k disjoint subsets is staggering, e.g., for $n = 100$ objects, there are $\mathcal{K}(100,2) \approx 10^{30}$ ways to partition them in two subsets. It can be shown that the problem to compute a set of k clusters of high quality is NP-complete [33]. Therefore fast solutions usually can only be achieved by using heuristic algorithms.

In addition to the identification of clusters, one is also interested in their description, i.e. in rules that allow to determine the cluster membership of each object, based on its properties. Especially in the case of high-dimensional data, where the objects have a high number of properties, such rules have to be efficient in the sense that their number is as small as possible and that they depend on a minimal number of properties only.

Given the above terminology, we define *cluster analysis in high-dimensional data* as the process of fast identification and efficient description of clusters. The clusters have to be of high quality with regard to a suitably chosen homogeneity measure.

1.1 Modeling

In the following we suggest a general model for cluster problems, supposing that the measure for the relationship between objects is given explicitly. It will be shown that the model — in contrast to other models suggested in the literature that are designed for geometric cluster problems — is usable for different fields of applications, because it is not only suitable for a geometrically based modeling, but also for dynamic cluster problems.

Let $\mathcal{A} := \{A_1, \ldots, A_q\}$ be a set of not necessarily ordered domains and define $\Omega := \bigotimes_{j=1}^{q} A_j := \{(a_1, \ldots, a_q)^T \,|\, a_j \in A_j\,, j = 1, \ldots, q\}$. We will refer to $A_1, \ldots, A_q$ as the *attributes* of Ω and to q as the *dimension* of Ω. Each finite subset $V = \{v_1, \ldots, v_n\} \subset \Omega$, $n \geq 2$, is called a *data set* in Ω and for each *data object* $v_i := (v_{i,1}, \ldots, v_{i,q})^T \in V$, the value $v_{i,j} \in A_j$ denotes the *property* of v_i for attribute A_j. We will further call each function $f : \Omega \longrightarrow \mathbf{R}_0^+$ with $f(v) = 0 \iff v \notin V$ a *frequency function* for the data set V and we define $f(M) := \sum_{v \in M} f(v)$ for any subset $M \subset \Omega$.

Suppose now that there exists a function $h : \Omega \times \Omega \longrightarrow [0, 1]$ so that $h(v, w) = h(w, v)$ for any $v, w \in V$. Then h will be called a *homogeneity function* for the data set V. We set $h_{max}(V) := \max_{v,w \in V} h(v, w)$ and call two objects $v_1, v_2 \in V$ maximally homogeneous, if $h(v_1, v_2) = h_{max}(V)$.

Based on given functions f and h the problem of clustering V in a given number k of subsets can be stated in the following general way:

Definition 1.1.1 *Let $k \in \{1, \ldots, n\}$ and $\mathcal{C} := \{C_1, \ldots, C_k\}$ any set of k non-void subsets $C_s \subset V$.*
(i) If $\bigcup_{s=1}^{k} C_s = V$ and $C_s \cap C_t = \emptyset$ for $1 \leq s < t \leq k$, then we call $\mathcal{C}$ a k-cluster set of the data set V.
(ii) Let $\mathcal{C}$ any k-cluster set of V. If $\mathcal{C}$ maximizes the weighted intra-cluster homogeneity

$$\Gamma_{f,h}(\mathcal{C}) := \frac{1}{k} \sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v \in C_s} \sum_{w \in C_s} h(v, w) f(v) f(w) \;\rightarrow\; \max, \tag{1.2}$$

then we call $\mathcal{C}$ an optimal k-cluster set of (V, f, h).

1.1.1 Geometric cluster problems

Many of the traditional clustering methods, including the famous k-*means* method [46], have in common that they are geometrically driven, i.e. they suppose that Ω can be modeled as a metric space, e.g., $\Omega \subset \mathbf{R}^q$, and that the relationship between objects is given by a *distance function* $d : \Omega \longrightarrow \mathbf{R}_0^+$, satisfying the following requirements for all $v, w, z \in \Omega$:

$$\begin{aligned} &(D1) \quad d(v,w) \geq 0 \\ &(D2) \quad d(v,v) = 0 \\ &(D3) \quad d(v,w) = d(w,v) \\ &(D4) \quad d(v,w) \leq d(v,z) + d(z,w). \end{aligned}$$

In the case that $\Omega \subset \mathbf{R}^q$, the *Euclidean distance* function is often used:

$$d_{euclid}(v,w) := \|v-w\| := \sqrt{(v-w)^T(v-w)}\ , v, w \in \mathbf{R}^q.$$

The basic idea of almost all geometrically driven cluster methods is the identification of a k-cluster set $\mathcal{C} := \{C_1, \ldots, C_k\}$ so that $\sum_{s=1}^{k} \mathrm{cost}(C_s)$ is minimized, where $\mathrm{cost} : \wp(\Omega) \longrightarrow \mathbf{R}_0^+$ is a cost function based on the distance function. The methods differ in the choice of the cost and the distance function and the several possible optimization strategies lead to different cluster algorithms. Many popular algorithms try to minimize the *sum-of-squares* cost function [20]:

$$\mathrm{cost}(C_s) := \frac{1}{f(C_s)} \sum_{v \in C_s} \sum_{w \in C_s} d(v,w)^2 f(v) f(w) \ \rightarrow \ \min\ .$$

The corresponding cluster problem can be formulated within our general definition:

Lemma 1.1.2 *Let Ω be a metric space with a distance function $d : \Omega \longrightarrow \mathbf{R}_0^+$. Further let $V := \{v_1, \ldots, v_n\} \subset \Omega$, $n \geq 2$, be any finite data set in Ω and $f : V \longrightarrow \mathbf{R}_0^+$ be any frequency function for V. Finally suppose that $\mathcal{C}$ is any k-cluster set of V.*
(a) Then $h_d : \Omega \times \Omega \longrightarrow [0,1]$, with

$$h_d(v,w) := 1 - \frac{d(v,w)^2}{(\max_{\widetilde{v},\widetilde{w} \in V} d(\widetilde{v},\widetilde{w}))^2}\ \ , v, w \in \Omega.$$

is a homogeneity function for V.
(b) $\mathcal{C}$ is an optimal k-cluster set of (V, f, h), if and only if

$$\sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v \in C_s} \sum_{w \in C_s} d(v,w)^2 f(v) f(w) \ \rightarrow \ \min\ .$$

Proof: (a) h_d is well defined, because $h_d(v,w) \in [0,1]$ for all $v, w \in \Omega$. Since d is a distance function, i.e. $d(v,w) = d(w,v)$ for any $v, w \in \Omega$, one further checks that $h_d(v,w) = h_d(w,v)$ and therefore h_d is a homogeneity function.
(b) Since $\max_{\widetilde{v},\widetilde{w}\in V} d(\widetilde{v},\widetilde{w})$, $f(V)$ are constant and positive values, we have:

$$
\begin{aligned}
& \min \sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v\in C_s} \sum_{w\in C_s} d(v,w)^2 f(v) f(w) \\
\iff \quad & \min \sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v\in C_s} \sum_{w\in C_s} \frac{d(v,w)^2}{(\max_{\widetilde{v},\widetilde{w}\in V} d(\widetilde{v},\widetilde{w}))^2} f(v) f(w) \\
\iff \quad & \max\ f(V) - \sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v\in C_s} \sum_{w\in C_s} \frac{d(v,w)^2}{(\max_{\widetilde{v},\widetilde{w}\in V} d(\widetilde{v},\widetilde{w}))^2} f(v) f(w) \\
\iff \quad & \max \sum_{s=1}^{k} \left(f(C_s) - \frac{1}{f(C_s)} \sum_{v\in C_s} \sum_{w\in C_s} \frac{d(v,w)^2}{(\max_{\widetilde{v},\widetilde{w}\in V} d(\widetilde{v},\widetilde{w}))^2} f(v) f(w) \right) \\
\iff \quad & \max \sum_{s=1}^{k} \frac{1}{f(C_s)} \left(f(C_s)^2 - \sum_{v\in C_s} \sum_{w\in C_s} \frac{d(v,w)^2}{(\max_{\widetilde{v},\widetilde{w}\in V} d(\widetilde{v},\widetilde{w}))^2} f(v) f(w) \right) \\
\iff \quad & \max \sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v\in C_s} \sum_{w\in C_s} \left(1 - \frac{d(v,w)^2}{(\max_{\widetilde{v},\widetilde{w}\in V} d(\widetilde{v},\widetilde{w}))^2} \right) f(v) f(w) \\
\iff \quad & \max \frac{1}{k} \sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v\in C_s} \sum_{w\in C_s} h_d(v,w) f(v) f(w).
\end{aligned}
$$

□

If $d = d_{euclid}$, then the sum-of-squares cost function is equivalent to the cost function used by algorithms based on the k-means method:

Lemma 1.1.3 *Let $C \subset V \subset \mathbf{R}^q$ any non-void subset of V and $f : \Omega \longrightarrow \mathbf{R}_0^+$ any frequency function for the data set V. Then we have*

$$
\sum_{v\in C} \|v - \bar{m}_C\|^2 f(v) = \frac{1}{2} \frac{1}{f(C)} \sum_{v\in C} \sum_{w\in C} \|v - w\|^2 f(v) f(w)\,,
$$

where

$$
\bar{m}_C := \frac{1}{f(C)} \sum_{v\in C} f(v) v
$$

denotes the centroid of C.

Proof:

$$
\begin{aligned}
& \sum_{v\in C} \|v-\bar{m}_C\|^2 f(v) \\
= \;& \sum_{v\in C} v^T v f(v) - 2\left(\sum_{v\in C} f(v)v^T\right)\bar{m}_C + \sum_{v\in C} f(v)\bar{m}_C^T\bar{m}_C \\
= \;& \sum_{v\in C} v^T v f(v) \;-\; f(C)\bar{m}_C^T\bar{m}_C \\
= \;& \frac{1}{f(C)}\left(\sum_{v\in C} f(C)v^T v f(v) - f(C)^2\bar{m}_C^T\bar{m}_C\right) \\
= \;& \frac{1}{f(C)}\left(\sum_{v\in C}\sum_{w\in C} v^T v f(v)f(w) - \sum_{v\in C}\sum_{w\in C} v^T w f(v)f(w)\right) \\
= \;& \frac{1}{2}\frac{1}{f(C)}\left(2\sum_{v\in C}\sum_{w\in C} v^T v f(v)f(w) - 2\sum_{v\in C}\sum_{w\in C} v^T w f(v)f(w)\right) \\
= \;& \frac{1}{2}\frac{1}{f(C)}\sum_{v\in C}\sum_{w\in C}\left(v^T v f(v)f(w) - 2v^T w f(v)f(w) + w^T w f(w)f(v)\right) \\
= \;& \frac{1}{2}\frac{1}{f(C)}\sum_{v\in C}\sum_{w\in C}\|v-w\|^2 f(v)f(w)
\end{aligned}
$$

□

A combination of Lemma 1.1.2 and Lemma 1.1.3 guarantees that geometric cluster problems, where the k-means method is suitable, can always be formulated within the suggested general model. Figure 1.1 shows a simple example of such a cluster problem in R^2 with $k = 3$. In the following sections, we will use this example for demonstration purposes.

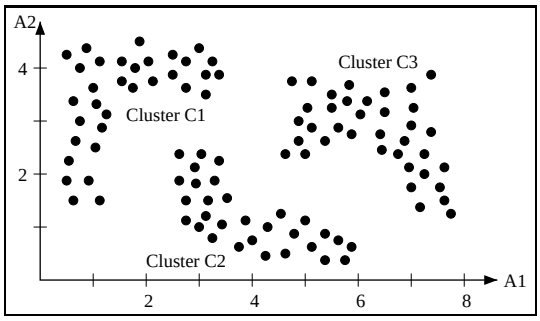


Figure 1.1: **Example: Clustering of data set in R^2 with $k = 3$.**

1.1.2 Dynamic cluster problems

Recently new cluster methods have been suggested using homogeneity measures not derived from a distance function or a more general data model [1, 5, 36]. The reason for this conceptual change is the emergence of new fields of application for cluster analysis, like e.g., the clustering of web-pages or of genomic data, where a geometrically driven modeling is often not suitable.

One of these new fields of application is the the analysis of dynamic systems. Here, an interesting problem is the identification of metastable sets of states, i.e. sets of states with a high probability that the dynamic system moves between states within the same set and a low probability of transitions between states of different sets. Although the state space of a dynamic system might be modeled as a geometric space, it is not advisable to equate metastable sets with geometrically based clusters inside this space: The dynamics between different states may not only depend on their geometric similarity. In the following we transform the identification of metastable sets of states of a dynamic system in a dynamic cluster problem, which will be described within our general model.

Let Ω be the set of all possible states of a dynamic system and choose any representative trajectory $X(1), \ldots, X(T) \in \Omega$. Set $V := \{X(t) \,|\, t = 1, \ldots, T\}$ and define a frequency function $f := \Omega \longrightarrow \mathbf{R}_0^+$ via $f(v) := |\{t \,|\, X(t) = v, \}|$, where $|M|$ denotes the number of elements in a finite set M. Further define for any $v, w \in V$:

$$S(v, w) := \frac{|\{t \,|\, X(t) = v, X(t+1) = w\}|}{f(v)} \tag{1.3}$$

so that $S(v, w)$ is the conditional probability of transitions from state v to state w in a single step. We can directly extend S on subsets of V, if we define for any non-void subsets $V_1, V_2 \subset V$:

$$\hat{S}(V_1, V_2) := \sum_{v \in V_1} \sum_{w \in V_2} \frac{f(v) S(v, w)}{f(V_1)}. \tag{1.4}$$

One easily checks that $\hat{S}(V_1, V_2)$ is the conditional probability of the dynamic system being in a state of set V_1 to move to a state of set V_2 in a single step.

The identification of k metastable sets of states of a dynamic system corresponds to the computation of k disjoint subsets $C_s \subset V$ so that $\hat{S}(C_s, C_s) \approx 1$ for $s = 1, \ldots, k$. Since this is equivalent to a maximization of $\sum_{s=1}^{k} \hat{S}(C_s, C_s)$, the identification of k metastable sets is equivalent to the identification of an optimal k-cluster set for (V, f, h_S) where h_S is a suitable homogeneity function:

Lemma 1.1.4 *Define* $h_S : \Omega \times \Omega \longrightarrow [0,1]$ *via*

$$h_S(v,w) := \begin{cases} \frac{1}{2}\left(\frac{S(v,w)}{f(w)} + \frac{S(w,v)}{f(v)}\right) & \text{if } v,w \in V \\ 0 \quad \text{else} \end{cases}$$

Then h_S *is a homogeneity function of* V.

Proof: Since $0 \leq |\{t \,|\, X(t) = v, X(t+1) = w\}| \leq f(v)$ for all $v, w \in V$, we have $S(v,w) \in [0,1]$. Therefore h_S is well defined and one easily checks that $h_S(v,w) = h_S(w,v)$ for any $v, w \in V$. □

Lemma 1.1.5 *For any* k*-cluster set* $\mathcal{C}$ *of* V *the weighted intra-cluster homogeneity with respect to* f *and* h_S *is given by*

$$\Gamma_{f,h_S}(\mathcal{C}) = \frac{1}{k}\sum_{s=1}^{k} \hat{S}(C_s, C_s).$$

Proof:

$$\begin{aligned}
\Gamma_{f,h_S}(\mathcal{C}) &= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v\in C_s}\sum_{w\in C_s} h_S(v,w) f(v) f(w) \\
&= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v\in C_s}\sum_{w\in C_s} \frac{1}{2}\left(f(v)S(v,w) + f(w)S(w,v)\right) \\
&= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{f(C_s)} \frac{1}{2}\left(\sum_{v\in C_s} f(v) \sum_{w\in C_s} S(v,w) + \sum_{w\in C_s} f(w) \sum_{v\in C_s} S(w,v)\right) \\
&= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{f(C_s)} \sum_{v\in C_s} f(v) \sum_{w\in C_s} S(v,w) \;=\; \frac{1}{k}\sum_{s=1}^{k} \hat{S}(C_s, C_s)
\end{aligned}$$

□

1.2 Problem reduction via representative clustering

A point very critical within the application of algorithms for the identification of clusters in high-dimensional data is the computational complexity, i.e. the correspondence between the time one needs to compute a solution and the number of data objects n, respectively the number of attributes q.

Suppose we have an algorithm that computes an optimal k-cluster set $\mathcal{C}$ of a data set V of size n and dimension q with respect to a frequency function f and

a homogeneity function h. One easily checks that we need $\mathcal{O}(n^2)$ values $h(v, w)$ to compute the weighted intra-cluster homogeneity $\Gamma_{f,h}(\mathcal{C})$. This usually makes a direct optimization of $\Gamma_{f,h}(\mathcal{C})$ impossible, if the number n is large. In the literature several heuristic optimization approaches are suggested, but unfortunately, most algorithms are designed for special applications and are therefore not generally usable. Moreover a mathematical justification is very often missing. In the following, we will describe another way to deal with large data sets that is motivated by principles of vector quantization and signal compression (see [35]) and that we will call *representative clustering*.

The reduction of cluster problems to a handier size via representative clustering rests upon the following assumption:

Optimal cluster assumption

Let $\mathcal{C}$ be any optimal k-cluster set of a data set $V \subset \Omega$ with respect to a frequency function f and a homogeneity function h. Then $\mathcal{C}$ assigns nearly maximally homogeneous objects in a predominant portion to the same cluster, i.e. if $C \in \mathcal{C}$ is any cluster and $v, w \in V$ are any data objects with $h(v, w) \leq h_{max}(V) - \epsilon$ for small $\epsilon > 0$, then usually we have: $v \in C \implies w \in C$.

Since each optimal k-cluster set of (V, f, h) maximizes the weighted intra-cluster homogeneity, this assumption should be true for most cluster problems.

Suppose now that the homogeneity function h meets the following two conditions:

- *Local maximum condition:* Objects $v_1, v_2 \in V$ are nearly maximally homogeneous, if they have nearly the same properties.
- *Global correspondence condition:* The homogeneity function h is nearly identical for any two nearly maximally homogeneous objects $v_1, v_2 \in V$:

$$h(v_1, v_2) \approx h_{max}(V) \implies h(v_1, v) \approx h(v_2, v) \text{ for all } v \in V.$$

In the case of geometric cluster problems, the possible homogeneity functions should meet the first condition and usually also the second one. For dynamic cluster problems, it is necessary that the state space Ω is build by a set of attributes. In this case moves between states with identical values for most attributes are usually very frequent, i.e. the local maximum condition holds, and typically, such states have very common dynamic properties, i.e. also the global correspondence condition holds.

If we successively replace objects $v_{i_1}, v_{i_2}, \ldots$ that have nearly the same properties by a representative object w_i, e.g., $w_i := v_{i_1}$, and define for w_i a compressed frequency value $\check{f}(w_i) := f(v_{i_1}) + f(v_{i_2}) + \ldots$, we come out with a data set $W = \{w_1, w_2, \ldots\}$ and a compressed frequency function $\check{f}$ of W.

Let $\mathcal{C} := \{C_1, \ldots, C_k\}$ be any optimal k-cluster set of $(W, \check{f}, h)$, then we can extend $\mathcal{C}$ on V, if we define $\hat{\mathcal{C}} := \{\hat{C}_1, \ldots, \hat{C}_k\}$ with $\hat{C}_s := \bigcup_{w_i \in C_s} \{v_{i_1}, v_{i_2}, \ldots\}$. Obviously $\hat{\mathcal{C}}$ is a k-cluster set of V. The local maximum condition assures that w_i and $v \in \{v_{i_1}, v_{i_2}, \ldots\}$ are nearly maximally homogeneous. Therefore the global correspondence condition guarantees:

$$
\begin{aligned}
& \Gamma_{\check{f},h}(\mathcal{C}) \\
= \quad & \frac{1}{k} \sum_{s=1}^{k} \frac{1}{\check{f}(C_s)} \sum_{w_i \in C_s} \sum_{w_j \in C_s} h(w_i, w_j) \check{f}(w_i) \check{f}(w_j) \\
= \quad & \frac{1}{k} \sum_{s=1}^{k} \frac{1}{f(\hat{C}_s)} \sum_{w_i \in C_s} \sum_{w_j \in C_s} h(w_i, w_j) \sum_{v_1 \in \{v_{i_1}, v_{i_2}, \ldots\}} f(v_1) \sum_{v_2 \in \{v_{j_1}, v_{j_2}, \ldots\}} f(v_2) \\
= \quad & \frac{1}{k} \sum_{s=1}^{k} \frac{1}{f(\hat{C}_s)} \sum_{w_i \in C_s} \sum_{v_1 \in \{v_{i_1}, v_{i_2}, \ldots\}} \sum_{w_j \in C_s} \sum_{v_2 \in \{v_{j_1}, v_{j_2}, \ldots\}} h(w_i, w_j) f(v_1) f(v_2) \\
\approx \quad & \frac{1}{k} \sum_{s=1}^{k} \frac{1}{f(\hat{C}_s)} \sum_{v_1 \in \hat{C}_s} \sum_{w_j \in C_s} \sum_{v_2 \in \{v_{j_1}, v_{j_2}, \ldots\}} h(v_1, w_j) f(v_1) f(v_2) \\
\approx \quad & \frac{1}{k} \sum_{s=1}^{k} \frac{1}{f(\hat{C}_s)} \sum_{v_1 \in \hat{C}_s} \sum_{v_2 \in \hat{C}_s} h(v_1, v_2) f(v_1) f(v_2) \\
= \quad & \Gamma_{f,h}(\hat{\mathcal{C}}).
\end{aligned}
$$

Suppose now that $\hat{\mathcal{C}}$ is not nearly optimal for (V, f, h). Then the optimal cluster assumption guarantees that there exist objects $v_1, v_2 \in V$ that are assigned to different clusters in $\hat{\mathcal{C}}$, although $h(v_1, v_2)$ is large. But this is a contradiction to the fact that nearly homogeneous objects are replaced by the same representative and therefore are assigned to the same cluster in $\hat{\mathcal{C}}$.

Let $V(j) := \{v_{*,j} \,|\, v = (v_{*,1}, \ldots, v_{*,q})^T \in V\}$ be the projection of V on the attribute A_j. Set $V_\Omega := \bigotimes_{j=1}^{q} V(j) = \{(a_1, \ldots, a_q)^T \,|\, a_j \in V(j)\,,\, j = 1, \ldots, q\}$. Obviously we have $V \subset V_\Omega \subset \Omega$ and $n = |V| \leq |V_\Omega| \leq n^q$. When analyzing high-dimensional data one often observes that V_Ω is rather sparse with respect to V, i.e. the *sparsity factor* $\frac{|V|}{|V_\Omega|}$ is very small. This guarantees that $|W|$ is smaller than n, i.e. we have reduced our cluster problem.

Figure 1.2 shows a reduction of our geometric cluster problem in R^2 via representative clustering in principle.

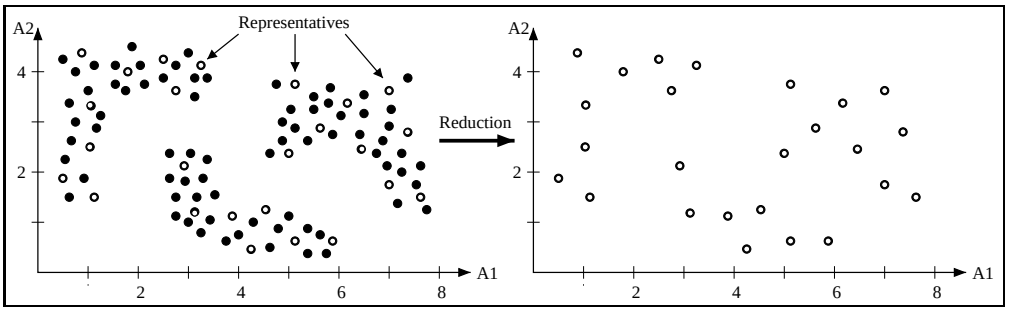


Figure 1.2: **Example: Reduction of geometric cluster problem in R^2.**

A problem reduction via representative clustering is only efficient, if $|W|$ is significantly smaller than the number n. Obviously the number of representatives depends strongly on the criterion that is used for the identification of objects with nearly the same properties. As a brute force approach one could think about using a very weak criterion that allows to replace much objects by the same representative. In this case the local maximum condition only holds, if we call two objects v_1, v_2 nearly maximal homogeneous, even if $h(v_1, v_2)$ is not so high. But then we cannot be sure that their homogeneity in relation to all other objects is nearly identical, i.e. that $h(v_1, v) \approx h(v_2, v)$ holds for all $v \in V$. If the global correspondence condition is violated too often, this usually has negative consequences for the quality of $\hat{\mathcal{C}}$.

In chapter 2 we will describe a concept called *decomposition* that can be used as a basis for the development of methods for an efficient problem reduction via representative clustering. We will replace the global correspondence condition for h by the construction of a compressed homogeneity function $\check{h}$ and define a more convenient condition that guarantees the optimality of $\hat{\mathcal{C}}$, if $\mathcal{C}$ is an optimal k-cluster set of $(W, \check{f}, \check{h})$. Moreover in chapter 4 a multilevel approach is presented that uses decomposition based representative clustering for a fast cluster identification.

1.3 Efficient cluster description

Besides the identification of clusters in high-dimensional data, also their efficient description is very important for most practical applications (see chapter 5). We want to know, which objects are homogeneous and also why they are homogeneous.

Obviously such a description can be achieved via rules that allow to determine the cluster membership of each object, based on its properties, i.e. rules like:

If $v = (v_{*,1}, \ldots, v_{*,q})^T \in V$ *has the properties* $v_{*,1} = a_1$ *and* $\ldots$ *and* $v_{*,q} = a_q$*, then* v *belongs to cluster* C_s*.*

A description based on such rules has to be consistent, i.e. it contains no rules assigning the same object v to different clusters.

Given any k-cluster set $\mathcal{C} := \{C_1, \ldots, C_k\}$ of a data set V in Ω, we can always generate rules for a cluster description in the following trivial way:

Define a function $c_\chi : V \longrightarrow \{1, \ldots, k\}$ via

$$c_\chi(v) := \sum_{s=1}^{k} s\, \chi_{C_s}(v) \quad \text{for all } v \in V,$$

where χ_{C_s} denotes the characteristic function of cluster C_s. Then for any object $v_i := (v_{i,1}, \ldots, v_{i,q})^T \in V$ we can state a rule r_i:

If $v = (v_{*,1}, \ldots, v_{*,q})^T$ *has the properties* $v_{*,1} = v_{i,1}$ *and* $\ldots$ *and* $v_{*,q} = v_{i,q}$*, then* v *belongs to cluster* $C_{c_\chi(v_i)}$*.*

Obviously the n rules $r_1, \ldots, r_n$ describe the clusters $C_1, \ldots, C_k$ consistently, but such a description is surely not efficient. We will demonstrate this by our example of a geometric cluster problem in $\mathbf{R}^2$ (see Fig. 1.1):

Cluster C_1 contains 33 data objects, i.e. we need 33 rules to describe this cluster if we use our trivial approach. If we allow rules that are slightly more complex, one easily checks that the following two rules are sufficient to describe cluster C_1:

If $v = (v_{*,1}, v_{*,2})^T$ *has the properties* $v_{*,1} = a_1$ *and* $v_{*,2} = a_2$ *with* $a_1 \in [0, 2]$*,* $a_2 \in [1, 5]$*, then* v *belongs to cluster* C_1*.*

If $v = (v_{*,1}, v_{*,2})^T$ *has the properties* $v_{*,1} = a_1$ *and* $v_{*,2} = a_2$ *with* $a_1 \in [2, 4]$*,* $a_2 \in [3, 5]$*, then* v *belongs to cluster* C_1*.*

This motivates the following definition of cluster membership rules:

Definition 1.3.1 *For any set* $\mathcal{B} := \{B_1, \ldots, B_q\}$ *with* $B_j \subset A_j$ *for* $j = 1, \ldots, q$*, we call* $r_\mathcal{B} : \Omega \longrightarrow \{0, 1\}$ *with*

$$r_\mathcal{B}(v) := \begin{cases} 1 & \text{if } (\forall j \in \{1, \ldots, q\})\ v_{*,j} \in B_j \\ 0 & \text{else} \end{cases}, \quad v := (v_{*,1}, \ldots, v_{*,q})^T \in \Omega,$$

a membership rule for cluster C_s*, if*

$$r_\mathcal{B}(v) = 1 \implies v \in C_s \quad \text{for all } v \in V.$$

Usually we need a set $r_s := \{r_{s,1}, \ldots, r_{s,m_s}\}$ of $m_s \in \mathbf{N}^+$ membership rules for each cluster C_s, to guarantee that each object $v \in C_s$ is assigned to cluster C_s by at least one rule, i.e. that we have

$$v \in C_s \implies (\exists\, r \in r_s)\, r(v) = 1 \quad \text{for all } v \in V.$$

We call such a set r_s a *complete membership rule set* for cluster C_s.

Based on complete membership rule sets for each cluster C_s, we can easily generate a description of $\mathcal{C}$:

Lemma 1.3.2 *Suppose there exists for each Cluster C_s of $\mathcal{C}$ a complete membership rule set $r_s := \{r_{s,1}, \ldots, r_{s,m_s}\}$. Let $\mathcal{H}_0$ denote the Heaviside function with*

$$\mathcal{H}_0(t) := \begin{cases} 0 & \text{if } t < 0 \\ 1 & \text{if } t \geq 0. \end{cases}$$

Then the function $c_r : V \longrightarrow \{1, \ldots, k\}$ with

$$c_r(v) := \sum_{s=1}^{k} s\, \mathcal{H}_0(-1 + \sum_{j=1}^{m_s} r_{s,j}(v)) \quad \text{for all } v \in V.$$

is a consistent description for $\mathcal{C}$, i.e. we have

$$c_r(v) = s \iff v \in C_s \quad \text{for all } v \in V.$$

Proof: "$\Longleftarrow$": Choose any $s \in \{1, \ldots, k\}$ and any $v \in C_s$. Since r_s is a complete membership rule set, there exists an $t \in \{1, \ldots, m_s\}$ so that $r_{s,t}(v) = 1$. Therefore we have $\mathcal{H}_0(-1 + \sum_{j=1}^{m_s} r_{s,j}(v)) = 1$. Suppose now that there exists another $p \in \{1, \ldots, k\}$ with $p \neq s$ and $\mathcal{H}(-1 + \sum_{j=1}^{m_p} r_{p,j}(v)) = 1$. If this is the case, there must exist a $\widetilde{t} \in \{1, \ldots, m_p\}$ so that $r_{p,\widetilde{t}}(v_i) = 1$. Since $r_{p,\widetilde{t}}$ is a membership rule for Cluster C_p, this implies $v \in C_p$. But this is a contradiction to $v \in C_s$. Therefore we have $c_r(v) = s$.
"$\Longrightarrow$": Choose any $s \in \{1, \ldots, k\}$ and any $v \in V \setminus C_s$. Since $\mathcal{C}$ is a k-cluster set of V there exists a $p \in \{1, \ldots, k\}$ with $p \neq s$ and $v \in C_p$. As already proofed above this guarantees $c_r(v) = p$ and therefore $c_r(v) \neq s$. □

Let $v = (v_{*,1}, \ldots, v_{*,q}) \in V$ be any data object and let $c_r : V \longrightarrow \{1, \ldots, k\}$ be a consistent description of $\mathcal{C}$ with corresponding complete membership rule sets $r_1, \ldots, r_k$. Then the determination of the cluster membership of v is rather simple: Find a membership rule $r_{\mathcal{B}} \in \bigcup_{s=1}^{k} r_s$ with $r_{\mathcal{B}}(v) = 1$, i.e, with $v_{*,j} \in B_j$ for $j = 1, \ldots, q$. Since c_r is consistent, there exists exactly one $s \in \{1, \ldots, k\}$

with $r_{\mathcal{B}} \in r_s$. Therefore data object v belongs to cluster C_s. Note that the existence of more than one membership rule $r \in r_s$ with $r(v) = 1$ is possible.

Obviously descriptions should be efficient in the sense that the corresponding complete membership rule sets $r_s := \{r_{s,1}, \ldots, r_{s,m_s}\}$ are minimal. i.e. the numbers m_s are as small as possible.

Often not all properties of a data object have to be considered to determine its cluster membership. Especially in the case of high-dimensional data, with a great number q of attributes A_j, a description based on a reduced set of attributes is of great interest.

We will illustrate this again by our two-dimensional example. Suppose that we restrict our data set to the data objects of cluster C_1 and cluster C_3. Then the following two rules will be sufficient to describe the clusters:

If $v = (v_{*,1}, v_{*,2})^T$ has the property $v_{*,1} = a_1$ with $a_1 \in [0, 4]$, then v belongs to cluster C_1.

If $v = (v_{*,1}, v_{*,2})^T$ has the property $v_{*,1} = a_1$ with $a_1 \in [4.5, 8]$, then v belongs to cluster C_3.

Obviously we only need attribute A_1 for a description of cluster C_1 and C_3, i.e. attribute A_2 has no influence on the discrimination of both clusters. Note that this is not true, for a description that includes cluster C_2.

We can easily extend our earlier definitions to work with reduced attribute sets:

Let $J := \{j_1, \ldots, j_m\} \subset \{1, \ldots, q\}$ any index subset of length m and let $\mathcal{A}(J) := \{A_j \,|\, j \in J\}$ be a reduced set of attributes of Ω. Set $\Omega(J) := \bigotimes_{j_t \in J} A_{j_t}$ and for $v := (v_{*,1}, \ldots, v_{*,q})^T \in \Omega$ denote by $v(J) := (v_{*,j_1}, \ldots, v_{*,j_m})^T \in \Omega(J)$ the projection on $\Omega(J)$. Further set $M(J) := \{v(J) \,|\, v \in M\} \subset \Omega(J)$ for any subset $M \subset \Omega$.

We can define *J-reduced membership rules* as a special kind of membership rules:

Definition 1.3.3 *Let $r_{\mathcal{B}}$ be any membership rule with $\mathcal{B} := \{B_1, \ldots, B_q\}$ and $B_j \subset A_j$ for $j = 1, \ldots, q$. We call $r_{\mathcal{B}}$ J-reduced, if $B_j = A_j$ for $j \notin J$. Let further r_s be a complete membership rule set of cluster C_s. We call r_s a complete J-reduced membership rule set, if each membership rule $r \in r_s$ is J-reduced.*

There exists an unique projection of any J-reduced membership rule on the subspace $\Omega(J)$:

Lemma 1.3.4 *Let $r_{\mathcal{B}}$ be any J-reduced membership rule with $\mathcal{B} := \{B_1, \ldots, B_q\}$ and $B_j \subset A_j$ for $j = 1, \ldots, q$. Then the function $\bar{r}_{\mathcal{B}} : \Omega(J) \longrightarrow \{0, 1\}$ with*

$$\bar{r}_{\mathcal{B}}(\bar{v}) := \begin{cases} 1 & \text{if } (\forall j \in J)\ v_{*,j} \in B_j \\ 0 & \text{else} \end{cases} \quad , \quad \bar{v} := (v_{*,j_1}, \ldots, v_{*,j_m})^T \in \Omega(J)$$

is the unique projection of $r_{\mathcal{B}}$ on $\Omega(J)$.

Proof: For any $v = (v_{*,1}, \ldots, v_{*,q})^T \in \Omega$ we have $v_{*,j} \in A_j = B_j$ for $j \notin J$, and therefore $r_{\mathcal{B}}(v) = \bar{r}_{\mathcal{B}}(v(J))$. □

Analogously to Lemma 1.3.2 we can achieve a description based on the reduced set of attributes $\mathcal{A}(J)$, if there exists for each cluster a complete J-reduced membership rule set:

Lemma 1.3.5 *Let $J \subset \{1, \ldots, q\}$ be any index subset of length m. Suppose there exists for each Cluster C_s of $\mathcal{C}$ a complete J-reduced membership rule set $r_s := \{r_{s,1}, \ldots, r_{s,m_s}\}$ and $\bar{r}_{s,j}$ denotes the unique projection of the membership rule $r_{s,j}$ on $\Omega(J)$, then the function $c_r : V \longrightarrow \{1, \ldots, k\}$ with*

$$c_r(v(J)) := \sum_{s=1}^{k} s\, \mathcal{H}_0(-1 + \sum_{j=1}^{m_s} \bar{r}_{s,j}(v(J))) \quad \text{for all}\, v \in V,$$

is a consistent description for $\mathcal{C}$ based on the reduced attribute set $\mathcal{A}(J)$, i.e. we have

$$c_r(v(J)) = s \iff v \in C_s \quad \text{for all } v \in V.$$

Obviously descriptions should be efficient in the sense that they are based on a maximally reduced attribute set $\mathcal{A}(J)$, i.e. $\mathcal{A}(J)$ should contain as less attributes as possible.

Efficient cluster description algorithm

Using the above definitions, the following general algorithm generates an efficient cluster description for a k-cluster set $\mathcal{C} := \{C_1, \ldots, C_k\}$ of a data set $V \in \Omega$:
(1) Find an index subset $J = \{j_1, \ldots, j_m\} \subset \{1, \ldots, q\}$ of minimal size so that there exists a function $c : V \longrightarrow \{1, \ldots, k\}$ with

$$c(v(J)) = s \iff v \in C_s \quad \text{for all } v \in V.$$

(2) Compute for each cluster C_s a minimally complete J-reduced membership rule set $r_s := \{r_{s,1}, \ldots, r_{s,m_s}\}$.
(3) Use $r := \{r_1, \ldots, r_k\}$ to construct a consistent description c_r of $\mathcal{C}$ based on the reduced attribute set $\mathcal{A}(J)$.

Since we are analyzing high-dimensional data, i.e. the dimension q is large, we obviously need heuristic solutions for step (1) and (2). For the development of suitable methods the concept of decomposition is very helpful: In section 2.4 we will describe techniques for the computation of membership rule sets based on *approximate box decompositions* and we will introduce the concept of *discriminating attributes* that allows the construction of heuristic algorithms to identify optimally reduced attribute sets $\mathcal{A}(J)$.

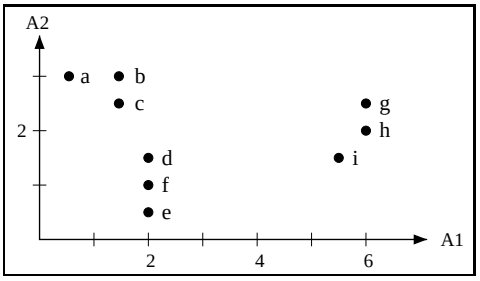
1.4 How many clusters?

Up to now, we have supposed that the number of clusters k is known a priori. But in many real world applications this is not the case. Looking at Eq. (1.1) one easily checks that the number of possible k-cluster sets explodes, if k is a further unknown parameter of the cluster problem. Obviously k is the most important parameter, i.e. with the words of cluster expert J. BEZDEK: *"It is clearly more important to be looking in the right solution space (within k) than it is to be comparing partitions across k because k specifies the number of clusters to look for, while the other parameters control the search for these substructures."* [6].

The definition of a general model for cluster problems with unknown cluster number is still an open problem. Usually it is not suitable to determine a correct number of clusters by computing for different k the optimal k-cluster sets $\mathcal{C}(k)$ and comparing the weighted intra-cluster homogeneities $\Gamma_{f,h}(\mathcal{C}(k))$, because most homogeneity functions tend to prefer extreme clusterings with $k = 1$ or $k = n$.

Example: Cluster problem with unknown number of clusters

We will illustrate this by the following simple example: Suppose we want to compute an optimal clustering of a data set $V = \{a, b, c, d, e, f, g, h, i\} \subset \mathbf{R}^2$ with a frequency function so that $f(v) = 1$ for all $v \in V$. We choose $h = h_d$ (see Lemma 1.1.2) based on the Euclidean distance function $d = d_{euclid}$. Figure 1.3 shows a plot of V and the corresponding homogeneity matrix.



	a	b	c	d	e	f	g	h	i
a	1.00	0.97	0.96	0.86	0.80	0.73	0.02	0.00	0.13
b	0.97	1.00	0.99	0.92	0.86	0.79	0.34	0.32	0.42
c	0.96	0.99	1.00	0.96	0.92	0.86	0.35	0.34	0.46
d	0.86	0.92	0.96	1.00	0.99	0.97	0.46	0.48	0.61
e	0.80	0.86	0.92	0.99	1.00	0.99	0.42	0.46	0.60
f	0.73	0.79	0.86	0.97	0.99	1.00	0.36	0.42	0.58
g	0.02	0.34	0.35	0.46	0.42	0.36	1.00	0.99	0.96
h	0.00	0.32	0.34	0.48	0.46	0.42	0.99	1.00	0.98
i	0.13	0.42	0.46	0.61	0.60	0.58	0.96	0.98	1.00

Figure 1.3: **Example: Cluster problem in R^2 with unknown cluster number k.** Left hand side: Plot of data set V. Right hand side: Homogeneity matrix of V based on Euclidean distance.

In Table 1.1 the optimal k-cluster sets $\mathcal{C}(k)$ of (V, f, h) and their weighted intra-cluster homogeneities $\Gamma_{f,h}(\mathcal{C}(k))$ are presented for different k. Obviously one would expect $k = 2, 3$ or 4 as a correct number of clusters, but a maximization of $\Gamma_{f,h}(\mathcal{C}(k))$ leads always to $k = 1$. Therefore we cannot use $\Gamma_{f,h}(\mathcal{C}(k))$ to judge which k is best.

optimal k-cluster set $\mathcal{C}(k)$	$\Gamma_{f,h}(\mathcal{C}(k))$
$\mathcal{C}(1) := V$	6.17
$\mathcal{C}(2) := \{\{a,b,c,d,e,f\},\{g,h,i\}\}$	4.24
$\mathcal{C}(3) := \{\{a,b,c\},\{d,e,f\},\{g,h,i\}$	2.96
$\mathcal{C}(4) := \{\{a\},\{b,c\},\{d,e,f\},\{g,h,i\}\}$	2.23
$\mathcal{C}(9) := \{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g\},\{h\},\{i\}\}$	1.00

Table 1.1: **Example: Optimal k-cluster sets of (V, f, h) for different k.**

In the literature [42, 6, 25, 51] several other measures are suggested to determine the validity of a given k-cluster set and so to find the optimal clustering, but all of these measures have the deficit that they first need the computation of optimal k-cluster sets for different k. In the worst scenario this requires the solution of n optimization problems. If n is large, this is a really heroic task.

Another possibility to cope with the problem of the unknown number of clusters might be to determine it in a pre-processing step. Via a projection of the high-dimensional data on a two-dimensional plane, one hopes that the cluster structure is not destroyed through the transformation and the number of clusters can be determined by visual investigation. A very popular tool for such a projection are *multidimensional-scaling* methods [49], e.g., SAMMON'S non-linear mapping algorithm [56]. The deficits of projection methods are obvious: For high-dimensional data it is unlikely that the cluster structure on the two-dimensional plane reflects the original structure. Moreover a visual investigation could be very subjective.

For cluster problems with a special type of homogeneity functions, exhibiting a stochastic property, we will present in chapter 4 a new method based on the theory of *Perron Cluster* analysis that allows the computation of a correct number of clusters. We will show that this method can be easily used together with the suggested multilevel cluster identification approach.

Chapter 2

Decomposition

In different research fields, decomposition usually describes the process of splitting a problem in smaller problems with less complexity. As was already motivated in section 1.2, a suitable reduction of a cluster problem can be achieved via a grouping of nearly maximally homogeneous objects and a representation of each group by a single object with compressed frequency value. If this kind of partitioning of the data set V exhibits a certain homogeneity property, we will call it a decomposition. After giving a general definition, we will introduce a special type of decomposition, the so called *approximate box decomposition*. Here the objects are pre-grouped in a way that they build a special subspace in Ω that has the shape of a multidimensional box if Ω is a metric space. We will develop a theory for an efficient reduction of cluster problems via representative clustering based on decomposition and we will present a basic reduction algorithm that will be refined in chapter 4. Finally we will show how an approximate box decomposition can be used to derive an efficient cluster description based on a minimal number of so called *discriminating attributes*.

2.1 General Definition

Let $V = \{v_1, \ldots, v_n\} \subset \Omega$ be any data set in Ω with frequency function f and homogeneity function h.

Definition 2.1.1 *Assume $n_k \in \mathbf{N}$ with $n_k \leq n$ and $\epsilon \in \mathbf{R}_0^+$ with $\epsilon \leq h_{max}(V)$. We call $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ an ϵ-decomposition of (V, h) with partitions Θ_s, if*

$$\bigcup_{s=1}^{n_k} \Theta_s = V \ , \quad \Theta_s \neq \emptyset \, , \quad \Theta_s \cap \Theta_p = \emptyset \quad \text{for } 1 \leq s < p \leq n_k$$

$$\text{and} \quad h(v, w) \geq h_{max}(V) - \epsilon \ \text{ for all } v, w \in \Theta_s, s = \{1, \ldots, n_k\}.$$

We further call

$$\vartheta_{f,h}(\Theta) := \frac{1}{f(V)} \sum_{s=1}^{n_k} \frac{1}{f(\Theta_s)} \sum_{v \in \Theta_s} \sum_{w \in \Theta_s} (h_{max}(V) - h(v,w)) f(v) f(w) \rightarrow \min$$

the decomposition error of Θ with respect to f and h.

Since $0 \leq h(v,w) \leq h_{max}(V)$ for all $v, w \in \Omega$, any n_k-clustering of V is an ϵ-decomposition of (V,h) with $\epsilon = h_{max}(V)$. The following Lemma guarantees $\vartheta_{f,h}(\Theta) \in [0, h_{max}(V)]$ for any ϵ-decomposition of (V,h):

Lemma 2.1.2 *Let Θ any ϵ-decomposition of (V,h), then we have: $\vartheta_{f,h}(\Theta) \leq \epsilon$.*

Proof: We have $(h_{max}(V) - h(v,w)) \leq \epsilon$ for all $v, w \in \Theta_s$ and therefore

$$\begin{aligned}
\vartheta_{f,h}(\Theta) &\leq \frac{1}{f(V)} \sum_{s=1}^{n_k} \frac{1}{f(\Theta_s)} \sum_{v \in \Theta_s} \sum_{w \in \Theta_s} \epsilon f(v) f(w) \\
&= \frac{\epsilon}{f(V)} \sum_{s=1}^{n_k} \frac{1}{f(\Theta_s)} \sum_{v \in \Theta_s} f(v) \sum_{w \in \Theta_s} f(w) \\
&= \frac{\epsilon}{f(V)} \sum_{s=1}^{n_k} \frac{1}{f(\Theta_s)} \sum_{v \in \Theta_s} f(v) f(\Theta_s) \\
&= \frac{\epsilon}{f(V)} \sum_{s=1}^{n_k} \sum_{v \in \Theta_s} f(v) = \frac{\epsilon}{f(V)} \sum_{s=1}^{n_k} f(\Theta_s) = \frac{\epsilon}{f(V)} f(V) = \epsilon
\end{aligned}$$

□

We will refer to Θ as a decomposition of V, if there exists a homogeneity function h and an $\epsilon \in [0, h_{max}(V)]$ so that Θ is an ϵ-decomposition of (V,h).

If we use the homogeneity measure $h = h_d$ (see Lemma 1.1.2) based on a distance function d, one easily checks that we have $h_{max} = 1$ and

$$\vartheta_{f,h}(\Theta) = \frac{1}{f(V)} \frac{1}{\max_{\widetilde{v},\widetilde{w} \in V} d(\widetilde{v},\widetilde{w})^2} \sum_{s=1}^{n_k} \frac{1}{f(\Theta_s)} \sum_{v \in \Theta_s} \sum_{w \in \Theta_s} d(v,w)^2 f(v) f(w).$$

Therefore, in this special case, we can use algorithms that try to optimize the sum-of-squares cost function to compute a decomposition for given n_k with minimal decomposition error. Figure 2.1 shows two possible decompositions with $n_k := 6$ partitions Θ_s for our example of a geometric cluster problem in R^2 using the Euclidean distance function $d = d_{euclid}$. The decomposition on the left hand side has been computed automatically via a simple hierarchical optimization method and leads to $\epsilon = 0.137$ and $\vartheta_{f,h}(\Theta) = 0.019$. The decomposition on the right hand side has been additionally optimized manually and leads to $\epsilon = 0.135$ and $\vartheta_{f,h}(\Theta) = 0.018$. Obviously ϵ is only a very rough upper bound of the decomposition error.

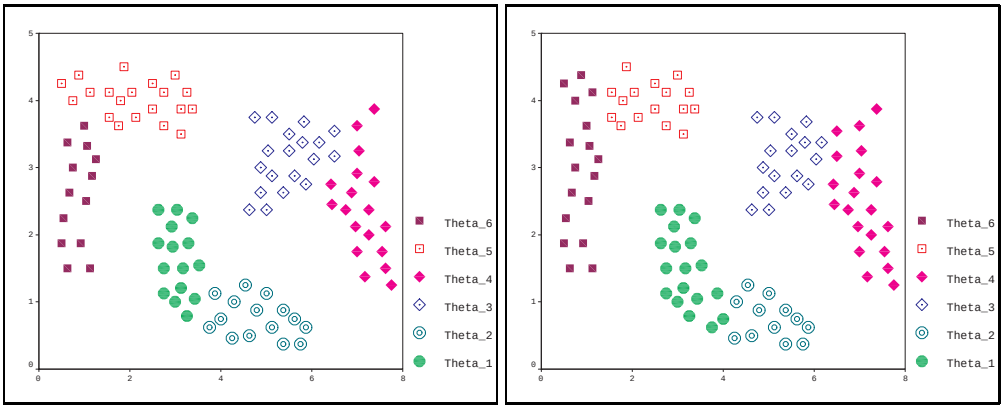


Figure 2.1: **Example: Two possible decompositions with six partitions in R^2.**

2.2 Approximate box decomposition

In the following we call any subset $B \subset \Omega$ a *box* in Ω, if there exist non-void subsets $B_1, \ldots, B_q$ with $B_j \subset A_j$ and $B = \bigotimes_{j=1}^{q} B_j$. We set $\mathrm{BOX}(\Omega) := \{B \mid B \,\mathrm{box\,in}\, \Omega\}$.

Definition 2.2.1 *Assume $n_k \in \mathbf{N}$ with $n_k \leq n$. We call (Θ,Δ) an approximate box decomposition of V with respect to f, whenever $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ is a decomposition of V and Δ is a set of n_k boxes $\Delta_1, \ldots, \Delta_{n_k} \in \mathrm{BOX}(\Omega)$ so that $\mathrm{overlap}_f(\Delta) \approx 0$ and $f(\Theta_s \cap \Delta_s) > 0$ for $s = 1, \ldots, n_k$. The value $\mathrm{overlay}_f(\Theta, \Delta) \in]0, 1]$ indicates how good Δ approximates Θ.*

Herein we use the terms *overlap* and *overlay* in the following way:

Definition 2.2.2 *Let $\mathcal{M} := \{M_1, \ldots, M_k\}$ be any set of $n_k \in N$ subsets of Ω with $f(M_s) > 0$ for $s = 1, \ldots, n_k$. Let Θ be a decomposition of V with n_k partitions Θ_s. Then the overlay of Θ and $\mathcal{M}$ with respect to f is given by*

$$\mathrm{overlay}_f(\Theta, \mathcal{M}) := \frac{1}{f(V)} \sum_{s=1}^{n_k} f(M_s \cap \Theta_s), \tag{2.1}$$

whereas the overlap of $\mathcal{M}$ with respect to f is given by

$$\mathrm{overlap}_f(\mathcal{M}) := \sum_{s=1}^{k} \frac{f(M_s \cap \bigcup_{p \neq s} M_p)}{f(\bigcup_{p=1}^{k} M_p)}. \tag{2.2}$$

If $\mathrm{overlay}_f(\Theta, \Delta) = 1$, we call (Θ, Δ) a perfect box decomposition of V, Note that if $\Delta(V) := \{\Delta_1 \cap V, \ldots, \Delta_{n_k} \cap V\}$ is a decomposition of V, $(\Delta(V), \Delta)$ is always a perfect box decomposition.

Figure 2.2 presents two approximate box decompositions based on the decompositions shown in Figure 2.1. On the left hand side, the six boxes does not approximate the decomposition perfectly, because two boxes overlap each other and four points are not covered, i.e. there is an insufficient overlay. On the right hand side of Figure 2.2, the decomposition is approximated perfectly with six boxes. Note that for the automatically computed decomposition, shown on the left hand side of Figure 2.1, no perfect approximation with six boxes is possible at all.

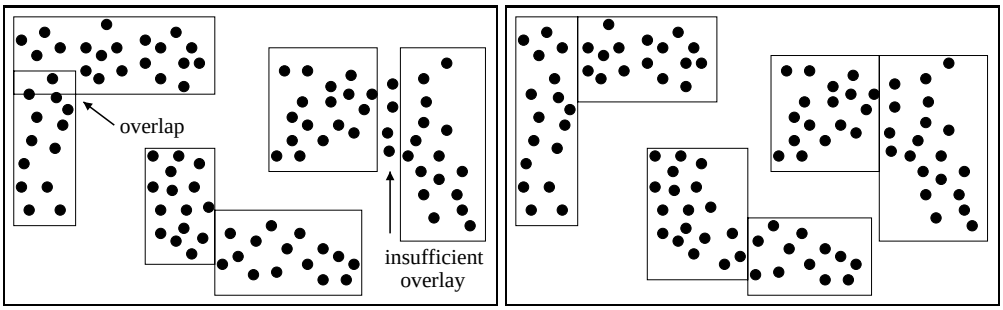


Figure 2.2: **Example: Approximate box decomposition ($n_k = 6$) in R^2.** Left hand side: Approximate box decomposition with insufficient overlay and overlap. Right hand side: Perfect box decomposition.

Example: Uniform box decomposition

We can always construct a perfect box decomposition: For $j \in \{1, \ldots, q\}$ choose any $m_j \in \mathbf{N}$ and any disjoint non-void subsets $B_{1,j}, \ldots, B_{m_j,j} \subset A_j$ so that $\bigcup_{i=1}^{m_j} B_{i,j} = A_j$. Set $m := \prod_{j=1}^{q} m_j$ and for any index tuple $(i_1, \ldots, i_q)$ with $1 \leq i_j \leq m_j$ choose an unique number $p = p(i_1, \ldots, i_q) \in \{1, \ldots, m\}$ and define $\Delta_p := \bigotimes_{j=1}^{q} B_{i_j,j}$. Obviously we have $\Delta_p \in \mathrm{BOX}(\Omega)$ for each $p \in \{1, \ldots, m\}$.

If we set $I(V) := \{p \,|\, \Delta_p \cap V \neq \emptyset\}$ and $\Delta_{I(V)} := \{\Delta_p \,|\, p \in I(V)\}$, then one easily checks that $(\Delta_{I(V)}(V), \Delta_{I(V)})$ is a perfect box decomposition of V because $\Delta_{I(V)}(V) := \{\Delta_p \cap V \,|\, p \in I(V)\}$ is a decomposition of V. Since the construction of Δ_p is uniform in the sense that each attribute of Ω is divided into m_j disjoint subsets, we call $(\Delta_{I(V)}(V), \Delta_{I(V)})$ an uniform box decomposition of Ω.

Note that the construction of the decomposition $\Delta_{I(V)}(V)$ is independent of the homogeneity function h and so the decomposition error is not guaranteed to be small. Further remember that, with increasing q, the number m grows exponentially, even if we split each attribute in only two subsets, i.e. if we set $m_j := 2$ for $j = 1, \ldots, q$. For example $q = 20$ leads to $m > 10^6$. So we usually have $m > n$ and therefore $|I(V)| \approx n$. But this makes an uniform box decomposition unsuitable for a reduction of high-dimensional cluster problems.

Figure 2.3 shows an example of an uniform box decomposition for our geometric cluster problem in R^2.

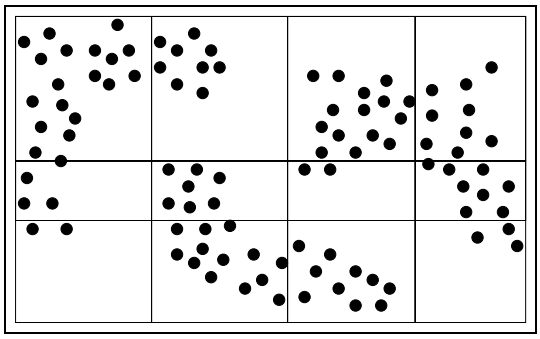


Figure 2.3: **Example: Uniform box decomposition in R^2.**

In chapter 3 we will present an adaptive method based on self-organized neural networks that allows to compute approximate box decompositions without the described shortages of an uniform procedure.

2.3 Decomposition based representative clustering

In section 1.2 we motivated the basic idea of a cluster problem reduction via representative clustering. We have presented a simple way to compute representatives $w_i \in \Omega$ with compressed frequency value $\check{f}(w_i)$. Further, we have shown that an optimal clustering of the representatives corresponds to an optimal clustering of the original data set V, if the homogeneity function h meets a local maximum and a global correspondence condition for all objects that are compressed to the same representative. Unfortunately this often leads to an unsatisfactory problem reduction, i.e. too many representatives are needed. The described conditions seems to be too strong for practical applications.

In this section we will develop a theory for cluster problem reduction via decomposition based representative clustering, without using any conditions for h. The objects are grouped together so that they are building partitions of a decomposition of the data set V. For the computation of an optimal k-cluster set of the representative set W, the original homogeneity function h is replaced by a compressed function $\check{h}$. We will show that if the decomposition is suitably fine, i.e. the decomposition error is small, this k-cluster set can be extended to an optimal k-cluster set of V with respect to f and h.

Definition 2.3.1 *Assume $n_k \in \mathbf{N}$ with $n_k \leq n$. Let $W := \{w_1, \ldots, w_{n_k}\} \subset V$ any subset of V and let Θ any decomposition of V with n_k partitions Θ_s.*
(i) We call W a codebook of Θ, if $w_s \in \Theta_s$ for $s = 1, \ldots, n_k$. We will refer to the data objects w_s as representatives or codebook vectors.
(ii) Let W any codebook of Θ, then we call the function $\check{f} : \Omega \longrightarrow \mathbf{R}_0^+$ with

$$\check{f}(w_s) := f(\Theta_s) \text{ for } s = 1, \ldots, n_k \text{ and } \check{f}(v) := 0 \text{ for } v \in \Omega \setminus W,$$

the compression of f on W. We set $\check{f}(M) := \sum_{w \in M} \check{f}(m)$ for any subset $M \subset \Omega$.
(iii) Let W any codebook of Θ, then we call the function $\check{h}_f : \Omega \longrightarrow [0, 1]$ with

$$\check{h}_f(w_s, w_p) := \frac{1}{\check{f}(w_s)\check{f}(w_p)} \sum_{v \in \Theta_s} \sum_{w \in \Theta_p} h(v, w) f(v) f(w) \text{ for } s, p = 1, \ldots, n_k$$

and $\check{h}_f(v, w) := 0$ for $v, w \in \Omega \setminus W$, the compression of h on W with respect to f.
(iv) For any k-cluster set $\mathcal{C} := \{C_1, \ldots, C_k\}$ of V, set $C_s(W) := C_s \cap W$. Then we call $\mathcal{C}(W) := \{C_1(W), \ldots, C_k(W)\}$ the compression of $\mathcal{C}$ on W.
(v) For any k-cluster set $\mathcal{C} := \{C_1, \ldots, C_k\}$ of a codebook W of Θ, we define $\hat{\mathcal{C}} := \{\hat{C}_1, \ldots, \hat{C}_k\}$ with $\hat{C}_s := \bigcup_{w_p \in C_s} \Theta_p$ and call $\hat{\mathcal{C}}$ the extension of $\mathcal{C}$ on V.

Lemma 2.3.2 *Assume $n_k \in \mathbf{N}$ with $k \leq n_k \leq n$ and let Θ be any decomposition of V with n_k partitions Θ_s and a codebook W. Then we have:*
(a) The compression $\check{f}$ is a frequency function for W and the compression $\check{h}_f$ is a homogeneity function for W.
(b) If $\mathcal{C}$ is a k-cluster set of W then the extension $\hat{\mathcal{C}}$ is a k-cluster set of V.

Proof: (a) and (b) follow directly from Definition 2.3.1. □

A decomposition is fine enough for a given k-cluster set, if each partition belongs to only one cluster:

Definition 2.3.3 *Let $\mathcal{C} := \{C_1, \ldots, C_k\}$ be any k-cluster set of V. Further assume $n_k \in \mathbf{N}$ with $k \leq n_k \leq n$ and let $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ be any decomposition of V. We call Θ a covering of $\mathcal{C}$, if there exist non-void disjoint index subsets $I_1, \ldots, I_k$ with $\bigcup_{s=1}^{k} I_s = \{1, \ldots, n_k\}$ so that $C_s = \bigcup_{p \in I_s} \Theta_p$.*

Obviously $\Theta_V := \{\{v\} \,|\, v \in V\}$ and $\Theta_{\mathcal{C}} := \mathcal{C}$ are trivial coverings of $\mathcal{C}$. But there exists also non-trivial coverings if $\mathcal{C}$ meets a stronger version of the optimal cluster assumption (see section 1.2):

Lemma 2.3.4 *Let $\mathcal{C}$ be any k-cluster set of V and $\epsilon \in \mathbf{R}_0^+$ with $\epsilon < h_{max}(V)$. If we have $(v \in C \implies w \in C)$ for any cluster $C \in \mathcal{C}$ and all $v, w \in V$ with $h(v, w) \geq h_{max}(V) - \epsilon$, then any ϵ-decomposition Θ of (V, h) is a covering of $\mathcal{C}$.*

Proof: Let $n_k \in \mathbf{N}$ with $k \leq n_k \leq n$ and $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ be any ϵ-decomposition of (V, h). For any cluster $C_s \in \mathcal{C}$ set $I_s := \{p \,|\, \Theta_p \cap C_s \neq \emptyset\}$. Then we have $\bigcup_{s=1}^{k} I_s = \{1, \ldots, n_k\}$ and $C_s \subset \bigcup_{p \in I_s} \Theta_p$. Obviously we are ready, if we show:

$$(\forall p \in I_s)\ \Theta_p \subset C_s.$$

But this follows directly: Since $p \in I_s$ there exists an object $v \in \Theta_p \cap C_s$. Then for all $w \in \Theta_p$ we have $h(v, w) \geq h_{max}(V) - \epsilon$ and therefore also $w \in C_s$. □

The next Lemma shows that the weighted intra-cluster homogeneity of any k-cluster set $\mathcal{C}$ of V and its compression on W are equal if there exists any covering of $\mathcal{C}$. We will use this fact in combination with Lemma 2.3.6 within the proof of the basic Theorem 2.3.7.

Lemma 2.3.5 *Let $\mathcal{C} := \{C_1, \ldots, C_k\}$ be any k-cluster set of V and Θ be any covering of $\mathcal{C}$ with n_k partitions Θ_p and a codebook $W := \{w_1, \ldots, w_{n_k}\}$. Then the compression $\mathcal{C}(W)$ is a k-cluster set of W with $\Gamma_{\check{f},\check{h}_f}(\mathcal{C}(W)) = \Gamma_{f,h}(\mathcal{C})$.*

Proof: Obviously $\mathcal{C}(W)$ is a k-cluster set, if $C_s(W) \neq \emptyset$ for $s = 1, \ldots, k$. But this follows immediately from the fact that Θ is a covering of $\mathcal{C}$ with codebook W. Further it follows that the index subsets $I_1, \ldots, I_k$ with $I_s := \{p \,|\, w_p \in C_s\}$ are non-void and disjoint and that we have $C_s = \bigcup_{p \in I_s} \Theta_p$.
Since $f(C_s) = \check{f}(C_s(W))$, this yields:

$$\begin{aligned}
\Gamma_{f,h}(\mathcal{C}) &= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{f(C_s)}\sum_{v\in C_s}\sum_{w\in C_s} h(v,w)f(v)f(w)\\
&= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{f(C_s)}\sum_{p_1\in I_s}\sum_{p_2\in I_s}\sum_{v\in\Theta_{p_1}}\sum_{w\in\Theta_{p_2}} h(v,w)f(v)f(w)\\
&= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{f(C_s)}\sum_{p_1\in I_s}\sum_{p_2\in I_s} \check{h}_f(w_{p_1},w_{p_2})\check{f}(w_{p_1})\check{f}(w_{p_2})\\
&= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{\check{f}(C_s(W))}\sum_{p_1\in I_s}\sum_{p_2\in I_s} \check{h}_f(w_{p_1},w_{p_2})\check{f}(w_{p_1})\check{f}(w_{p_2})\\
&= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{\check{f}(C_s(W))}\sum_{w_{p_1}\in C_s(W)}\sum_{w_{p_2}\in C_s(W)} \check{h}_f(w_{p_1},w_{p_2})\check{f}(w_{p_1})\check{f}(w_{p_2})\\
&= \Gamma_{\check{f},\check{h}_f}(\mathcal{C}(W))
\end{aligned}$$

□

The covering property of a decomposition can be transmitted to its extension:

Lemma 2.3.6 *Let Θ be any covering of $\widetilde{\mathcal{C}}$ with n_k partitions Θ_p and a codebook $W := \{w_1, \ldots, w_{n_k}\}$. If $\mathcal{C} := \{C_1, \ldots, C_k\}$ is a k-cluster set of W, then Θ is a covering of the extension $\hat{\mathcal{C}}$ of $\mathcal{C}$ on V.*

Proof: Set $J_s := \{p \,|\, w_p \in C_s\}$ for $s = 1, \ldots, k$. Since $\mathcal{C}$ is a k-cluster set of W, we have $J_s \neq \emptyset$, $J_s \cap J_p = \emptyset$ for $1 \leq s < p \leq k$ and $\bigcup_{s=1}^{k} J_s = \{1, \ldots, n_k\}$. By definition of $\hat{\mathcal{C}}$, we further have $\hat{C}_s := \bigcup_{w_p \in C_s} \Theta_p = \bigcup_{p \in J_s} \Theta_p$ and therefore Θ is a covering of $\hat{\mathcal{C}}$. □

Using the previous lemmata we can proof the basic theorem of decomposition based representative clustering:

Theorem 2.3.7 *Let $\widetilde{\mathcal{C}} := \{\widetilde{C}_1, \ldots, \widetilde{C}_k\}$ be any optimal k-cluster set of (V, f, h). Further let Θ be any covering of $\widetilde{\mathcal{C}}$ with n_k partitions Θ_p and a codebook W. If $\mathcal{C}$ is an optimal k-cluster set of $(W, \check{f}, \check{h}_f)$, then the extension $\hat{\mathcal{C}}$ is an optimal k-cluster set of (V, f, h).*

Proof: (i) Let $\widetilde{\mathcal{C}}(W) := \{\widetilde{C}_1(W), \ldots, \widetilde{C}_k(W)\}$ with $\widetilde{C}_s(W) := \widetilde{C}_s \cap W$ be the compression of $\widetilde{\mathcal{C}}$. Since Θ is an covering of $\widetilde{\mathcal{C}}$, we can apply Lemma 2.3.5 and yield:

$$\Gamma_{f,h}(\widetilde{\mathcal{C}}) \;=\; \Gamma_{\check{f},\check{h}_f}(\widetilde{\mathcal{C}}(W)).$$

(ii) Let $\hat{\mathcal{C}}(W) := \{\hat{C}_1(W), \ldots, \hat{C}_k(W)\}$ with $\hat{C}_s(W) := \hat{C}_s \cap W$ be the compression of $\hat{\mathcal{C}}$. Then one easily checks that $\hat{\mathcal{C}}(W) = \mathcal{C}$. Since Lemma 2.3.6 guarantees that Θ is a covering of $\hat{\mathcal{C}}$, we can again apply Lemma 2.3.5 and yield:

$$\Gamma_{f,h}(\hat{\mathcal{C}}) \;=\; \Gamma_{\check{f},\check{h}_f}(\mathcal{C}).$$

(iii) Since $\mathcal{C}$ is an optimal k-cluster set of $(W, \check{f}, \check{h}_f)$ and $\widetilde{\mathcal{C}}$ is an optimal k-cluster set of (V, f, h), we have

$$\Gamma_{\check{f},\check{h}_f}(\mathcal{C}) \;\geq\; \Gamma_{\check{f},\check{h}_f}(\widetilde{\mathcal{C}}(W)) \;\text{ and }\; \Gamma_{f,h}(\widetilde{\mathcal{C}}) \;\geq\; \Gamma_{f,h}(\hat{\mathcal{C}}).$$

Using $(i) - (iii)$ we get

$$0 \;\geq\; \Gamma_{f,h}(\hat{\mathcal{C}}) \;-\; \Gamma_{f,h}(\widetilde{\mathcal{C}}) \;=\; \Gamma_{f,h}(\hat{\mathcal{C}}) \;-\; \Gamma_{\check{f},\check{h}_f}(\widetilde{\mathcal{C}}(W)) \;\geq\; \Gamma_{f,h}(\hat{\mathcal{C}}) \;-\; \Gamma_{\check{f},\check{h}_f}(\mathcal{C}) \;=\; 0$$

and therefore $\Gamma_{f,h}(\hat{\mathcal{C}}) \;=\; \Gamma_{f,h}(\widetilde{\mathcal{C}})$. Since $\widetilde{\mathcal{C}}$ is an optimal k-cluster set, this guarantees that $\hat{\mathcal{C}}$ is also optimal. □

From Theorem 2.3.7 we can derive a basic algorithm for the reduction of cluster problems via representative clustering based on decomposition:

Basic reduction algorithm

Suppose we want to compute an optimal k-cluster set of a data set V with respect to a frequency function f and a homogeneity function h.
(1) To reduce the complexity of the cluster problem, we have to compute first a decomposition $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ of V and a codebook W so that Θ is an covering of an optimal k-cluster set of (V, f, h).
(2) Next we compute an optimal *representative clustering*, i.e. an optimal k-cluster set $\mathcal{C}$ of $(W, \check{f}, \check{h}_f)$.
(3) Finally we have to extend $\mathcal{C}$ on V. The resulting $\hat{\mathcal{C}}$ is an optimal k-cluster set of (V, f, h).

Obviously such an algorithm makes only sense if in step (1) the optimal k-cluster set has not to be known a priori and the number n_k is much smaller than the number n of objects in V.

Using the optimal cluster assumption (see section 1.2) and Lemma 2.3.4, we can suppose that for sufficiently small ϵ, each ϵ-decomposition of V is a covering of each optimal k-cluster set of (V, f, h). This motivates the following assumption:

Covering assumption

If a decomposition Θ of V is sufficiently fine, i.e. if $\vartheta_{f,h}(\Theta)$ is small, then there exists a nearly optimal k-cluster set of (V, f, h) so that Θ is a covering of it.

Obviously the fineness of Θ corresponds with the number of partitions n_k. Therefore we need a method that — given an upper bound of n_k — tries to compute a maximally fine decomposition, while using only a minimal number of partitions. In chapter 3 we will present such a method based on KOHONEN'S Self-Organizing Maps (SOM). Since the choice of the upper bound for n_k is rather arbitrary, in chapter 4 we will refine our basic reduction algorithm to a multilevel algorithm that iterates the steps (1) and (2) until a sufficiently fine decomposition and corresponding optimal representative clustering is found.

Example: Representative clustering of a geometric cluster problem in R^2

We will give a short demonstration of our basic reduction algorithm by our example of a geometric cluster problem in R^2.

Since $h_{max}(V) = 1$, any ϵ-decomposition Θ with $\epsilon = 0.05$ should be fine enough to use it within our algorithm. Figure 2.4 shows a suitable ϵ-decomposition of the 100 points in the data set V with $n_k = 10$ partitions.

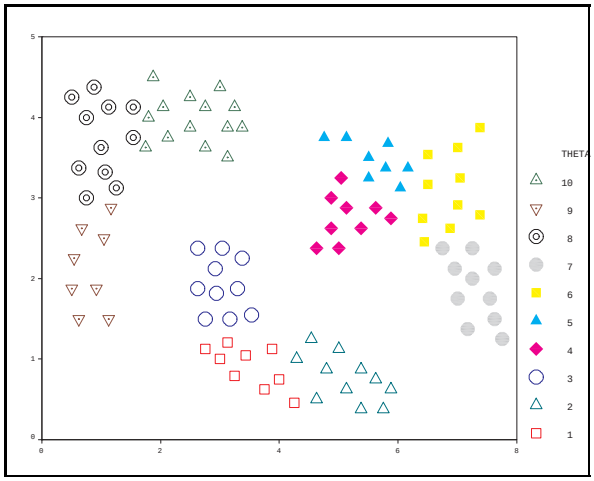


Figure 2.4: **Example: Covering with $n_k = 10$ partitions of 3-cluster set in R^2.**

Now we have to choose any codebook $W := \{w_1, \ldots, w_{10}\}$ of Θ and to compute the compressed functions $\check{f}$ and $\check{h}$ according to Definition 2.3.1.

One easily checks that $\{\{w_1, w_2, w_3\}, \{w_4, w_5, w_6, w_7\}, \{w_8, w_9, w_{10}\}\}$ is an optimal 3-cluster set of $(W, \check{f}, \check{h})$. An extension on V directly leads to the three clusters C_1, C_2 and C_3 (see Figure 1.1). Note that the 3-cluster set $\mathcal{C} := \{C_1, C_2, C_3\}$ meets the condition $(v \in C \implies w \in C)$ for any cluster $C \in \mathcal{C}$ and all $v, w \in V$ with $h(v, w) \geq h_{max}(V) - \epsilon$. Therefore Lemma 2.3.4 guarantees that our decomposition Θ is a covering of $\mathcal{C}$, i.e. that it was fine enough.

Decomposition clustering

Instead of clustering codebook vectors, we can also cluster a decomposition itself: Let $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ be any decomposition of V. Then Θ can be interpreted as a data set in $\hat{\Omega} := \wp(\Omega)$, where $\wp(\Omega) := \{M \mid M \subset \Omega\}$ denotes the power set of Ω. We can extend the frequency function f and the homogeneity function h on subsets of Ω:

Definition 2.3.8
(a) We call $\hat{f} : \wp(\Omega) \longrightarrow \mathbf{N}$ with $\hat{f}(M) := \sum_{v \in M} f(v)$ for any subset $M \subset \Omega$, the set extension of f. We set $\hat{f}(\mathcal{M}) := \sum_{M \in \mathcal{M}} \hat{f}(M)$ for $\mathcal{M} \subset \wp(\Omega)$.
(b) We call $\hat{h}_f : \wp(\Omega) \times \wp(\Omega) \longrightarrow [0, 1]$, with

$$\hat{h}_f(V_1, V_2) := \begin{cases} \frac{1}{\hat{f}(V_1)\hat{f}(V_2)} \sum_{v \in V_1} \sum_{w \in V_2} h(v, w) f(v) f(w) & \text{if } V_1 \cap V, V_2 \cap V \neq \emptyset \\ 0 & \text{else} \end{cases}$$

for any subsets $V_1, V_2 \subset \Omega$, the set extension of h with respect to f.

Note that we have $0 \leq \hat{h}_f(V_1, V_2) \leq 1$ and $\hat{h}_f(V_1, V_2) = \hat{h}_f(V_2, V_1)$ for any non-void subsets $V_1, V_2 \subset V$.

The following Theorem guarantees that the computation of an optimal k-cluster set of $(W, \check{f}, \check{h})$ is equivalent to the computation of an optimal k-cluster set of $(\Theta, \hat{f}, \hat{h})$, if Θ is any decomposition of V with codebook W. This makes it possible to replace the clustering of codebook vectors by a direct clustering of the corresponding partitions of the decomposition within step (2) of the basic reduction algorithm.

Theorem 2.3.9 *Let $W := \{w_1, \ldots, w_{n_k}\}$ be any codebook of Θ.*
(i) Let $\mathcal{C} := \{C_1, \ldots, C_k\}$ be any k-cluster set of Θ. Then there exist k non-void disjoint index subsets I_s with $\bigcup_{s=1}^{k} I_s = \{1, \ldots, n_k\}$ so that $C_s = \{\Theta_p \,|\, p \in I_s\}$. If we set $\check{C}_s(W) := \{w_p \,|\, p \in I_s\}$, then $\check{\mathcal{C}}(W) := \{\check{C}_1(W), \ldots, \check{C}_k(W)\}$ is a k-cluster set of W with $\Gamma_{\hat{f},\hat{h}_f}(\mathcal{C}) = \Gamma_{\check{f},\check{h}_f}(\check{\mathcal{C}}(W))$.
(ii) Let $\mathcal{C} := \{C_1, \ldots, C_k\}$ be any k-cluster set of W. If we set $I_s := \{p \,|\, w_p \in C_s\}$, then the index subsets $I_1, \ldots, I_k$ are non-void and disjoint with $\bigcup_{s=1}^{k} I_s = \{1, \ldots, n_k\}$. The extension $\hat{\mathcal{C}}(\hat{\Omega}) := \{\hat{C}_1(\hat{\Omega}), \ldots, \hat{C}_k(\hat{\Omega})\}$ with $\hat{C}_s(\hat{\Omega}) := \{\Theta_p \,|\, p \in I_s\}$ is a k-cluster set of Θ with $\Gamma_{\check{f},\check{h}_f}(\mathcal{C}) = \Gamma_{\hat{f},\hat{h}_f}(\hat{\mathcal{C}}(\hat{\Omega}))$.

Proof: Since (ii) follows analogously, we only show (i):

$$
\begin{aligned}
\text{(a)}\ \Gamma_{\hat{f},\hat{h}_f}(\mathcal{C}) &= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{\hat{f}(C_s)}\sum_{V_1\in C_s}\sum_{V_2\in C_s}\hat{h}_f(V_1,V_2)\hat{f}(V_1)\hat{f}(V_2)\\
&= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{\sum_{p\in I_s} f(\Theta_p)}\sum_{p_1\in I_s}\sum_{p_2\in I_s}\hat{h}_f(\Theta_{p_1},\Theta_{p_2})\hat{f}(\Theta_{p_1})\hat{f}(\Theta_{p_2})\\
&= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{\sum_{p\in I_s}\check{f}(w_p)}\sum_{p_1\in I_s}\sum_{p_2\in I_s}\sum_{v\in\Theta_{p_1}}\sum_{w\in\Theta_{p_2}}h(v,w)f(v)f(w)\\
&= \frac{1}{k}\sum_{s=1}^{k}\frac{1}{\check{f}(\check{C}_s(W))}\sum_{p_1\in I_s}\sum_{p_2\in I_s}\check{h}_f(w_{p_1},w_{p_2})\check{f}(w_{p_1})\check{f}(w_{p_2})\\
&= \Gamma_{\check{f},\check{h}_f}(\check{\mathcal{C}}(W))
\end{aligned}
$$

□

We will use this equivalence of representative clustering and decomposition clustering in the discussion of our main Theorem 4.3.9 in chapter 4.

2.4 Efficient cluster description via approximate box decomposition

In this section we will describe, how approximate box decompositions can be used to generate efficient cluster descriptions according to section 1.3.

2.4.1 Computation of membership rules

We can easily determine cluster membership rules for a k-cluster set $\mathcal{C}$, if we have an approximate box decomposition of V that is a covering of $\mathcal{C}$:

Lemma 2.4.1 *Assume $n_k \in \mathbf{N}$ with $k \leq n_k \leq n$. Let $\mathcal{C} := \{C_1, \ldots, C_k\}$ be any k-cluster set of V and $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ be any covering of $\mathcal{C}$ with non-void disjoint index subsets $I_1, \ldots, I_k$ so that $C_s = \bigcup_{p \in I_s} \Theta_p$. Further suppose the existence of any $\Delta := \{\Delta_1, \ldots, \Delta_{n_k}\}$ so that (Θ,Δ) is an approximate box decomposition of V with respect to f.*
(i) For $p \in \{1, \ldots, n_k\}$ there exist for each $j \in \{1, \ldots, q\}$ a subset $B_{p,j} \subset A_j$ so that $\Delta_p = \bigotimes_{j=1}^{q} B_{p,j}$.
(ii) Set $\mathcal{B}_p := \{B_{p,1}, \ldots, B_{p,q}\}$ for $p \in \{1, \ldots, n_k\}$ and define $r_{\mathcal{B}_p} : \Omega \longrightarrow \{0, 1\}$ with

$$r_{\mathcal{B}_p}(v) := \begin{cases} 1 & \text{if } (\forall j \in \{1, \ldots, q\})\ v_{*,j} \in B_{p,j} \\ 0 & \text{else} \end{cases}, \quad v := (v_{*,1}, \ldots, v_{*,q})^T \in \Omega.$$

If $p \in I_s$ and $f(\Delta_p \setminus C_s) = 0$, then $r_{\mathcal{B}_p}$ is a membership rule for cluster C_s.
(iii) If $f(\Delta_p \setminus C_s) = 0$ for all $p \in I_s$ and $C_s \subset \bigcup_{p \in I_s} \Delta_p$, then $r_s := \{r_{\mathcal{B}_p} \mid p \in I_s\}$ is a complete membership rule set of cluster C_s.

Proof: (i) Follows directly from $\Delta_p \in \mathrm{BOX}(\Omega)$.
(ii) We have

$$f(\Delta_p \setminus C_s) = 0 \iff \Delta_p \cap V \subset C_s$$

and therefore

$$r_{\mathcal{B}_p}(v) = 1 \implies v \in \Delta_p \subset C_s \ \text{ for all } v \in V.$$

(iii) From (ii) follows that $r_{\mathcal{B}_p}$ is a membership rule of C_s for each $p \in I_s$. Since $C_s \subset \bigcup_{p \in I_s} \Delta_p$, we have

$$v \in C_s \implies (\exists p \in I_s)\, v \in \Delta_p \iff (\exists p \in I_s)\, r_{\mathcal{B}_p}(v) = 1\,.$$

□

Note that the condition $f(\Delta_p \setminus C_s) = 0$ is only violated if boxes from different clusters overlap each other. Therefore this condition is weaker than the condition $\mathrm{overlap}_f(\Delta) = 0$.

Membership rule set algorithm

From Lemma 2.4.1 we can derive an algorithm to compute complete membership rule sets that are nearly minimal for a k-cluster set $\mathcal{C}$:
(1) Compute an approximate box decomposition (Θ, Δ) of V so that Θ is a covering of $\mathcal{C}$, Δ fits the conditions of Lemma 2.4.1 and $n_k \ll n$.
(2) Construct the n_k membership rules $r_{\mathcal{B}_p}$ as described in Lemma 2.4.1. Since for each cluster a minimally complete membership rule set must contain at least one rule, we need at least k membership rules to describe a k-cluster set $\mathcal{C}$. If the difference of n_k and k is not to large, the complete membership rule sets r_s are nearly minimal.

Example: Complete membership rule set for a 3-cluster set in R^2 based on approximate box decomposition.

For our geometrically based cluster problem in R^2 with $k = 3$, Figure 2.5 shows an approximate box decomposition (Ω, Δ) that covers the optimal 3-cluster set. Obviously the overlap between the boxes causes no problems and therefore we can use $\Delta := \{\Delta_1, \ldots, \Delta_{n_k}\}$, with boxes $\Delta_p = B_{p,1} \times B_{p,2}$ and subsets $B_{p,j} \subset \mathbf{R}$ according to Table 2.1, to determine minimal membership rule set for the optimal 3-cluster set $\{C_1, C_2, C_3\}$.

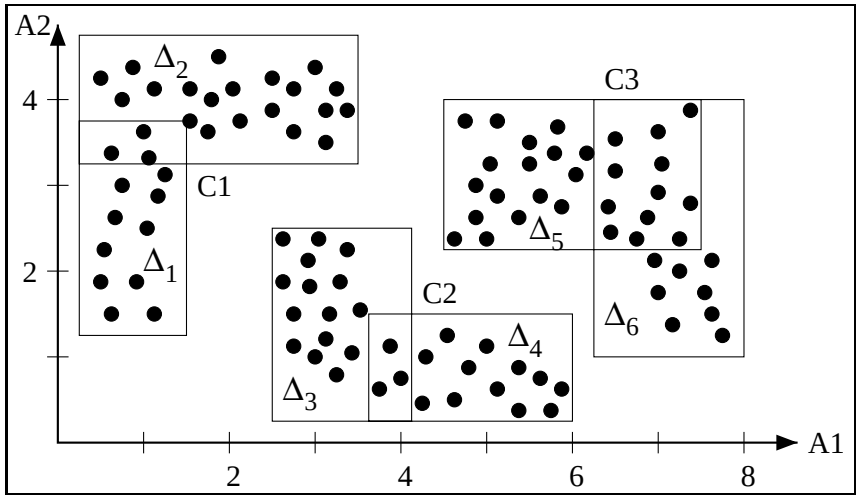


Figure 2.5: **Example: Approximate box decomposition that is a covering of a 3-cluster set in R^2.** Unproblematic overlap between boxes of the same cluster.

If we define the membership rules $r_{\mathcal{B}_p}$ as described in Lemma 2.4.1, then $r_1 := \{r_{\mathcal{B}_1}, r_{\mathcal{B}_2}\}$ (respectively $r_2 := \{r_{\mathcal{B}_3}, r_{\mathcal{B}_4}\}$, $r_3 := \{r_{\mathcal{B}_5}, r_{\mathcal{B}_6}\}$) is a complete membership rule set of cluster C_1 (respectively C_2, C_3). One easily checks that r_1, r_2 and r_3 are minimal.

p	$B_{p,1}$	$B_{p,2}$
1	[0.25, 1.5]	[1.25, 3.75]
2	[0.25, 3.5]	[3.25, 4.75]
3	[2.5, 4.125]	[0.25, 2.5]
4	[3.625, 6]	[0.25, 1.5]
5	[4.25, 7.5]	[2.25, 4]
6	[6.25, 8]	[1, 4]

Table 2.1: **Example: Approximate box decomposition that is a covering of a** 3**-cluster set in** R^2**.**

Instead of Δ we could also use the box decomposition that is shown on the right hand side of Figure 2.2. But note that the approximate box decomposition on the left hand side leads to an incomplete membership rule set for cluster C_3. The uniform box decomposition from Figure 2.3 is also suitable, but the corresponding membership rule sets are not minimal.

2.4.2 Discriminating attributes

Since we are interested in efficient cluster descriptions, we have not only to determine complete membership rule sets, we have also to reduce them as much as possible (see section 1.3). Therefore we have to identify the *discriminating attributes* of the cluster problem, i.e. the attributes that are necessary to determine the cluster membership of each data object.

Let $V = \{v_1, \ldots, v_n\} \subset \Omega$ be any data set in Ω with frequency function f and homogeneity function h. Further let $\mathcal{C} := \{C_1, \ldots, C_k\}$ be any k-cluster set of V and $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ be any covering of $\mathcal{C}$ with non-void disjoint index subsets $I_1, \ldots, I_k$ so that $\bigcup_{s=1}^{k} I_s = \{1, \ldots, n_k\}$ and $C_s = \bigcup_{p \in I_s} \Theta_p$ for $s = 1, \ldots, k$. Remember that for any index subset $J \in \{1, \ldots, q\}$, $v(J)$ denotes the projection of $v \in \Omega$ on $\Omega(V)$, where $\Omega(V)$ is spanned by the attributes A_j with $j \in J$. Remember further that we have defined $M(J) := \{v(J) \,|\, v \in M\}$ for any subset $M \subset \Omega$.

Definition 2.4.2 *Let $J \subset \{1,\ldots,q\}$ be any non-void index subset and denote by $J^c := \{1,\ldots,q\} \setminus I$ its complement.*
(a) We call the attribute set $\mathcal{A}(J^c) := \{A_j \,|\, j \in J^c\}$ redundant for $\mathcal{C}$ if we have:

$$v \in C_s \iff v(J) \in \bigcup_{p \in I_s} \Theta_p(J) \ \text{ forall } v \in V.$$

(b) We call the attribute set $\mathcal{A}(J^c)$ maximally redundant for $\mathcal{C}$ if there exists no subset $\widetilde{J} \subset \{1,\ldots,q\}$ so that $\mathcal{A}(\widetilde{J}^c)$ is redundant for $\mathcal{C}$ and $|J| > |\widetilde{J}|$.
(c) We call attribute A_i an univariate discriminating attribute of $\mathcal{C}$, if $\mathcal{A}(\{j\})$ is not redundant for $\mathcal{C}$.
(d) We call the attributes $A_j \in \mathcal{A}(J)$ multivariate discriminating attributes of $\mathcal{C}$ if $\mathcal{A}(J^c)$ is maximally redundant for $\mathcal{C}$.

The following Lemma is an extension of Lemma 2.4.1:

Lemma 2.4.3 *Suppose there exist any $\Delta := \{\Delta_1,\ldots,\Delta_{n_k}\}$ so that (Θ,Δ) is an approximate box decomposition of V with respect to f. Choose any $s \in \{1,\ldots,k\}$ and any $p \in I_s$. Define $r_{\mathcal{B}_p}$ according to Lemma (2.4.1) and suppose further that $f(\Delta_p \setminus C_s) = 0$, then we have:*
The function $r_{\mathcal{B}_p(J)}$ with $\mathcal{B}_p(J) := \{B_{p,1}(J),\ldots,B_{p,q}(J)\}$ and

$$B_{p,j}(J) := \begin{cases} B_j & \text{if } j \in J \\ A_j & \text{else} \end{cases}, \ \text{for } j \in \{1,\ldots,q\},$$

is a J-reduced membership rule for cluster C_s if $\mathcal{A}(J^c)$ is redundant for $\mathcal{C}$.

Proof: We have

$$f(\Delta_p \setminus C_s) = 0 \iff \Delta_p \cap V \subset C_s = \bigcup_{p \in I_s} \Theta_p \iff \Delta_p(J) \subset \bigcup_{p \in I_s} \Theta_p(J)$$

and therefore

$$r_{\mathcal{B}_p(V)}(v) = 1 \implies v(J) \in \Delta_p(J) \subset \bigcup_{p \in I_s} \Theta_p(J) \iff v \in C_s.$$

□

Analogously to Lemma 2.4.1 one easily checks that $\{r_{\mathcal{B}_p(J)} \,|\, p \in I_s\}$ is a J-reduced complete membership rule set of cluster C_s, if $f(\Delta_p \setminus C_s) = 0$ for all $p \in I_s$ and $C_s \subset \bigcup_{p \in I_s} \Delta_p$. Moreover if $\mathcal{A}(J^c)$ is maximally redundant, $\{r_{\mathcal{B}_p(J)} \,|\, p \in I_s\}$ is optimally reduced.

Discriminating attributes identification algorithm

Suppose that $\mathcal{C}$ is any optimal k-cluster set of (V, f, h) and that there exist any $\Delta := \{\Delta_1, \ldots, \Delta_{n_k}\}$ so that (Θ,Δ) is an approximate box decomposition of V with respect to f. Then the following algorithm can be used to determine the multivariate discriminating attributes of $\mathcal{C}$:
(1) Choose $0 < \delta \ll 1$. Set $J_{opt} := \{1, \ldots, q\}$ and $\delta_{opt} := 0$.
(2) Let $J \subset \{1, \ldots, q\}$ be any index subset of minimal size so that

$$\text{overlap}_f(\hat{\Delta}(J) \leq \text{overlap}_f(\Delta) + \delta,$$

where $\hat{\Delta}(J) := \{\hat{\Delta}_1(J), \ldots, \hat{\Delta}_{n_k}(J)\}$ with

$$\hat{\Delta}_p(J) \subset \Omega \text{ and } v \in \hat{\Delta}_p(J) \iff v(J) \in \Delta_p(J) \text{ for all } v \in \Omega.$$

(3) If $|J| < q$, then goto step (5).
(4) If $\delta_{opt} = 0$, then goto step (7), else stop.
(5) If $\mathcal{A}(J^C)$ is not redundant for $\mathcal{C}$, then decrease δ and goto step (2).
(6) If $|J| < |J_{opt}|$, then set $J_{opt} := J$ and $\delta_{opt} := \delta$, else stop.
(7) If $|J| > 1$, then increase δ and goto step (2), else stop.

For cluster problems with a special type of homogeneity function, that exhibits a stochastic property, in chapter 4 we are going to present a method that allows to proof quickly if $\mathcal{A}(J^C)$ is redundant for $\mathcal{C}$.

Example: Discriminating attributes of cluster problem with unknown number of clusters

If we look again at our simple example from section 1.4, we can easily identify the discriminating attributes corresponding to the optimal k-cluster sets for differently chosen k.

Obviously for the clusterings $\mathcal{C}(1) - \mathcal{C}(4)$ we need for each $v \in V$ only the value for attribute A_1 to determine the cluster membership.

Formally spoken, if we set $J := 1$ and choose $k \in \{1, \ldots, 4\}$, then we have for each cluster $C \in \mathcal{C}(k)$ and for all $v \in V$:

$$v \in C_s \iff v(J) \in C_s(J).$$

Since $\Theta := \mathcal{C}(k)$ is always a trivial covering of k-cluster set $\mathcal{C}(k)$, the attribute set $\mathcal{A}(J^c) = \{A_2\}$ is redundant. Further it is maximally redundant, because it is not possible that a redundant attribute set contains all attributes. Therefore $A_1 \in \mathcal{A}(J)$ is a multivariate discriminating attribute of $\mathcal{C}(k)$, $k = 1, \ldots, 4$.

To illustrate the working of the suggested identification algorithm, we use it to determine the discriminating attributes of $\mathcal{C} := \mathcal{C}(2)$:

- At the beginning we set $\Theta := \mathcal{C}$ and $\Delta := \{\Delta_1, \Delta_2\}$, with boxes $\Delta_1 := B_{1,1} \times B_{1,2} := [0.5, 2] \times [0.5, 3]$ and $\Delta_2 := B_{2,1} \times B_{2,2} := [5.5, 6] \times [1.5, 2.5]$. Then (Θ, Δ) is an approximate box decomposition of V.

- In step (1) we choose a small δ, e.g., $\delta := 0.01$. We set $J_{opt} := \{1, \ldots, q\}$ and $\delta_{opt} := 0$.

- Obviously in step (2) it is enough to investigate $J_1 := \{1\}$ and $J_2 := \{2\}$. Extending the projections $\Delta_s(J_1) := B_{s,1}$ and $\Delta_s(J_2) := B_{s,2}$ we got $\hat{\Delta}_s(J_1) := B_{s,1} \times \mathbf{R}$ and $\hat{\Delta}_s(J_2) := \mathbf{R} \times B_{s,2}$ for $s = 1, 2$. This leads to $\mathrm{overlap}_f(\hat{\Delta}(J)) := 0$ and $\mathrm{overlap}_f(\hat{\Delta}(J_2)) := 0.56$. Since we have $\mathrm{overlap}_f(\Delta) = 0$, we set $J := J_1$.

- At step (3) we have $|J| = 1 < 2 = q$ and therefore we jump to step (5).

- Now we have to prove, if $\mathcal{A}(J^C) = \{A_2\}$ is redundant. This is the case and we go to step (6).

- Since $|J| = 1 < 2 = |J_{opt}|$, we set $J_{opt} := J$ and $\delta_{opt} := \delta$.

- At step (7) we stop, because $|J| = 1$. The result of the algorithm is $J_{opt} := 1$ and determines A_1 as the only multivariate discriminating attribute of $\mathcal{C}$. One easily checks, that δ_{opt} is a kind of quality indicator of the computation. If δ_{opt} is sufficiently small, we can be confident that we have identified the correct multivariate discriminating attributes of clustering $\mathcal{C}$.

Chapter 3

Adaptive Decomposition by Self-Organized Neural Networks

In this chapter we will describe two methods, based on self-organized neural networks [1], that can be used to compute a decomposition $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ of a data set V with homogeneity function h. The decomposition is adaptive in the sense that the number n_k is chosen automatically — only an upper bound $\Bbbk \in \mathbf{N}$ has to be fixed a priori — so that Θ is fine enough to use it within our basic reduction algorithm (see section 2.3). Moreover, the second method that is an recently developed extension of the first one (see [29]), allows to compute non-uniform approximate box decompositions.

Since each decomposition of V is also a kind of clustering of V, the computation of a decomposition with small decomposition error (see Eq. (2.1)) has to be done heuristically in a shorter time than $\mathcal{O}(n^2)$. Otherwise there is no advantage of our basic reduction algorithm in comparison with a direct computation of an optimal k-cluster set of V.

We suppose that $\Omega \subset \mathbf{R}^q$ is a metric space, otherwise we will extend it sufficiently as described in the appendix. Further we assume that there exists a distance function $\mathrm{dist} : \Omega \times \Omega \longrightarrow \mathbf{R}$ so that for all $v, w \in V$ the following local maximum condition holds:

$$\mathrm{dist}(v, w) \approx 0 \implies h(v, w) \approx h_{max}(V) \ . \tag{3.1}$$

Usually, this condition is given for geometric cluster problems and also for many dynamic cluster problems (see the earlier discussion in section 1.2).

[1] For an introduction to neural networks see, e.g., [55]

3.1 Self-Organizing Maps (SOM)

Let V any data set in Ω with frequency function f. The following Lemma describes a way to compute an adaptive decomposition based on a given codebook:

Lemma 3.1.1 *Assume* $\Bbbk \in \mathbf{N}$ *and* $W := \{w_1, \ldots, w_\Bbbk\} \subset \Omega$.
Set $\Theta_W := \{\Theta_{w_1}, \ldots, \Theta_{w_\Bbbk}\}$ *with partitions* $\Theta_{w_p} \subset \Omega$ *so that for all* $v \in \Omega$*:*

$$v \in \Theta_{w_p} \iff p = \min\{s \,|\, \mathrm{dist}(v, w_s) = \min_{i=1,\ldots,\Bbbk} \mathrm{dist}(v, w_i)\}. \tag{3.2}$$

Further set $I := \{p \,|\, \Theta_{w_p}(V) \neq \emptyset, p = 1, \ldots, \Bbbk\}$ *with* $\Theta_{w_p}(V) := \Theta_{w_p} \cap V$. *Then* $\Theta_{W_I}(V) := \{\Theta_{w_p}(V) \,|\, p \in I\}$ *is a decomposition of* V *with* $n_k := |I|$ *partitions.*

Since we have $\mathrm{dist}(v, w) \leq \mathrm{dist}(v, w_s) + dist(w, w_s)$ for all $v, w \in \Theta_{w_s}(V)$, each method that tries to compute a W so that for $v \in \Theta_{w_s}(V)$ the distances $\mathrm{dist}(v, w_s)$ are minimized, can be used to generate a decomposition of V with small decomposition error.

At first, one might think of pure vector quantization (VQ) methods (see [35]). These methods often try to minimize the *distortion value* which is defined as:

$$\frac{1}{f(V)} \sum_{s=1}^{\Bbbk} \sum_{v \in \Theta_{w_s}(V)} \mathrm{dist}(v, w_s) f(v). \tag{3.3}$$

However, they have the tendency to produce codebook vectors that are maximally different, to achieve a more uniform decomposition of V. This might cause problems of so called *pseudo-clusters*, i.e. clusters C with nearly zero frequency value $f(C)$. Therefore it seems better to use a method that tends to gather codebook vectors in some more robust way. Here a powerful method are KOHONEN'S Self-Organizing Maps (SOM). The corresponding algorithm usually produces fast and good solutions even for high-dimensional Ω. It can be easily adapted to the case of cyclic data which will be essential for using it within biomolecular data (see chapter 5). Further it has the feature of topology approximation which avoids the appearance of pseudo-clusters and leads to decompositions that are rather robust under changes of the number $\Bbbk$.

In the following we give a short general description of the SOM method. For an exhaustive presentation see [48].

To be in correspondence with the usual notation in the literature, we suppose that there exists a probability distribution P_ρ on Ω with a probability density function $\rho : \Omega \to \mathbf{R}_0^+$ so that $\rho(v) = \frac{f(v)}{f(V)}$ for $v \in V$. If this is not the case, one has to replace all integral signs by sums and has to use f directly.

Each SOM is formed by a q-dimensional input-layer that is fully connected with the two-dimensional Kohonen layer, which is a neural $m_x \times m_y$ grid G with

rectangular or hexagonal topology and $\mathbb{k} = m_x m_y$ grid neurons. The coordinate tuple of each neuron s on the grid is denoted by $z_s \in G$ and each neuron s is uniquely related to a q-dimensional codebook vector w_s. After a suitable initialization of the codebook vectors, the SOM is trained in L time steps by a repeated presentation of vectors of the q-dimensional input space Ω according to the probability distribution P_ρ. For each presented input vector the SOM computes a so called winner neuron and its neighboring neurons on the grid and adapts the related codebook vectors so that the distance to the input vector is reduced. To achieve convergence, the learning rate of the distance reduction $\alpha : \{0, \ldots, L\} \to [0, 1]$ and the width of the neighborhood of the winner neuron, the so called neighborhood radius function $\gamma : \{0, \ldots, L\} \to \mathbf{R}_0^+$, shrink to zero with time. After a suitable number of training steps the codebook vectors that are related to neighboring neurons on the grid, are neighboring in the input space according to the chosen distance function. Therefore the codebook vectors not only determine via Eq. (3.2) a decomposition of Ω, but also approximate the topology of the input space via the neighborhood structure of the grid.

Algorithmic Realization In the following we describe the initialization of the codebook vectors, the definition of the winner neuron together with its grid neighborhood and the specification of the codebook adaptation rule.

Initialization. We suggest to choose the initial values $w_1(0), \ldots, w_{\mathbb{k}}(0)$ as approximately P_ρ-distributed random vectors with $w_s(0) \in \Omega$.

Winner neuron and grid neighborhood. Let $x = (x_1, \ldots, x_q)^T \in \Omega$ be an any input vector and $w_1, \ldots, w_{\mathbb{k}} \in \Omega$ the actual codebook vectors of the SOM. Then we call neuron $p \in \{1, \ldots, \mathbb{k}\}$ the *winner neuron* for input x, if

$$p = \min\{s \,|\, \mathrm{dist}(x, w_s) = \min_{i=1,\ldots,\mathbb{k}} \mathrm{dist}(x, w_i)\}. \tag{3.4}$$

Note that Eq. (3.4) is equivalent to $x \in \Theta_{w_p}$, if Θ_{w_p} is defined according to Eq. (3.2).

To determine the neighboring neurons of the winner neuron, one has to specify a grid distance function $\eta : G \times G \times \mathbf{R}^+ \to [0, 1]$. Usually one uses either the bubble grid distance

$$\eta_{\mathrm{bubble}}(z_s, z_p, \gamma) := \begin{cases} 0 & \text{if } \|z_s - z_p\| \leq \gamma \\ 1 & \text{else,} \end{cases}$$

or the Gaussian grid distance

$$\eta_{\mathrm{gaussian}}(z_s, z_p, \gamma) := 1 - \exp\left(-\frac{\|z_s - z_p\|^2}{2\gamma^2}\right),$$

where γ denotes the actual neighborhood radius and $\|\cdot\|$ the two-dimensional Euclidean distance. A neuron s belongs to the neighborhood of winner neuron p if $\eta(z_s, z_p, \gamma) < 1$. If we choose $\eta_{gaussian}$, then the neighborhood of each neuron covers *all* grid neurons.

Codebook adaptation rules. Let neuron p be the winner neuron for input $x(t) = (x_1(t), \ldots, x_q(t))^T \in \Omega$ at time t and $w_1(t), \ldots, w_{\mathbb{k}}(t) \in \Omega$ the actual codebook vectors. Further let $\alpha(t)$ and $\gamma(t)$ be two time-dependent linear or log-linear functions that decrease to zero with $\alpha(0) \leq 1$ and $\gamma(0) \leq \frac{\min\{m_x, m_y\}}{2}$.
Then the new codebook vectors $w_1(t+1), \ldots, w_{\mathbb{k}}(t+1)$ are computed as

$$w_s(t+1) := w_s(t) + \alpha(t)\,\mathrm{neigh}(z_s, z_p, t)\,(x(t) - w_s(t)) \tag{3.5}$$

with $\mathrm{neigh}(\mathrm{z_s}, \mathrm{z_p}, \mathrm{t}) := 1 - \eta(\mathrm{z_s}, \mathrm{z_p}, \gamma(\mathrm{t}))$.

In the case that we set $\gamma(0) = 0$, the SOM is a pure VQ algorithm and therefore optimizes the distortion value [48]. If we allow neighborhood learning, e.g., $\gamma(0) > 0$, the formulation of an energy function that is minimized by the SOM is not possible [47]. Recently slight modifications of the adaptation rules have been suggested that allows the formulation of an energy function without destroying the essential features of the SOM [38, 40]. For further theoretical investigations of the SOM algorithm, especially a comparison to pure VQ methods, see [54, 11, 12].

3.2 Self-Organizing Box Maps (SOBM)

The basic idea of the recently developed Self-Organizing Box Maps (SOBM) method [29] is to compute *codebook boxes* $\hat{W}_s := (\hat{W}_{s_1}, \ldots, \hat{W}_{s_q}) \in \mathrm{BOX}(\Omega)$ with $\hat{W}_{s_i} = [l_{s_i}, r_{s_i}] \subset \mathbf{R}$ instead of codebook vectors $w_s \in \Omega$. This is done in such a way that each codebook box is a nearly optimal box approximation of its corresponding partition $\Theta_{\hat{W}_s} \subset \Omega$:

We will call any set $B = \bigotimes_{i=1}^{q}[l_i, r_i] \in \mathrm{BOX}(\Omega)$ with $l_i, r_i \in \mathbf{R}$ an optimal box approximation of a set $M \subset \Omega$ with respect to P_ρ, if

$$P_\rho(B \setminus M) + P_\rho(M \setminus B) \;\rightarrow\; \min\,.$$

Algorithmic Realization Obviously, this change of concept induces changes of the SOM algorithm, which we arrange here:

Initialization. Let $w_1(0), \ldots, w_{\mathbb{k}}(0)$ be different initial values for the codebook vectors of the traditional SOM, e.g., approximately P_ρ-distributed random vectors with $w_s(0) \in \Omega$ for $s = 1, \ldots \mathbb{k}$. For our extended algorithm, we choose $\hat{W}_s(0) := \bigotimes_{i=1}^{q}[l_{s_i}(0), r_{s_i}(0)]$ with $l_{s_i}(0) = W_{s_i}(0)$ and

$r_{s_i}(0) = W_{s_i}(0) + \epsilon$ in terms of a small positive value ϵ, the initial width of the interval so that $\hat{W}_s \cap \hat{W}_p = \emptyset$ for all $s, p \in \{1, \ldots, \Bbbk\}$.

Winner neuron. We suppose that the problem specific q-dimensional distance function $\mathrm{dist}(x, y)$ with $x, y \in \Omega$ can be written as a function F of q one-dimensional distance measures $d_i(x_i, y_i)$, which means that $\mathrm{dist}(x, y) = F(d_1(x_1, y_1), \ldots, d_q(x_q, y_q))$. Note that many popular distance measures, as e.g., the Euclidean distance, just exhibit this feature. Obviously we need a distance measure DIST that permits to compute the distance between an input vector $x \in \Omega$ and codebook boxes $\hat{W}_s \in \mathrm{BOX}(\Omega)$. For that purpose, we suggest

$$\mathrm{DIST}(x, \hat{W}_s) := F(\hat{d}_1(x_1, \hat{W}_{s_1}), \ldots, \hat{d}_q(x_q, \hat{W}_{s_q}))$$

with

$$\hat{d}_i(x_i, \hat{W}_{s_i}) := \begin{cases} 0 & \text{if } x_i \in \hat{W}_{s_i} \\ \min\{d_i(x_i, l_{s_i}), d_i(x_i, r_{s_i})\} & \text{else.} \end{cases}$$

Then the winner neuron p has to match a condition analogous to Eq. (3.4):

$$p = \min\{s \,|\, \mathrm{DIST}(x, \hat{W}_s) = \min_{i=1,\ldots,\Bbbk} \mathrm{DIST}(x, \hat{W}_i)\}. \tag{3.6}$$

Obviously we can use Eq. (3.6) to define for each codebook box $\hat{W}_s$ the corresponding partition $\hat{\Theta}_s := \Theta_{\hat{W}_s} \subset \Omega$ analogously to Eq. (3.2).

Codebook adaptation rules. In analogy to the SOM algorithm, the SOBM algorithm has to adapt the codebook *boxes*. This will be done by the following rules:

$$\begin{aligned} l_{s_i}(t+1) \;:=\; & l_{s_i}(t) \\ & + g(l_{s_i}(t), r_{s_i}(t), x_i(t))\ \alpha(t)\, \mathrm{neigh}(z_s, z_p, t)\, (x_i(t) - l_{s_i}(t)) \\ & - \alpha(t)\ c(l_{s_i}(t), r_{s_i}(t)) \end{aligned}$$

$$\begin{aligned} r_{s_i}(t+1) \;:=\; & r_{s_i}(t) \\ & + g(-r_{s_i}(t), -l_{s_i}(t), -x_i(t))\ \alpha(t)\, \mathrm{neigh}(z_s, z_p, t)\, (x_i(t) - r_{s_i}(t)) \\ & + \alpha(t)\ c(l_{s_i}(t), r_{s_i}(t)) \end{aligned}$$

with a linear function $g : \mathbf{R}^3 \to [0, 1]$, $g(a, b, x) := \begin{cases} 1 & \text{if } x < a \\ 0 & \text{if } x > b \\ \frac{b-x}{b-a} & \text{else} \end{cases}$

and a function $c : \mathbf{R}^2 \to \mathbf{R}_0^+$ that is independent of the input $x(t)$ and will be defined later.

Note that instead of the above function g, also a smoother "sigmoid" function like $\overline{g}(a, b, x) := 1 - \frac{1}{1+\exp(-x+\frac{a+b}{2})}$ can be chosen in principle.

Suppose for the time being that $c = 0$, then one easily verifies that the left interval boundary is only adapted, if the input is left of the right interval boundary and vice versa. Further one observes that inputs outside the interval have a greater influence on the adaptation of the nearest interval boundary, as when they are inside the interval. In the following we will motivate the suggested adaptation rule.

One easily verifies, that after the initialization we have $\hat{W}_s \subset \hat{\Theta}_s$ for all $s = 1, \ldots, k$. Suppose now an input x that belongs to $\hat{\Theta}_s$. If $x_i \notin \hat{W}_{s_i}$, we have to widen the interval. Therefore the nearest interval boundary is "pulled" towards x_i. This is just the same method as in the original SOM algorithm. If $x_i \in \hat{W}_{s_i}$ the first strategy is to do nothing, because in this case the box seems to be all right. This however, turns out to be not a good idea, because the $\hat{\Theta}_s$ change over time so that we can observe $\hat{W}_s \setminus \hat{\Theta}_s \neq \emptyset$ after several adaptation steps. If this difference becomes larger, it is not only possible that $P_\rho(\hat{W}_s \setminus \hat{\Theta}_s)$ increases so that $\hat{W}_s$ is no longer a good box approximation of $\hat{\Theta}_s$. Also the probability grows that one observes overlaps between the boxes after the algorithm stops (see Figure 3.1).

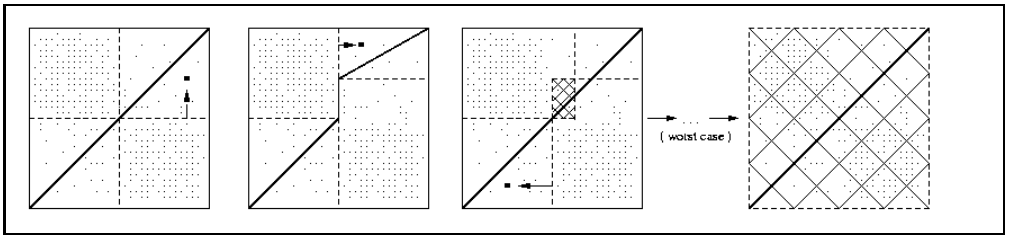


Figure 3.1: **Poor partitioning in the absence of interval shrinkage.**

If, however the overlap between the boxes is too large, $\hat{W}$ and its corresponding decomposition are no longer an approximate box decomposition. Therefore it is necessary to shrink the intervals. This could be done by adapting the interval boundaries when even the input x_i is inside the interval, the so called interior adaptation. It is obvious that the adaptation of the nearest boundary should be greater than that of the opposite side. By doing this a new problem arises: Usually after some time there are more inputs x_i inside the interval than outside. As a consequence, the interval shrinks faster than it grows, which implies that the value $P_\rho(\hat{W}_s)$ shrinks, too. But then the box approximation of $\hat{\Theta}_s$ is not as good as it

could be. Therefore one has to introduce something like a damping coefficient or a correction term, which reduces the inter-interval adaptation. Such a parameter will depend on the ratio of the inputs inside and outside the interval. A direct computation would be impracticable, because it is very time consuming. So one has to think about certain heuristics, which only consider the interval width. Our approaches with a damping coefficient, appeared to supply unsatisfactory results. Excellent results were obtained by another approach, which uses an analytically derived correction term. This approach will be presented subsequently.

Correction term

Without loss of generality, we suppose that there exist $a_i, b_i \in \mathbf{R}$ so that we have $\Omega_\rho := \{x \in \Omega |\, \rho(x) > 0\} \subset \bigotimes_{i=1}^{q} [a_i, b_i]$. Let $\hat{\Theta}_s(t)$ be the decomposition that is defined via $\hat{W}_s(t)$ and let $\Delta_s(t) := \bigotimes_{i=1}^{q} [l^*_{s_i}(t), r^*_{s_i}(t)]$ be an optimal box approximation of $\hat{\Theta}_s(t)$ with minimal volume, i.e.

$$\mathrm{boxvol}(\Delta_s(t)) := \prod_{i=1}^{q} (r^*_{s_i}(t) - l^*_{s_i}(t)) \;\rightarrow\; \min .$$

For our further expositions we define for $M \subset \Omega$ with $P_\rho(M) > 0$, the conditional probability density function ρ_M on M via

$$\rho_M(\omega) \;:=\; \begin{cases} \frac{\rho(\omega)}{P_\rho(M)} & \text{if } \omega \in M \\ 0 & \text{else.} \end{cases}$$

Using $\rho_{\hat{\Theta}_s(t)}$, we can compute the conditional expectation value $E(\hat{W}_s(t+1))$ for each actual codebook vector $\hat{W}_s(t)$ under the condition that s is the winner neuron. Note that this implicitly ensures $P_\rho(\hat{\Theta}_s(t)) > 0$.

We have $E(\hat{W}_s(t+1)) = \bigotimes_{i=1}^{q} [E(l_{s_i}(t+1)), E(r_{s_i}(t+1))]$ with

$$\begin{aligned} E(l_{s_i}(t+1)) \;&:=\; \int_{\Omega_\rho} l_{s_i}(t+1) \rho_{\hat{\Theta}_s(t)}(X)\, dx = \int_{a_i}^{b_i} l_{s_i}(t+1) \rho_{\hat{\Theta}_s(t),i}(x_i)\, dx_i \,, \\ E(r_{s_i}(t+1)) \;&:=\; \int_{\Omega_\rho} r_{s_i}(t+1) \rho_{\hat{\Theta}_s(t)}(X)\, dx = \int_{a_i}^{b_i} r_{s_i}(t+1) \rho_{\hat{\Theta}_s(t),i}(x_i)\, dx_i \end{aligned}$$

and

$$\rho_{\hat{\Theta}_{s(t),i}}(x_i) \quad :=$$

$$\int_{a_1}^{b_1} \cdots \int_{a_{i-1}}^{b_{i-1}} \int_{a_{i+1}}^{b_{i+1}} \cdots \int_{a_q}^{b_q} \rho_{\hat{\Theta}_{s(t)}}((x_1, \ldots, x_q)^T)\, dx_1 \ldots dx_{i-1}\, dx_{i+1} \ldots dx_q\,.$$

Upon considering our above adaptation rule we obtain:

$$\begin{aligned} E(l_{s_i}(t+1)) \;=\; & l_{s_i}(t) \\ & + \int_{a_i}^{l_{s_i}(t)} \alpha(t)(x_i - l_{s_i}(t))\rho_{\hat{\Theta}_{s(t),i}}(x_i)\, dx_i \\ & + \int_{l_{s_i}(t)}^{r_{s_i}(t)} \frac{(r_{s_i}(t) - x_i)}{(r_{s_i}(t) - l_{s_i}(t))}\alpha(t)(x_i - l_{s_i}(t))\rho_{\hat{\Theta}_{s(t),i}}(x_i)\, dx_i \\ & - \alpha(t)\, c(l_{s_i}(t), r_{s_i}(t)) \end{aligned}$$

and

$$\begin{aligned} E(r_{s_i}(t+1)) \;=\; & r_{s_i}(t) \\ & + \int_{r_{s_i}(t)}^{b_i} \alpha(t)(x_i - r_{s_i}(t))\rho_{\hat{\Theta}_{s(t),i}}(x_i)\, dx_i \\ & + \int_{l_{s_i}(t)}^{r_{s_i}(t)} \frac{(x_i - l_{s_i}(t))}{(r_{s_i}(t) - l_{s_i}(t))}\alpha(t)(x_i - r_{s_i}(t))\rho_{\hat{\Theta}_{s(t),i}}(x_i)\, dx_i \\ & + \alpha(t)\, c(l_{s_i}(t), r_{s_i}(t))\,. \end{aligned}$$

Since $\Delta_s(t)$ is an optimal box approximation of $\hat{\Theta}_s(t)$, we may assume that

$$P_{\rho_{\hat{\Theta}_{s(t)}}}(\Delta_s(t)) = \int_{\omega \in \Delta_s(t)} \rho_{\hat{\Theta}_{s(t)}}(\omega)\, d\omega \approx 1.$$

Therefore, for simplicity, we suppose that the i-th components x_i of the inputs $X \in \hat{\Theta}_s(t)$ are uniformly distributed over $[l_i^*(t), r_i^*(t)]$ so that

$$\rho_{\hat{\Theta}_{s(t),i}}(x_i) \;:=\; \begin{cases} \frac{1}{r_i^*(t) - l_i^*(t)} & \text{if } X = (x_1, \ldots, x_q)^T \in \Delta_s(t) \\ 0 & \text{else}\,. \end{cases}$$

Hence, we arrive at:

$$\begin{aligned}
E(r_{s_i}(t+1)) &= r_{s_i}(t) \\
&\quad + \int_{r_{s_i}(t)}^{r_i^*(t)} \frac{\alpha(t)}{(r_i^*(t) - l_i^*(t))}(x_i - r_{s_i}(t))\, dx_i \\
&\quad + \int_{l_{s_i}(t)}^{r_{s_i}(t)} \frac{(x_i - l_{s_i}(t))}{(r_{s_i}(t) - l_{s_i}(t))} \frac{\alpha(t)}{(r_i^*(t) - l_i^*(t))}(x_i - r_{s_i}(t))\, dx_i \\
&\quad + \alpha(t)\, c(l_{s_i}(t), r_{s_i}(t)) \\
&= r_{s_i}(t) \\
&\quad + \frac{\alpha(t)}{(r_i^*(t) - l_i^*(t))} \frac{(r_i^*(t) - r_{s_i}(t))^2}{2} \\
&\quad + \frac{\alpha(t)}{(r_i^*(t) - l_i^*(t))} \int_{l_{s_i}(t)}^{r_{s_i}(t)} \frac{(x_i - l_{s_i}(t))(x_i - r_{s_i}(t))}{(r_{s_i}(t) - l_{s_i}(t))}\, dx_i \\
&\quad + \alpha(t)\, c(l_{s_i}(t), r_{s_i}(t)) \\
&= r_{s_i}(t) \\
&\quad + \frac{\alpha(t)}{(r_i^*(t) - l_i^*(t))} \frac{(r_i^*(t) - r_{s_i}(t))^2}{2} \\
&\quad - \frac{\alpha(t)}{(r_i^*(t) - l_i^*(t))} \frac{(r_{s_i}(t) - l_{s_i}(t))^2}{6} \\
&\quad + \alpha(t)\, c(l_{s_i}(t), r_{s_i}(t))\,.
\end{aligned}$$

For the left hand boundary, we analogously obtain:

$$\begin{aligned}
E(l_{s_i}(t+1)) &= l_{s_i} \\
&\quad - \frac{\alpha(t)}{(r_i^*(t) - l_i^*(t))} \frac{(l_{s_i}(t) - l_i^*(t))^2}{2} \\
&\quad + \frac{\alpha(t)}{(r_i^*(t) - l_i^*(t))} \frac{(r_{s_i}(t) - l_{s_i}(t))^2}{6} \\
&\quad - \alpha(t)\, c(l_{s_i}(t), r_{s_i}(t))\,.
\end{aligned}$$

By means of the intuitive choice

$$c(l_{s_i}(t), r_{s_i}(t)) := \frac{1}{6}\left(r_{s_i}(t) - l_{s_i}(t)\right) \tag{3.7}$$

we end up with

$$\begin{aligned} E(l_{s_i}(t+1)) &= l_{s_i} - \frac{1}{2}\alpha(t)\frac{(l_{s_i}(t) - l_i^*(t))^2}{(r_i^*(t) - l_i^*(t))} \\ &\quad - \alpha(t)\left(1 - \psi_{s_i}(t)\right) c(l_{s_i}(t), r_{s_i}(t)) \end{aligned}$$

and

$$\begin{aligned} E(r_{s_i}(t+1)) &= r_{s_i} + \frac{1}{2}\alpha(t)\frac{(r_i^*(t) - r_{s_i}(t))^2}{(r_i^*(t) - l_i^*(t))} \\ &\quad + \alpha(t)\left(1 - \psi_{s_i}(t)\right) c(l_{s_i}(t), r_{s_i}(t)) \end{aligned}$$

in terms of some model quantity

$$\psi_{s_i}(t) := \frac{(r_{s_i}(t) - l_{s_i}(t))}{(r_i^*(t) - l_i^*(t))}\,.$$

This quantity measures the deviation of the actual interval width from the optimal one.

In the following, we have to assure that the intervals are always well defined, i.e. we always have $l_{s_i}(t) < r_{s_i}(t)$ for all $t \in \{0, \ldots, L\}$.

Lemma 3.2.1 *For any $s \in \{1, \ldots, q\}$ and all $t \in \{0, \ldots, L\}$ we have*

$$l_{s_i}(t) < r_{s_i}(t) \implies l_{s_i}(t+1) < r_{s_i}(t+1).$$

Proof: Let p be the winner neuron for input X(t). Then one easily verifies:

$$
\begin{aligned}
(1) \qquad x_i(t) < l_{s_i}(t) \qquad &\Longrightarrow \\
r_{s_i}(t+1) - l_{s_i}(t+1) \quad &= \quad \left(1 + \frac{\alpha(t)}{3}\right)(r_{s_i}(t) - l_{s_i}(t)) \\
&\quad - \underbrace{\underbrace{\alpha(t)}_{\geq 0}\ \underbrace{\mathrm{neigh}(z_s, z_p, t)}_{\geq 0}\ \underbrace{(x_i(t) - l_{s_i}(t))}_{<0}}_{\leq 0} \\
&\geq \quad r_{s_i}(t) - l_{s_i}(t)
\end{aligned}
$$

$$
\begin{aligned}
(2) \qquad x_i(t) > r_{s_i}(t) \qquad &\Longrightarrow \\
r_{s_i}(t+1) - l_{s_i}(t+1) \quad &= \quad \left(1 + \frac{\alpha(t)}{3}\right)(r_{s_i}(t) - l_{s_i}(t)) \\
&\quad + \underbrace{\alpha(t)\,\mathrm{neigh}(z_s, z_p, t)\,(x_i(t) - r_{s_i}(t))}_{\geq 0} \\
&\geq \quad r_{s_i}(t) - l_{s_i}(t)
\end{aligned}
$$

$$
\begin{aligned}
(3) \quad x_i(t) \in [l_{s_i}(t), r_{s_i}(t)] \quad &\Longrightarrow \\
r_{s_i}(t+1) - l_{s_i}(t+1) \quad &= \quad \left(1 + \frac{\alpha(t)}{3}\right)(r_{s_i}(t) - l_{s_i}(t)) \\
&\quad + \alpha(t)\,\mathrm{neigh}(z_s, z_p, t)\,\frac{(x_i(t) - l_{s_i}(t))}{(r_{s_i}(t) - l_{s_i}(t))}(x_i(t) - r_{s_i}(t)) \\
&\quad - \alpha(t)\,\mathrm{neigh}(z_s, z_p, t)\,\frac{(r_{s_i}(t) - x_i(t))}{(r_{s_i}(t) - l_{s_i}(t))}(x_i(t) - l_{s_i}(t)) \\
&= \quad \left(1 + \frac{\alpha(t)}{3}\right)(r_{s_i}(t) - l_{s_i}(t)) \\
&\quad - 2\,\alpha(t)\,\underbrace{\mathrm{neigh}(z_s, z_p, t)}_{\leq 1}\,\underbrace{\frac{(r_{s_i}(t) - x_i(t))(x_i(t) - l_{s_i}(t))}{(r_{s_i}(t) - l_{s_i}(t))}}_{\leq \frac{1}{4}(r_{s_i}(t) - l_{s_i}(t))\ (*)} \\
&\geq \quad \left(1 + \frac{\alpha(t)}{3} - \frac{\alpha(t)}{2}\right)(r_{s_i}(t) - l_{s_i}(t)) \\
&= \quad \left(1 - \frac{\alpha(t)}{6}\right)(r_{s_i}(t) - l_{s_i}(t))
\end{aligned}
$$

$$
(*) \qquad \max_{l \leq x \leq r}(r - x)(x - l) \quad = \quad \frac{1}{4}(r - l)^2 \quad \text{for all } l, r \in \mathbf{R}
$$

Because $\alpha(t) \leq 1$ for all $t \in \{0, \ldots, L\}$, we have in all three cases:

$$(r_{s_i}(t) - l_{s_i}(t)) > 0 \quad \Longrightarrow \quad (r_{s_i}(t+1) - l_{s_i}(t+1)) > 0.$$

□

Note that Lemma 3.2.1 is usually not true if $\alpha(t) \geq 6$.

Hence if $l_{s_i}(0) < r_{s_i}(0)$, Lemma 3.2.1 guarantees that $c(l_{s_i}(t), r_{s_i}(t)) > 0$ and $\psi_{s_i}(t) > 0$ for all $t \in \{0, \ldots, L\}$.

Therefore we obtain

$$\begin{aligned} \hat{W}_{s_i}(t) \subset [l_i^*(t), r_i^*(t)] \quad &\Longrightarrow \quad \psi_{s_i}(t) \in]0, 1] \\ &\Longrightarrow \quad E(l_{s_i}(t+1)) < l_{s_i}(t) \text{ and } E(r_{s_i}(t+1)) > r_{s_i}(t) \end{aligned}$$

and

$$\hat{W}_{s_i}(t) = [l_i^*(t), r_i^*(t)] \quad \Longrightarrow \quad E(l_{s_i}(t+1)) = l_{s_i}(t) \text{ and } E(r_{s_i}(t+1)) = r_{s_i}(t).$$

If we choose $\hat{W}_s(0) \in \Delta_s(0)$ we can be confident that $\psi_{s_i}(L) \approx 1$ and therefore $\hat{W}_{s_i}(L) \approx [l_i^*(L), r_i^*(L)]$, whenever we use our extended algorithm with L time steps and L large enough. This means that $\hat{W}_s(L) \approx \Delta_s(L)$ and therefore $\hat{W}_s(L)$ is nearly an optimal box approximation of $\hat{\Theta}_s(L)$ with respect to ρ. Obviously the chosen function c is a suitable correction term for the interval shrinkage.

Using this correction term the presented SOBM algorithm is suitable to generate approximate box decompositions of V (see Definition 2.2.1):

Lemma 3.2.2 *Assume $\hat{W} := \{\hat{W}_1, \ldots, \hat{W}_{\Bbbk}\} \subset \Omega$ so that $\hat{W}_p \in \mathrm{BOX}(\Omega)$ is a nearly optimal box approximation of $\Theta_{\hat{W}_p}$ for $p = 1 \ldots, \Bbbk$. Set $\Theta_{\hat{W}_p}(V) := \Theta_{\hat{W}_p} \cap V$. Then $\Theta_{\hat{W}_I}(V) := \{\Theta_{\hat{W}_{p(V)}} \,|\, p \in I\}$ with $I := \{p \,|\, \Theta_{\hat{W}_p}(V) \neq \emptyset\}$ is a decomposition of V with $n_k := |I| \leq \Bbbk$ partitions and $(\Theta_{\hat{W}_I}(V), \hat{W}_I)$ with $\hat{W}_I := \{\hat{W}_p \,|\, p \in I\}$ is an approximate box decomposition.*

Proof: There exists a small $\delta > 0$ so that for any $p \in I$ we have

$$f(\hat{W}_p \setminus \Theta_{\hat{W}_p}) + f(\Theta_{\hat{W}_p} \setminus \hat{W}_p) < \delta f(V).$$

This guarantees $f(\Theta_{\hat{W}_p} \cap \hat{W}_p) > f(\Theta_{\hat{W}_p}) - \delta f(V)$ for any $p \in I$. Since $\Theta_{\hat{W}_I}(V)$ is a decomposition of V by construction and $f(M \cap V) = f(M)$ for any subset M of Ω, this yields:

$$\mathrm{overlay}_f(\Theta_{\hat{W}_I}(V), \hat{W}_I) > 1 - \delta n_k.$$

One easily verifies that for any $p \in I$, we have

$$f(\hat{W}_p \cap \bigcup_{\widetilde{p} \neq p} \hat{W}_{\widetilde{p}}) \leq \sum_{s \in I} f(\hat{W}_s \setminus \Theta_{\hat{W}_s}) = \sum_{s \in I} f(\hat{W}_s) - \sum_{s \in I} f(\hat{W}_s \cap \Theta_{\hat{W}_s})$$

Since $\sum_{s \in I} f(\hat{W}_s) \leq f(V)$, this yields:

$$\begin{aligned} \text{overlap}_f(\hat{W}_I) &\leq \sum_{s \in I} \left(1 - \frac{\sum_{s \in I} f(\hat{W}_s \cap \Theta_{\hat{W}_s})}{\sum_{s \in I} f(\hat{W}_s)}\right) \\ &\leq \sum_{s \in I} \left(1 - \text{overlay}_f(\Theta_{\hat{W}_I}(V), \hat{W}_I)\right) < \delta n_k^2. \end{aligned}$$

□

3.3 Comparison SOM - SOBM

Upon comparing codebooks W and $\hat{W}$, computed by the original SOM and the SOBM algorithm with the same parameters and initialization, one will observe clear similarities. In most cases the orientation of the maps and the visually identifiable clusters are equal (see subsection 5.2.2 for an example).

For each codebook vector $w_p \in W$ one can usually find a codebook box $\hat{W}_s \in \hat{W}$ with $w_p \in \hat{W}_s$. Therefore the SOBM algorithm will be at least as powerful as the classical SOM algorithm. In the following, however, we will show that the SOBM algorithm has important advantages.

For simplicity we suppose that we have only an one-dimensional input space $\Omega = \mathbf{R}$. We want to compute a 2×1 map with neurons s and $\overline{s}$, the Euclidean distance function and $\text{neigh}(z_s, z_{\overline{s}}, t) = 0$ for $t \in \{0, \ldots, L\}$.

For the purpose of illustration, we define two probability density functions ρ_1 and ρ_2 (see Figure 3.2):

$$\rho_1(x) := \begin{cases} 2.5 & \text{if } x \in [0.8,\, 1] \\ 0.5 & \text{if } x \in [-1,\, 0] \\ 0 & \text{else} \end{cases}$$

$$\rho_2(x) := \begin{cases} 2.5 & \text{if } x \in [0.8,\, 1] \\ 0.625 & \text{if } x \in [-1,\, -0.6] \\ 0.625 & \text{if } x \in [-0.4,\, 0] \\ 0 & \text{else.} \end{cases}$$

We have used the original SOM algorithm and our extended algorithm with $c = 0$ and c as defined in Eq. (3.7) to compute the codebooks for ρ_1 and ρ_2.

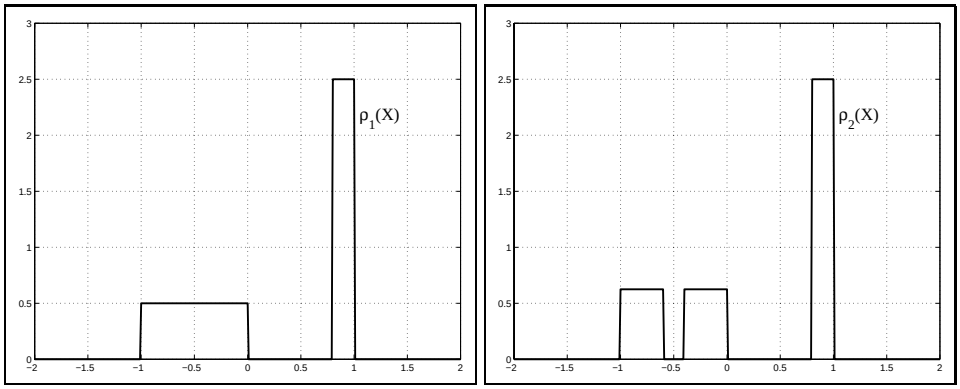


Figure 3.2: **Probability density functions ρ_1 and ρ_2**

Table 3.1 shows the results (random codebook initialization, $\alpha(0) = 0.9$ and $L = 10000$).

	ρ_1
SOM	$w_s = -0.5,\ w_{\overline{s}} = 0.9$
SOBM($c = 0$)	$\hat{W}_s = [-0.75, -0.25],\ \hat{W}_{\overline{s}} = [0.85, 0.95]$
SOBM	$\hat{W}_s = [-1.00, 0.00], \hat{W}_{\overline{s}} = [0.80, 1.00]$
	ρ_2
SOM	$w_s = -0.5, w_{\overline{s}} = 0.9$
SOBM($c = 0$)	$\hat{W}_s = [-0.78, -0.22],\ \hat{W}_{\overline{s}} = [0.85, 0.95]$
SOBM	$\hat{W}_s = [-1.05, 0.07], \hat{W}_{\overline{s}} = [0.80, 1.01]$

Table 3.1: **Codebooks for ρ_1 and ρ_2**

Obviously, the following three observations are of interest:

- The probability density function ρ_1 is positive on $[-1, 0]$ and $[0.8, 1]$. Although these intervals are of different width, we get no hint about this fact, if we look at the codebook vectors w_s and $w_{\overline{s}}$.
- The codebook boxes are box approximations of the partitions, which they implicitly define. These approximations are perfect if we use the correction term c as defined in Eq. (3.7).
- The point codebooks are equal for both probability density functions, i.e. although ρ_1 and ρ_2 are different, we cannot distinguish them by looking at

the codebook vectors. The situation is quite different if we use the correction term c and look at the codebook boxes. Here we see that the interval width of $\hat{W}_s$ in the case of ρ_2 is larger then in the case of ρ_1. If we look deeper, we see that the difference is approximately the width of the hole between -0.4 and -0.6 of ρ_2. This is not surprising, because the correction terms for $\hat{W}_s$ are equal in both cases, but the power of the interval shrinkage for $\hat{W}_s$ is lower in the case of ρ_2. Therefore the interval $\hat{W}_s$ can grow stronger in this case. Although we cannot derive the differences between ρ_1 and ρ_2 from looking at the different $\hat{W}_s$, we at least get a hint that there are differences.

We have made similar observations for higher-dimensional input spaces and larger maps.

Additionally we want to show an intriguing feature of the SOBM algorithm. Look at he following probability density functions ρ_3:

$$\rho_3(x) := \frac{1}{2\sigma\sqrt{2\pi}} \left(\exp(-\frac{1}{2}\left(\frac{(x-\mu_1)}{\sigma}\right)^2 + \exp(-\frac{1}{2}\left(\frac{(x-\mu_2)}{\sigma}\right)^2 \right).$$

One observes that $\hat{W}_s \approx [\mu_1 - \sigma, \mu_1 + \sigma]$ and $\hat{W}_{\overline{s}} \approx [\mu_2 - \sigma, \mu_2 + \sigma]$. The approximation is the better, the larger the difference is between μ_1 and μ_2. Figure 3.3 shows ρ_3 with $\mu_1 = -0.5$, $\mu_2 = 0.5$ and $\sigma = 0.27$ and Table 3.2 gives the corresponding computational results.

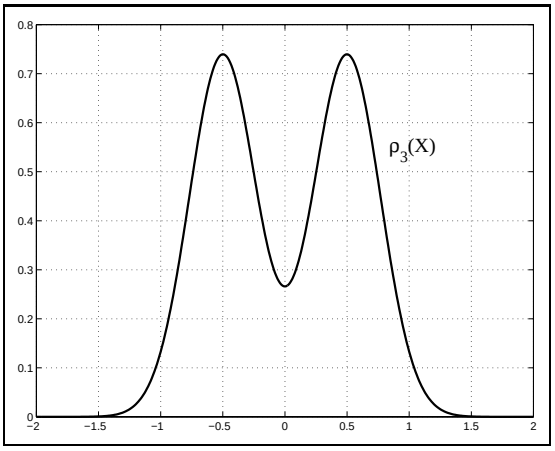


Figure 3.3: **Probability density functions** ρ_3

	ρ_3
SOM	$w_s = -0.5,\ w_{\overline{s}} = 0.5$
SOBM($c = 0$)	$\hat{W}_s = [-0.67, -0.33], \hat{W}_{\overline{s}} = [0.31, 0.67]$
SOBM	$\hat{W}_s = [-0.85, -0.15],\ \hat{W}_{\overline{s}} = [0.13, 0.86]$

Table 3.2: **Codebook vectors for** ρ_3

Although the concept of codebook boxes develops its full power still within the computation of approximate box decompositions the advantages in comparison with point codebooks are already obvious.

A disadvantage of the SOBM algorithm is that it requires more computing time than the classical SOM algorithm. Although the difference depends on the chosen implementation, one easily checks that the number of variables that have to be adapted and to be evaluated are doubled. Therefore in the worst case the SOBM algorithm doubles the computing time of the original SOM algorithm.

3.4 Computational complexity

To speed up the computing time, one may think about a combination of the SOM and the SOBM algorithm. In the following we suggest such a combination, which has turned out to be quiet powerful in our first applications (see chapter 5).

As usual in the original SOM algorithm, we first compute in $L_1 := u \cdot \mathbb{k}$ steps the codebook vectors $w_1, \ldots, w_{\mathbb{k}}$ with a suitable average number of codebook updates u, e.g., $u = 100$, a large learning rate at the beginning, e.g., $\alpha(0) = 1$, and with neighborhood adaptation, i.e. $\mathrm{neigh}(z_s, z_p, t) > 0$ for $t < L_1$. This is often called the *ordering phase* of the SOM algorithm.

After this ordering phase one usually passes on to another adaptation cycle with $L_2 \geq L_1$ adaptation steps, a low learning rate α and no neighborhood adaptation, i.e. $\mathrm{neigh}(z_s, z_p, t) = 0$ for $s \neq p$ and $t \in [L_1, L_1 + L_2]$. After this so called *convergence phase* of the SOM algorithm, the codebook vectors are rather stable and good representatives of the input space and the used probability distribution.

To achieve convergence, in the classical SOM algorithm L_2 is usually much larger than L_1, e.g., a factor 3 or more. In our combined approach, we set $L_2 \approx L_1$ and use the SOBM algorithm for an additional convergence phase: We first ini-

tialize the codebook boxes $\hat{W}_s(0)$ by using the earlier computed representatives $w_s(L_2)$ within the described initialization routine. Then we adapt the codebook boxes in $L_3 \approx L_2$ time steps with a low learning rate and no neighborhood adaptation.

Summarizing, as a result of this combination — original SOM algorithm plus additional convergence phase with SOBM algorithm — we obtain a shorter computing time, as if we only use the SOBM algorithm, while getting comparable results. Additionally we avoid possible negative effects of the neighborhood adaptation on the generation of the codebook boxes.

Up to now, we have not answered the question, if the SOM/SOBM algorithm needs less than $\mathcal{O}(n^2)$ operations to compute a decomposition of any data set $V \subset \Omega$ with n data objects and dimension q.

Let u denote the average number of codebook updates that is sufficient to guarantee convergence of the SOM/SOBM algorithm, i.e. we need $\mathcal{O}(u \cdot \Bbbk)$ adaptation steps. Since we have to compute the winner neuron and to adapt the codebook within each adaptation step, each of these steps costs $\mathcal{O}(q \cdot \Bbbk)$ operations. Therefore we need $\mathcal{O}(u \cdot q \cdot \Bbbk^2)$ operations to generate a suitable codebook. In addition, the computation of a decomposition based on this codebook according to Eq. (3.2) can be done with $\mathcal{O}(q \cdot \Bbbk \cdot n)$ operations.

Since for large cluster problems we usually have

$$u \cdot \Bbbk \leq n \quad \text{and} \quad q \ll n,$$

we totally need

$$\mathcal{O}(u \cdot q \cdot \Bbbk^2 + q \cdot \Bbbk \cdot n) = \mathcal{O}(\Bbbk \cdot n)$$

operations to compute a decomposition of the data set V via the SOM/SOBM algorithm.

If we choose $\Bbbk$ significantly smaller than n, e.g., $\Bbbk = \mathcal{O}(\log n)$, this guarantees that we can compute a decomposition much faster than $\mathcal{O}(n^2)$.

Therefore the SOM/SOBM algorithm is a suitable heuristic for the computation of decompositions.

3.5 Practical extensions

In the following we shortly describe two practical extensions of the SOM and the SOBM algorithm, whenever they are used for computing decompositions of a data set V with frequency function f and homogeneity function h.

3.5.1 Pruning

Neuron pruning is a classical technique in the field of neural networks, to simplify the network architecture and therefore also the corresponding model. In our setting each neuron of the Kohonen layer corresponds with one codebook vector w_p. If now n_k is too large after the convergence phase of the SOM, we eliminate those neurons, whose associated codebook vector w_s only represents a small number of input objects, i.e. w_s with $f(\Theta_{w_p}) < \delta_1$ for sufficiently large δ_1, e.g., $\delta_1 := \frac{f(V)}{n_k}$.

Note that after such a neuron pruning, the corresponding decomposition Θ has changed, especially n_k is smaller than before. Pruning has the additional advantage that it prevents the appearing of pseudo clusters (see the earlier discussion in section 3.1).

3.5.2 Early stopping

A main problem of the SOM algorithm is the fact that the number of training steps of the convergence phase has to be fixed a priori and therefore must be set to a large value, because otherwise we cannot be sure that we will reach convergence. If we use our combined SOM/SOBM algorithm, the choice of the length of the SOM convergence phase is rather uncritical, because we have an additional convergence phase of the SOBM algorithm. At a first view, the determination of the right number of convergence steps for the SOBM algorithm seems to be as problematic as for the SOM algorithm. But if we look closer, we detect a nice early stopping criterion for the SOBM algorithm:

To guarantee that $(\Theta_{\hat{W}}(V), \hat{W})$ is an approximate box decomposition, we have to ensure that $\mathrm{overlap}(\hat{W})$ is small. Therefore we have to stop the adaptation of the codebook boxes, if $\mathrm{overlap}(\hat{W}) > \delta_2$ with small δ_2, e.g., $\delta_2 = 0.001$.

Chapter 4

Multilevel Representative Clustering

In this chapter we will extend the basic reduction algorithm (see section 2.3) to a multilevel approach. The main idea is to iterate the decomposition based representative clustering method until the decomposition is fine enough so that the optimal solution of the reduced cluster problem determines an optimal clustering of the original cluster problem.

We will present a general approach that can be always used if the number of clusters k is known a priori. For special homogeneity functions we will additionally describe a powerful extension based on *Perron Cluster* analysis that can be used for cluster problems, where the number of clusters is unknown.

4.1 General approach

Let $V = \{v_1, \ldots, v_n\} \subset \Omega$ be any data set in Ω with frequency function f and homogeneity function h. A point very critical within the application of the basic reduction algorithm (see section 2.3) is the fulfillment of the condition that the decomposition of V has to be a covering of an optimal k-cluster set of (V, f, h).

Suppose now that we have any decomposition Θ of V with codebook W and any optimal k-cluster set $\mathcal{C}$ of $(W, \check{f}, \check{h}_f)$. We know that the extension $\hat{\mathcal{C}}$ of $\mathcal{C}$ on V is a k-cluster set of (V, f, h). Since Θ is a covering of $\mathcal{C}$ by construction, it is also a covering of $\hat{\mathcal{C}}$. Therefore we have $\Gamma_{\check{f},\check{h}_f}(\mathcal{C}) = \Gamma_{f,h}(\hat{\mathcal{C}})$. At the moment we cannot be sure that Θ is also a covering of an optimal k-cluster set of (V, f, h), what would imply that $\hat{\mathcal{C}}$ is optimal. Therefore we try to refine Θ.

Let Θ‘ with codebook vector W‘ be the result of a suitable refinement process, e.g., as it will be described in the next section. If we now compute an optimal k-cluster set $\mathcal{C}$‘ of $(W$‘$, \check{f}, \check{h}_f)$, we can extend it to $\hat{\mathcal{C}}$‘ and compute the weighted intra-cluster homogeneity $\Gamma_{f,h}(\hat{\mathcal{C}}$‘$) = \Gamma_{\check{f},\check{h}_f}(\mathcal{C}$‘$)$. If $\Gamma_{f,h}(\hat{\mathcal{C}}$‘$) > \Gamma_{f,h}(\hat{\mathcal{C}})$, the new clustering is better, i.e. Θ was definitely not a covering of an optimal k-cluster set

of (V, f, h). With Θ‘ we have a new candidate that is a finer decomposition as Θ and that is a covering of a k-cluster set with improved quality.

From the above reflections one easily derives the main idea of the new multilevel representative clustering approach: Refine iteratively the decomposition Θ, until no further improvement of the corresponding representative clustering is observable.

Multilevel Reduction Algorithm

The following algorithm embeds the basic reduction algorithm in a multilevel refinement process. Note that k has to be known a priori.
(1) Compute a decomposition Θ (based on a codebook W) with adaptive choice of n_k, $k \leq n_k \leq \Bbbk \ll n$.
(2) Compute an optimal k-cluster set $\mathcal{C}$ of $(W, \check{f}, \check{h}_f)$.
(3) Extend $\mathcal{C}$ on V: $\hat{\mathcal{C}}$ is a k-cluster set of (V, f, h). Since Θ is a covering of $\hat{\mathcal{C}}$, we have $\Gamma_{\check{f},\check{h}_f}(\mathcal{C}) = \Gamma_{f,h}(\hat{\mathcal{C}})$.
(4) Refine Θ so that the new decomposition Θ‘ of V with codebook W' is also a covering of $\hat{\mathcal{C}}$.
(5) Compute an optimal k-cluster set $\mathcal{C}$‘ of $(W$‘$, \check{f}, \check{h}_f)$.
(6) Extend $\mathcal{C}$‘ on V: $\hat{\mathcal{C}}$‘ is k-cluster set of (V, f, h). Since Θ‘ is a covering of $\hat{\mathcal{C}}$‘, we have $\Gamma_{\check{f},\check{h}_f}(\mathcal{C}$‘$) = \Gamma_{f,h}(\hat{\mathcal{C}}$‘$)$.
(7) If $\Gamma_{\check{f},\check{h}_f}(\mathcal{C}$‘$) > \Gamma_{\check{f},\check{h}_f}(\mathcal{C})$ then set $\Theta := \Theta$‘ and go to step (4), else stop.

For the computation of Θ with adaptive choice of n_k and the codebook W we can use the algorithms described in chapter 3. In the following section we will describe techniques for a refinement of an existing decomposition so that the quality of the corresponding codebook clustering increases.

4.2 Adaptive decomposition refinement

Let $\widetilde{\mathcal{C}}$ be any optimal k-cluster set of (V, f, h) and $\hat{\mathcal{C}}$ be any nearly optimal k-cluster set of (V, f, h). Further let $\Theta := \{\Theta_1, \ldots, \Theta_{n_k}\}$ any decomposition of V with codebook $W = \{w_1, \ldots, w_{n_k}\}$ so that Θ is a covering of $\hat{\mathcal{C}}$, but not of $\widetilde{\mathcal{C}}$. Then there exist clusters $\widetilde{C}_1, \widetilde{C}_2 \in \widetilde{\mathcal{C}}$, $\hat{C}_1, \hat{C}_2 \in \hat{\mathcal{C}}$ and partitions $\Theta_s, \Theta_p \in \Theta$ so that $\widetilde{C}_1 \cap \Theta_s \neq \emptyset$, $\widetilde{C}_2 \cap \Theta_s \neq \emptyset$, $\widetilde{C}_1 \cap \Theta_p \neq \emptyset$ and $\Theta_s \subset \hat{C}_1, \Theta_p \subset \hat{C}_2$.

Suppose now that $\bar{\Theta}$ is a decomposition of $\bar{V} := \Theta_s \cup \Theta_p$. Then the refined decomposition Θ‘ $:= \Theta \setminus \{\Theta_s, \Theta_p\} \cup \{\bar{\Theta}_i \cap \Theta_s \,|\, \bar{\Theta}_i \in \bar{\Theta}\} \cup \{\bar{\Theta}_i \cap \Theta_p \,|\, \bar{\Theta}_i \in \bar{\Theta}\}$ would be better fitting to $\widetilde{\mathcal{C}}$, while still being a covering of $\hat{\mathcal{C}}$. The problem is how to identify the partitions Θ_s and Θ_p, without knowing $\widetilde{\mathcal{C}}$.

The following qualitative observation offers a heuristic solution: Since $\widetilde{\mathcal{C}}$ is optimal, we have $\hat{h}_f(\widetilde{C}_1, \widetilde{C}_1) \gg 0$ and therefore $\hat{h}_f(\Theta_s \cap \widetilde{C}_1, \Theta_p \cap \widetilde{C}_1) \gg 0$. But this gives $\hat{h}_f(\Theta_s, \Theta_p) \gg 0$, which is equivalent to $\check{h}_f(w_s, w_p) \gg 0$. So if we refine all partitions with $\check{h}_f(w_s, w_p) \gg 0$, we can be sure to refine also all partitions which destroy the covering property of Θ for $\widetilde{\mathcal{C}}$. Note that partitions that are already fitting to $\widetilde{\mathcal{C}}$, are also fitting after a refinement.

Decomposition refinement algorithm

Let Θ be any decomposition of V with codebook $W := \{w_1, \ldots, w_{n_k}\}$.
(1) Identify all indices $s, p \in \{1, \ldots, n_k\}$ so that $\check{h}_f(w_s, w_p) > \sigma$ with $\sigma \gg 0$. Let I be the resulting index subset.
(2) Set $\bar{V} := \bigcup_{s \in I} \Theta_s$.
(3) Compute a decomposition $\bar{\Theta}$ of $\bar{V}$ with $\bar{n}_k$ partitions $\bar{\Theta}_i$.
(4) Set $\Theta' := \Theta \setminus \{\Theta_s \,|\, s \in I\} \cup \{\bar{\Theta}_i \cap \Theta_s \,|\, \bar{\Theta}_i \in \bar{\Theta}\,,\ s \in I\}$.

Obviously the above algorithm increases the number of partitions from n_k to maximally $n_k + (\bar{n}_k - 1)|I|$ partitions. Often several of the new partitions $\bar{\Theta}_i \cap \Theta_s$ are nearly empty. Therefore step (4) is improved by the following condition: Θ_s is replaced only by those $\bar{\Theta}_i \cap \Theta_s$, with $f(\bar{\Theta}_i \cap \Theta_s) \gg 0$. Note that in this case the refined Θ has to be adapted slightly to guarantee that it is still a decomposition. This can be easily done, if we use the SOM algorithm for the computation of the decomposition of $\bar{V}$:

Let $\bar{W}$ be the codebook of $\bar{\Theta}$ generated by the SOM algorithm. For each $s \in I$ we set $I_s := \{i \,|\, f(\bar{\Theta}_i \cap \Theta_s) \gg 0\}$. Then the reduced codebook W_{I_s} defines a decomposition $\bar{\Theta}_{W_{I_s}}$ of $\bar{V}$. If we replace Θ_s by $\{\bar{\Theta}_{s,i} \cap \Theta_s \,|\, \bar{\Theta}_{s,i} \in \bar{\Theta}_{W_{I_s}}\}$ for all $s \in I$, the refined Θ is still a decomposition of V.

Instead of the suggested refinement algorithm, one could also think about using methods that tries to grow the SOM adaptively [26, 17]. In this case one has to assure that the growing process is driven by the homogeneity function h. If the cluster problem is geometrically based, this should be no problem.

4.3 Approach based on Perron Cluster analysis

In this section, we will extend our general multilevel cluster approach by using results and methods from the theory of *Perron Cluster* analysis that has been recently developed by DEUFLHARD ET AL.. We will show that for cluster problems with a stochastic homogeneity functions, this extended approach can be used for a fast identification and efficient description of clusters, even if a correct number of clusters k is not known a priori.

4.3.1 Theoretical background

In the following, we will give a short description of the theory of Perron Cluster analysis. For details and proofs see [16, 13].

Suppose we have a primitive stochastic $n_{\mathbb{k}} \times n_{\mathbb{k}}$ matrix $\mathcal{S}$, i.e. there exist an $m \in N$ so that $\mathcal{S}^m > 0$, the entries $\mathcal{S}_{i,j}$ are non-negative and the sum of each row equals one. As a consequence, the constant vector $e = (1, \ldots, 1)^T$ is an eigenvector corresponding to the simple eigenvalue $\lambda_1 = 1$ of $\mathcal{S}$. For all other eigenvalues λ_i of $\mathcal{S}$ we have $|\lambda_i| < 1$.

Let $\pi = (\pi_1, \ldots, \pi_{n_{\mathbb{k}}})^T$ any strictly positive distribution so that $\pi^T e = 1$ and $\pi^T \mathcal{S} = \pi^T$. We suppose that $\mathcal{S}$ is reversible with respect to π, i.e. $\mathcal{D}^2\mathcal{S} = \mathcal{S}^T\mathcal{D}^2$, where $\mathcal{D} := diag(\sqrt{\pi_i})$ is called a *weighting matrix* of $\mathcal{S}$. If $\mathcal{S}$ is reversible, it is self-adjoint with respect to the weighted scalar product $< x, y >_\pi := x^T\mathcal{D}^2 y$ and consequently, all eigenvalues are real. Additionally there exist a basis of π-orthogonal right eigenvectors, which diagonalizes $\mathcal{S}$ and for every right eigenvector Y there is an associated left eigenvector $\bar{Y} = \mathcal{D}^2 Y$, which corresponds to the same eigenvalue.

In the following let $I_1, \ldots, I_k$ any disjoint index subsets with $I_p \subset \{1, \ldots, n_{\mathbb{k}}\}$, $p \in \{1, \ldots, k\}$, and $\bigcup_{p=1}^k I_p = \{1, \ldots, n_{\mathbb{k}}\}$. Based on these index subsets we define a so called *coupling matrix* $\hat{\mathcal{S}} := (\mathcal{S}_{I_s,I_p})_{1 \le s,p \le k}$ via

$$\mathcal{S}_{I_s,I_p} := \sum_{i \in I_s} \sum_{j \in I_p} \frac{\pi_i \mathcal{S}(i,j)}{\sum_{i \in I_s} \pi_i}. \tag{4.1}$$

Lemma 4.3.1 *The matrix $\hat{\mathcal{S}}$ is stochastic and reversible with respect to the distribution $\hat{\pi} := (\hat{\pi}_1, \ldots, \hat{\pi}_k)^T$ where $\hat{\pi}_p := \sum_{i \in I_p} \pi_i$.*

Proof: Since $\mathcal{S}$ is stochastic, i.e. $\sum_{j=1}^{n_{\mathbb{k}}} \mathcal{S}(i,j) = 1$ for $1 \le i \le n_{\mathbb{k}}$, we have

$$\begin{aligned} \sum_{p=1}^{k} \mathcal{S}_{I_s,I_p} &= \sum_{i \in I_s} \frac{\pi_i}{\hat{\pi}_s} \sum_{p=1}^{k} \sum_{j \in I_p} \mathcal{S}(i,j) \\ &= \sum_{i \in I_s} \frac{\pi_i}{\hat{\pi}_s} \sum_{j=1}^{n_{\mathbb{k}}} \mathcal{S}(i,j) = \sum_{i \in I_s} \frac{\pi_i}{\hat{\pi}_s} = 1 \end{aligned}$$

and therefore $\hat{\mathcal{S}}$ is stochastic. We further have

$$\hat{\pi}_s \mathcal{S}_{I_s,I_p} = \sum_{i \in I_s} \sum_{j \in I_p} \pi_i \mathcal{S}(i,j) \text{ for } 1 \le s,p \le k.$$

Since $\mathcal{S}$ is reversible, i.e. $\pi_i \mathcal{S}(i,j) = \pi_j \mathcal{S}(j,i)$ the reversibility of $\hat{\mathcal{S}}$ follows immediately. □

We are interested in index subsets $I_1, \ldots, I_k$ that lead to a nearly diagonal coupling matrix:

Definition 4.3.2 *Choose $p \in \{1, \ldots, k\}$. Then we call I_p an almost invariant aggregate of $\mathcal{S}$, if $\mathcal{S}_{I_p, I_p} \approx 1$. If I_p is an almost invariant aggregate of $\mathcal{S}$ for all $p \in \{1, \ldots, k\}$, we call $I_1, \ldots, I_k$ a covering set of almost invariant aggregates of $\mathcal{S}$. In this case we call k an optimal number of almost invariant aggregates of $\mathcal{S}$.*

One easily checks that almost invariant aggregates correspond to a permutation of $\mathcal{S}$ so that the matrix is nearly block-diagonal:

Lemma 4.3.3 *Let $I_1, \ldots, I_k \subset \{1, \ldots, n_{\mathbb{k}}\}$ any covering set of almost invariant aggregates of $\mathcal{S}$. Then the indices $\{1, \ldots, n_{\mathbb{k}}\}$ can be ordered so that the matrix $\mathcal{S}$ is of block-diagonally dominant form:*

$$\mathcal{S} = D + E = \begin{pmatrix} D_{1,1} & E_{1,2} & \ldots & E_{1,k} \\ E_{2,1} & D_{2,2} & \ldots & E_{2,k} \\ \ldots & \ldots & \ldots & \ldots \\ E_{k,1} & E_{k,2} & \ldots & D_{k,k} \end{pmatrix}.$$

Herein the perturbation matrix E satisfies $E = O(\epsilon)$ where ϵ is some perturbation parameter.

Supposing that the conditions of Lemma 4.3.3 hold, we set:

$$\mathcal{S}(\epsilon) := \mathcal{S}(0) + \epsilon \mathcal{S}^{(1)} + \epsilon^2 \mathcal{S}^{(2)} + \ldots,$$

where $\mathcal{S}(0) = D$ is the unperturbed part of $\mathcal{S}$.

It follows from perturbation theory [45] that the spectrum of $\mathcal{S}(\epsilon)$ can be divided into two parts:

1. The *Perron Cluster* including the *Perron Root* $\lambda_1 = 1$ and the $k - 1$ eigenvalues $\lambda_2(\epsilon), \ldots, \lambda_k(\epsilon)$ approaching 1 for $\epsilon \to 0$.
2. The remaining part of the spectrum, bounded away from 1 for $\epsilon \to 0$.

The eigenvectors corresponding to eigenvalues of the Perron Cluster have a useful property:

Lemma 4.3.4 *Let $\lambda_1(\epsilon), \ldots, \lambda_k(\epsilon)$ be the Perron Cluster of $\mathcal{S}$. Then there exits a covering set of almost invariant aggregate $I_1, \ldots, I_k$ of $\mathcal{S}$ so that the eigenvectors $Y_1, \ldots, Y_k \in \mathbf{R}^{n_{\mathbb{k}}}$, corresponding to $\lambda_1(\epsilon), \ldots, \lambda_k(\epsilon)$, are almost constant on each I_s, i.e. we have for all $s \in \{1, \ldots, k\}$:*

$$i, j \in I_s \implies (\forall p \in \{1, \ldots, k\})\, Y_p(i) \approx Y_p(j).$$

The above theoretical results lead to a powerful method for the determination of an optimal number k of almost invariant aggregates of $\mathcal{S}$:

Suppose there exist — a priori unknown — index subsets $I_1, \ldots, I_k$ so that the conditions of Lemma 4.3.3 hold. Then there exists an ϵ_* so that $\mathcal{S}(\epsilon_*) = \mathcal{S}$. If ϵ_* is sufficiently small, we can find a large gap within the spectrum of $\mathcal{S}$ between the eigenvalues λ_k and λ_{k+1} of $\mathcal{S}$. In this case k is an optimal number of almost invariant aggregates of $\mathcal{S}$.

But we cannot only determine an optimal number of almost invariant aggregates, also the index subsets themselves can be computed based on Lemma 4.3.4:

Let $Y_1, \ldots, Y_k \in \mathbf{R}^{n_\mathbb{k}}$ be the eigenvectors corresponding to the eigenvalues $\lambda_1(\epsilon), \ldots, \lambda_k(\epsilon)$ of $\mathcal{S}$. Then the identification of k groups of nearly identical k-tuple $Y(i) := (Y_1(i), \ldots, Y_k(i))^T$ of eigenvector components associated with each $i \in \{1, \ldots, n_\mathbb{k}\}$, is sufficient to identify the covering set of almost invariant aggregates $I_1, \ldots, I_k$ of $\mathcal{S}$. Obviously such a grouping can be done via the computation of a k-cluster set of the set $V_Y := \{Y(1), \ldots, Y(n_\mathbb{k})\}$ with frequency function $f_Y(v) := 1$ for $v \in V_Y$ and a suitable homogeneity function h_Y, e.g., $h_Y = h_d$, where d is a distance function in $\mathbf{R}^k$.

4.3.2 Stochastic homogeneity functions

In the following we suppose that the homogeneity function h is stochastic in V with respect to f:

Definition 4.3.5 *We call any homogeneity function $h : \Omega \times \Omega \longrightarrow [0, 1]$ stochastic in V with respect to f if we have*

$$\sum_{w \in V} h(v, w) f(w) = 1 \quad \text{forall}\, v \in V. \tag{4.2}$$

Set $P(v, w) := h(v, w) f(w)$ for any $v, w \in V$. We can directly extend P on subsets of V, if we define for any non-void subsets $V_1, V_2 \subset V$:

$$\hat{P}(V_1, V_2) := \sum_{v \in V_1} \sum_{w \in V_2} \frac{f(v) P(v, w)}{f(V_1)}. \tag{4.3}$$

Using earlier definitions (see section 2.3) we get:

Lemma 4.3.6 $\hat{P}(V_1, V_2) = \hat{h}_f(V_1, V_2) \hat{f}(V_2)$ *for any non-void $V_1, V_2 \subset V$.*

Proof:

$$\begin{aligned}\hat{h}_f(V_1,V_2) &= \frac{1}{f(V_1)f(V_2)}\sum_{v\in V_1}\sum_{w\in V_2}h(v,w)f(v)f(w)\\ &= \frac{1}{f(V_1)f(V_2)}\sum_{v\in V_1}\sum_{w\in V_2}P(v,w)f(v)\\ &= \frac{1}{\hat{f}(V_1)\hat{f}(V_2)}\hat{P}(V_1,V_2)f(V_1) \;=\; \frac{\hat{P}(V_1,V_2)}{\hat{f}(V_2)}\end{aligned}$$

□

We have a sort of reversibility of P with respect to f:

Lemma 4.3.7 *$f(v)P(v,w) = f(w)P(w,v)$ for all $v,w \in V$.*

Proof: Since h is a homogeneity function, we have $h(v,w) = h(w,v)$ and therefore also

$$f(v)P(v,w) \;=\; f(v)h(v,w)f(w) \;=\; f(v)h(w,v)f(w) \;=\; P(w,v)f(w).$$

for all $v,w \in V$. □

From Lemma 4.3.7 directly follows:

$$f(V_1)\hat{P}(V_1,V_2) = f(V_2)\hat{P}(V_2,V_1)$$

for all non-void subsets $V_1, V_2 \subset V$.

Based on $\hat{P}$ and a decomposition of V we can define a stochastic and reversible matrix $\mathcal{S}$:

Lemma 4.3.8 *Let $\Theta := \{\Theta_1,\ldots,\Theta_{n_\Bbbk}\}$ be any decomposition of V. Define the $n_\Bbbk \times n_\Bbbk$ matrix $\mathcal{S}$ via $\mathcal{S}(i,j) := \hat{P}(\Theta_i,\Theta_j)$. Further set $\pi := (\pi_1,\ldots,\pi_{n_\Bbbk})^T$ with $\pi_s := \frac{f(\Theta_s)}{f(V)}$. Then we have:*
(i) If for any $i,j \in \{1,\ldots,n_k\}$ there exist $p_1,\ldots,p_m$, $m \in N$, so that $p_1 = i$, $p_m = j$ and $\mathcal{S}(p_t,p_{t+1}) > 0$ for $1 \le t \le m-1$, then the matrix $\mathcal{S}$ is primitive.
(ii) The matrix $\mathcal{S}$ is stochastic.
(iii) π is a strictly positive distribution with $\pi^T e = 1$ and $\pi^T\mathcal{S} = \pi^T$.
(iv) The matrix $\mathcal{S}$ is reversible with respect to π.

Proof:
(i) is obvious and (ii) follows directly from the fact that h is stochastic and Θ is a decomposition of the data set V.

(iii) Obviously we have $\pi^T e = 1$. Further let $\mathcal{S}_{*j} := (\mathcal{S}(1,j),\ldots,\mathcal{S}(n_{\Bbbk},j))^T$ be the j-th column of the matrix $\mathcal{S}$. Using Lemma 4.3.7 we have:

$$\begin{aligned}
\pi^T\mathcal{S}_{*j} &= \frac{1}{f(V)}\sum_{i=1}^{n_{\Bbbk}} f(\Theta_i)\hat{P}(\Theta_i,\Theta_j) \\
&= \frac{1}{f(V)}\sum_{i=1}^{n_{\Bbbk}} f(\Theta_i)\sum_{v\in\Theta_i}\sum_{w\in\Theta_j}\frac{f(v)P(v,w)}{f(\Theta_i)} \\
&= \frac{1}{f(V)}\sum_{i=1}^{n_{\Bbbk}}\sum_{v\in\Theta_i}\sum_{w\in\Theta_j} f(v)P(v,w) \\
&= \frac{1}{f(V)}\sum_{i=1}^{n_{\Bbbk}}\sum_{v\in\Theta_i}\sum_{w\in\Theta_j} f(w)P(w,v) \\
&= \frac{1}{f(V)}\sum_{w\in\Theta_j} f(w)\sum_{i=1}^{n_{\Bbbk}}\sum_{v\in\Theta_i} P(w,v) \\
&= \frac{1}{f(V)}\sum_{w\in\Theta_j} f(w) \;=\; \pi_j.
\end{aligned}$$

(iv) For any $i,j\in\{1,\ldots,n_{\Bbbk}\}$ we have:

$$\begin{aligned}
\pi_i\mathcal{S}(i,j) &= \frac{f(\Theta_i)}{f(V)}\hat{P}(\Theta_i,\Theta_j) \\
&= \frac{1}{f(V)}\sum_{v\in\Theta_i}\sum_{w\in\Theta_j} f(v)P(v,w) \\
&= \frac{1}{f(V)}\sum_{v\in\Theta_i}\sum_{w\in\Theta_j} f(w)P(w,v) \\
&= \frac{f(\Theta_j)}{f(V)}\sum_{w\in\Theta_j}\sum_{v\in\Theta_i}\frac{f(w)P(w,v)}{f(\Theta_j)} \;=\pi_j\mathcal{S}(j,i).
\end{aligned}$$

□

Based on $\mathcal{S}$ we can use Perron Cluster analysis to determine an optimal number k and to identify the almost invariant aggregates of $\mathcal{S}$. The following Theorem shows that a covering set of k almost invariant aggregates of $\mathcal{S}$ corresponds to a nearly optimal k-cluster set of $(\Theta,\hat{f},\hat{h})$ what we know is equivalent to a nearly optimal representative clustering for any codebook W of Θ (see Theorem 2.3.9).

Theorem 4.3.9 *Let k be an optimal number of almost invariant aggregates of the matrix $\mathcal{S}$ and let $I_1, \ldots, I_k \subset \{1, \ldots, n_{\mathbb{k}}\}$ be the corresponding covering set of almost invariant aggregates. Then we have:*
(i) $\frac{1}{k}\sum_{s=1}^{k} \mathcal{S}_{I_s,I_s} \geq 1 - \epsilon_*$, with small $\epsilon_* := 1 - \min_s \mathcal{S}_{I_s,I_s}$
(ii) If we set $\bar{\mathcal{C}} := \{\bar{C}_1, \ldots, \bar{C}_k\}$ with $\bar{C}_s = \{\Theta_p \,|\, p \in I_s\}$, then $\bar{\mathcal{C}}$ is an nearly optimal k-cluster set of $(\Theta, \hat{f}, \hat{h})$, with $\Gamma_{\hat{f},\hat{h}_f}(\bar{\mathcal{C}}) = \frac{1}{k}\sum_{s=1}^{k} \mathcal{S}_{I_s,I_s}$

Proof:
(i) Since each I_s is almost invariant, we have $\mathcal{S}_{I_s,I_s} \approx 1$ for $s = 1, \ldots, k$.
(ii) Obviously $\mathcal{C}$ is a k-cluster set of Θ. We have:

$$\begin{aligned}
\Gamma_{\hat{f},\hat{h}_f}(\bar{\mathcal{C}}) &= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{\hat{f}(\bar{C}_s)} \sum_{V_1 \in C_s} \sum_{V_2 \in C_s} \hat{h}_f(V_1, V_2)\hat{f}(V_1)\hat{f}(V_2) \\
&= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{\hat{f}(\bar{C}_s)} \sum_{p_1 \in I_s} \sum_{p_2 \in I_s} \hat{h}_f(\Theta_{p_1}, \Theta_{p_2})\hat{f}(\Theta_{p_1})\hat{f}(\Theta_{p_2}) \\
&= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{\hat{f}(\bar{C}_s)} \sum_{p_1 \in I_s} \sum_{p_2 \in I_s} \hat{P}(\Theta_{p_1}, \Theta_{p_2})\hat{f}(\Theta_{p_1}) \\
&= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{\hat{f}(\bar{C}_s)} \sum_{p_1 \in I_s} \sum_{p_2 \in I_s} \mathcal{S}(p_1, p_2) f(\Theta_{p_1}) \\
&= \frac{1}{k}\sum_{s=1}^{k} \frac{1}{\hat{f}(\bar{C}_s)} \sum_{p_1 \in I_s} \sum_{p_2 \in I_s} \frac{(\sum_{p \in I_s} f(\Theta_p))\mathcal{S}(p_1, p_2) f(\Theta_{p_1})}{\sum_{p \in I_s} f(\Theta_p)} \\
&= \frac{1}{k}\sum_{s=1}^{k} \frac{\sum_{p \in I_s} f(\Theta_p)}{\hat{f}(\bar{C}_s)} \sum_{p_1 \in I_s} \sum_{p_2 \in I_s} \frac{\mathcal{S}(p_1, p_2)\pi_{p_1}}{\sum_{p \in I_s} \pi_p} \\
&= \frac{1}{k}\sum_{s=1}^{k} \mathcal{S}_{I_s,I_s}.
\end{aligned}$$

Since $\frac{1}{k}\sum_{s=1}^{k} \mathcal{S}_{I_s,I_s} \leq 1$, we have $\Gamma_{\hat{f},\hat{h}_f}(\bar{\mathcal{C}}) \leq 1$ and therefore (i) guarantees that $\bar{\mathcal{C}}$ is nearly optimal. □

If we set $\mathcal{C} := \{C_1, \ldots, C_k\}$ with $C_s := \bigcup_{p \in I_s} \Theta_p$, then using Theorem 2.3.9 and Lemma 2.3.5 we have $\Gamma_{f,h}(\mathcal{C}) = \Gamma_{\hat{f},\hat{h}_f}(\bar{\mathcal{C}})$ and therefore $\mathcal{C}$ is a nearly optimal k-cluster set of (V, f, h).

Note that it is possible that there exist different k so that k is an optimal number of invariant aggregates. But this is not surprising, because cluster problems might also have different correct numbers of clusters.

Multilevel Reduction Algorithm for stochastic homogeneity functions

We can use our results for a special version of the multilevel reduction algorithm that can be used even if the number of clusters k is not known a priori:
(1) Compute a decomposition Θ (based on a codebook W) with adaptive choice of $n_{\mathbb{k}}$, $n_{\mathbb{k}} \leq \mathbb{k} \ll n$.
(2a) Compute the matrix $\mathcal{S}$.
(2b) Compute an optimal number k of almost invariant aggregates of $\mathcal{S}$ via Perron Cluster analysis.
(2c) Compute an optimal k-cluster set of (V_Y, f_Y, h_Y), leading to a covering set of k almost invariant aggregates $I_1, \ldots, I_k \subset \{1, \ldots, n_{\mathbb{k}}\}$ of $\mathcal{S}$.
(3) Set $\mathcal{C} := \{C_1, \ldots, C_k\}$ with $C_s = \bigcup_{p \in I_s} \Theta_p$. Then $\mathcal{C}$ is a k-cluster set of (V, f, h) with $\Gamma_{f,h}(\mathcal{C}) = \frac{1}{k} \sum_{s=1}^{k} \mathcal{S}_{I_s, I_s}$.
(4) Refine Θ so that the new decomposition Θ‘ of V with codebook W' is also a covering of $\mathcal{C}$.
(5) Repeat the steps (2a)-(2c) and (3) with Θ‘ instead of Θ, leading to a k‘-clustering $\mathcal{C}$‘ of (V, f, h).
(6) If k‘ $\neq k$ then set $\Theta := \Theta$‘ and $k = k$‘ and go to step (4)
(7) If $\Gamma_{f,h}(\mathcal{C}\text{‘}) > \Gamma_{f,h}(\mathcal{C})$, then set $\Theta := \Theta$‘ and go to step (4), else stop.

Identification of discriminating attributes based on Perron Cluster analysis.

In section 2.4.2 we have presented an algorithm for the identification of discriminating attributes. We now give a simple heuristic criterion to decide if an attribute set $\mathcal{A}(J^C)$ is redundant or not:

Let $\mathcal{C} := \{C_1, \ldots, C_k\}$ be any optimal k-cluster set of a data set V with a covering Θ_W that is defined based on a codebook W according to Eq. (3.2). Further let $W(J)$ be the projection of W on $\Omega(J)$ and $\Theta_{W(J)} := \{\Theta_1, \ldots, \Theta_{n_k}\}$ be the corresponding decomposition of $V(J)$. If the eigenvalue spectrum of the matrix $\mathcal{S}$ corresponding to Θ_W, is nearly the same as the spectrum of matrix $\mathcal{S}$‘ corresponding to $\Theta_{W(J)}$, then the attribute set $\mathcal{A}(J^C)$ is redundant.

The above criterion uses the obvious fact that an attribute set $\mathcal{A}(J^C)$ is redundant, if the cluster structure of the cluster problem is independent of the attributes in $\mathcal{A}(J^C)$.

Natural and artificial stochastic homogeneity functions

For a reversible dynamic system, the homogeneity function h_s, as defined in Lemma 1.1.4, is stochastic:

Lemma 4.3.10 *Let $(X(t))_{t=1,\dots,T}$ any representative trajectory of length $T \in \mathbf{N}$ of a reversible dynamic system in Ω, i.e. $|\{t \,|\, X(t) = v, X(t+1) = w\}| = |\{t \,|\, X(t) = w, X(t+1) = v\}|$ for all $v, w \in \Omega$. Then the homogeneity function h_s is stochastic with respect to the frequency function f that is given by $f(v) := |\{t \,|\, X(t) = v\}|$.*

Proof: We have $f(v)S(v,w) = f(w)S(w,v)$ and so $h_S(v,w) = \frac{S(v,w)}{f(w)}$ for any $v, w \in V$. Since $\sum_{w \in V} S(v,w) = 1$ for all $v \in V$, h_S is stochastic. □

Note that the condition for $\mathcal{S}$ primitive (see Lemma 4.3.8) is true for any decomposition of $V := \{X(t) \,|\, t = 1, \dots, T\}$ because one easily checks that for all $v, w \in V$, there exist $v_1, \dots, v_m \in V$, $m \leq T$, so that $v = v_1$, $w = v_m$ and $S(v_i, v_{i+1}) > 0$ for $1 \leq i \leq m-1$.

In addition to natural given stochastic homogeneity functions, we can also construct them artificially: For each homogeneity function h there exists a transformation into a stochastic homogeneity function $\widetilde{h}$ with respect to a suitable frequency function:

Lemma 4.3.11 *Let V be any data set in Ω with homogeneity function h and frequency function f. Set $\widetilde{f}(v) := \sum_{w \in V} h(v,w) f(v) f(w)$ for all $v \in \Omega$. Define $\widetilde{h} : \Omega \times \Omega \longrightarrow [0,1]$ via*

$$\widetilde{h}(v,w) := \frac{h(v,w) f(v) f(w)}{\widetilde{f}(v)\widetilde{f}(w)} \; v, w \in \Omega.$$

Then $\widetilde{h}$ is a stochastic homogeneity function with respect to $\widetilde{f}$.

For well structured simple cluster problems, i.e. cluster problems with clusters of nearly identical size and a nearly identical homogeneity and a nearly constant frequency function, we have $\widetilde{f}(v) \approx \text{const.}$ and therefore $\widetilde{h}(v,w) \approx \text{c} \cdot h(v,w)$, where c is a constant value. This guarantees that an optimal k-cluster set of $(V, \widetilde{f}, \widetilde{h})$ is nearly an optimal k-cluster set of (V, f, h). Note that in the case of geometric cluster problems with a distance function d, we usually have $h_d(v,w) > 0$ for nearly all $v, w \in V$, because h_d vanishes only for objects with maximal distance. Therefore the constructed matrix $\mathcal{S}$ will be primitive. We will use this observation to compute an optimal number of clusters for our simple example from section 1.4:

Example: Determination of a correct number of clusters

Obviously the cluster problem for the data set V as given by Figure 1.3 is well structured and the frequency function f is constant with $f(v) = 1$ for all $v \in V$. For $h = h_d$ with $d = d_{euclid}$ we get $\widetilde{f}(v) \in [4.90, 7.24]$ for $v \in V$. To reduce the variance we slightly modify our homogeneity function. We set

$$h(v, w) := 1 - \frac{d(v, w)}{(\max_{v,w \in V} d(v, w))} \quad , v, w \in \Omega.$$

Obviously this homogeneity function is still suitable for the computation of geometrically based clusters. Now we get $\widetilde{f}(v) \in [3.81, 5.78]$ for $v \in V$, i.e. the variance has decreased. The modification of h has no influence on the ranking of optimal k-cluster sets for different k. We still cannot use the values $\Gamma_{f,h}(\mathcal{C}(k))$ to determine the optimal number of clusters (see Table 4.1).

optimal k-cluster set $\mathcal{C}(k)$	$\Gamma_{f,h}(\mathcal{C}(k))$
$\mathcal{C}(1) := V$	4.85
$\mathcal{C}(2) := \{\{a, b, c, d, e, f\}, \{g, h, i\}\}$	3.67
$\mathcal{C}(3) := \{\{a, b, c\}, \{d, e, f\}, \{g, h, i\}$	2.72
$\mathcal{C}(4) := \{\{a\}, \{b, c\}, \{d, e, f\}, \{g, h, i\}\}$	2.10
$\mathcal{C}(9) := \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{h\}, \{i\}\}$	1.00

Table 4.1: **Example: Optimal k-cluster sets of (V, f, h) for different k with modified homogeneity function.**

Based on $\widetilde{h}$ and the trivial decomposition $\Theta_V := \{\{v\} \,|\, v \in V\}$, we can compute the matrix $\mathcal{S}$. The spectrum of $\mathcal{S}$ is given in Table 4.2:

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9
1.000	0.577	0.165	0.046	0.041	0.025	0.018	0.015	0.010

Table 4.2: **Example: Spectrum of matrix $\mathcal{S}$.**

Obviously the large gap between the Perron Cluster and the remaining part of the spectrum is at $k = 2$, indicating that $\mathcal{S}$ has two almost invariant aggregates. Therefore the optimal number of clusters for our cluster problem is also 2. The fact that the distance between the Perron Root and λ_2 is also very large, is a result of the artificial construction of the stochastic homogeneity function $\widetilde{h}$. We will see in chapter 5 that for natural stochastic homogeneity functions, as e.g., the

dynamically based function h_S, the Perron Cluster is always approaching 1, if at least there exist two clusters within the data.

Since the eigenvector associated with the Perron Root is the constant vector $e = (1, \ldots, 1)^T$, we only need the eigenvector Y_2 associated with λ_2, to compute the almost invariant aggregates.

We have $Y_2 = (-0.35, -0.21, -0.20, -0.13, -0.13, -0.13, 0.54, 0.53, 0.42)^T$. Comparing the components of Y_2 we can directly identify the almost invariant aggregates $I_1 := \{1, \ldots, 6\}$ and $I_2 := \{7, \ldots, 9\}$. One easily checks that this solution corresponds to the optimal k-cluster set $\mathcal{C}(2)$ of (V, f, h).

Chapter 5

Applications

5.1 Conformational analysis of biomolecules

5.1.1 Introduction

The analysis of biomolecular structure and function is one of the real challenges of scientific computing nowadays. Advances in this area will have tremendous impact on the design and identification process of new pharmaceutical drugs. The enrichment of chemical databases with structural and functional information will allow the use of *virtual screening* procedures, reducing time and costs of the pharmaceutical research decidedly.

The key concept to characterize *structure* has become the characterization in terms of *geometric conformations*, often just called conformations in the literature. In contradiction to structure, *function*, seems to depend on the dynamic properties of the molecule and therefore should be rather characterized by what has been called *metastable conformations*. Any type of conformations consists of sets of possible molecular states. In geometric conformations such sets are defined via the geometric similarity of different states. In metastable conformations such sets are defined via the high probability of the molecule to stay in such a set, once it is in such a set.

In classical molecular dynamics [2] a molecule is modeled by a Hamiltonian function

$$H(q,p) \;=\; \tfrac{1}{2}\, p^T M^{-1} p \;+\; V(q),$$

where q and p are the corresponding positions and momenta of the atoms, M denotes the diagonal mass matrix, and V is a differentiable potential. The Hamiltonian function H is defined on the phase space. The corresponding canonical

equations of motion

$$\dot{q} = M^{-1}p, \qquad \dot{p} = -\text{grad}\, V \tag{5.1}$$

describe the dynamics of the molecule. The formal solution of (5.1) with initial state $x_0 = (q(0), p(0))$ is given by $x_t = (q(t), p(t)) = \Phi_V^\tau x_0$, where Φ_V^τ denotes the flow.

In [14] a first attempt had been made to identify metastable conformations on the basis of the so-called Perron-Frobenius operator. That approach, though principally opening the door to the new concept of conformation dynamics, had been more or less restricted to toy molecules. In a further step, performing some momenta averaging based on the Boltzmann distribution f_0 for given heat bath temperature, the Perron-Frobenius operator in phase space has been replaced by a different Markov operator in position space [58, 59]. This new operator has much nicer theoretical properties and it may be interpreted as the transfer operator of an underlying Markov chain $X(t)$. This Markov chain can be realized via Hybrid Monte-Carlo (HMC) methods [22]:

- random choice of momenta from a Gaussian distribution,
- deterministic propagation of the molecular system by the flow Φ_V^τ with potential V and over short time τ,
- acceptance or rejection of new configurations by an appropriate transition kernel K of the underlying Markov process, e.g., Metropolis-Hastings.

Like classical Monte-Carlo, HMC also suffers from possible *trapping* in local potential wells. In order to overcome this unwanted effect, an adaptive temperature version has been worked out [22] that embeds the given problem into a family of problems with flow $\Phi_V^{\tau,s}$ in terms of an embedding parameter $s \in [0, 1]$. At $s = 0$, only a few metastable subsets need to be identified, whereas at $s = 1$ a rich structure of conformations might arise. Two types of embedding are in quite common use: *temperature embedding* and *potential embedding*. Upon examining the equations of motion, one immediately sees that, in the context of HMC, temperature embedding can be realized by the following flow:

$$\Phi_V^{\tau,s} = \Phi_{sV}^{s^{-2}\tau}, \tag{5.2}$$

which requires a scaling of the potential and the time step of propagation [58].

Any kind of embedding stimulates the idea of a hierarchical algorithm consisting of the following steps:

1. Simulate the molecular system for a specific parameter (say, high temperature), which causes the flow to overcome specific energy barriers.

2. Identify metastable subsets.

3. Increase the parameter (say, lower the temperature), but restrict the simulation to one of the metastable subsets. Go to step 1.

This algorithm will generate a hierarchy of subsets that can be sampled independently at each level. The restriction of an HMC-simulation to a given metastable subset C_s requires only a slight modification of the Markov kernel K to K_s [23]. The additional rule is that any configuration outside the subset C_s will be rejected. Detailed balance still holds for this modified Markov kernel so that K_s is still reversible. Since C_s is metastable, only a few rejections will be expected with respect to the new rule. Moreover, trapping should thus be avoided, since energy barriers towards all other metastable subsets can be ignored. A further exploitation of this embedding structure is given in [23], where an uncoupling/coupling technique has been suggested and worked out.

A schematic diagram of such a hierarchy is given in Figure 5.1. As can been seen there, each cluster needs to be described by appropriate boundaries. To save computer time over the whole simulation, one is interested in efficient descriptions of the identified metastable subsets (see section 1.3).

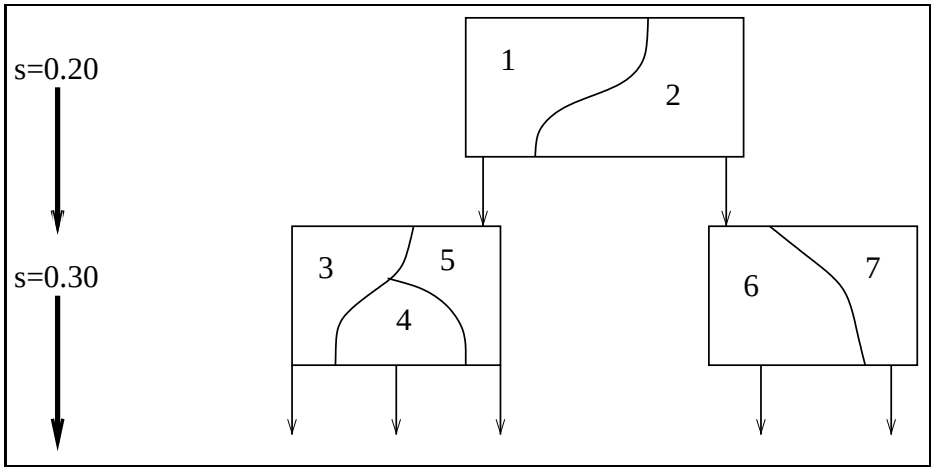


Figure 5.1: **Hierarchical scheme of clustering combined with parameter embedding.** The numbers denote metastable conformations at different levels of the hierarchical embedding scheme.

As described in section 1.1, the problem of finding metastable conformations can be transformed into a cluster problem, if we use a sufficiently long Markov chain $X(t)$ as a representative trajectory. Since $X(t)$ is reversible (see [59]), we can use Perron Cluster analysis to determine an optimal number k of metastable conformations (see section 4.3).

Based on an uniform box decomposition, the conformations of small molecules like *n-pentane* have been recently analyzed successfully [58]. For larger molecules such a simple decomposition is not possible, because the number of boxes explode (see section 2.2). Therefore the use of approximate box decompositions, computed via the SOBM algorithm, allows for the first time the conformational analysis of molecules of practically relevant size.

5.1.2 Adaptation of SOM and SOBM to cyclic data

One easily checks that the computing time of the SOM and the SOBM algorithm strongly depends on the dimension of Ω. The dimension of the position space of molecules is three times the number of atoms and therefore it is very large even for small molecules. The following observation leads to a reduction of the dimension: For each molecule there exists a set of so called *torsion angles*, which are sufficient for a rough reconstruction of the spatial position of each atom of the molecule together with the corresponding equilibrium bonds and angles [39]. Without loss of generality we assume each torsion angle within $[-\pi, \pi]$. Then we define Ω as the space spanned by the torsion angles of the molecule. Since the analysis of cyclic data is different from non-cyclic data (see [24] for a comprehensive introduction), it is not surprising that we have to adapt the SOM and the SOBM algorithm to cyclic data.

First one has to choose a suitable distance measure. We suggest to use the distance on the q-dimensional unit circle, i.e. we define $\mathrm{dist} : \Omega \times \Omega \to \mathbf{R}_0^+$ via

$$\begin{aligned} \mathrm{dist}(x,y) &:= F(d_1(x_1,y_1),\ldots,d_q(x_q,y_q)) := (\sum_{i=1}^{q} d_i(x_i,y_i))^{1/2} \\ \text{with} \quad d_i(x_i,y_i) &:= (\sin(x_i)-\sin(y_i))^2 + (\cos(x_i)-\cos(y_i))^2 \end{aligned}$$

for $x, y \in \Omega$, where x_i and y_i denote the values of the ith torsion angle.

Next we have to assure that the codebook is adapted in the right direction (see Figure 5.2). For the SOM algorithm this requires that the input vector $x(t)$ or the old codebook vector $w_s(t)$, respectively, may need to be transformed first, before the new codebook vector $w_s(t+1)$ can be computed according to Eq. (3.5):

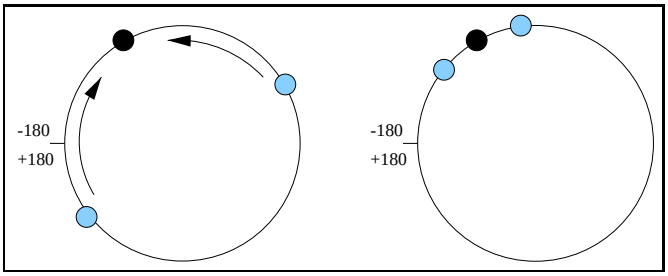


Figure 5.2: **Example: Adaptation of the codebook vector (grey) in direction of the input vector (black) on the shortest way.**

Cyclic Transformation Rules (SOM)

1. IF $w_{s_i}(t) \geq 0$ AND $x_i(t) < 0$ AND $\text{abs}(w_{s_i}(t)) + \text{abs}(x_i(t)) > \pi$
   THEN $x_i(t) := x_i(t) + 2\pi$

2. IF $w_{s_i}(t) < 0$ AND $x_i(t) \geq 0$ AND $\text{abs}(w_{s_i}(t)) + \text{abs}(x_i(t)) > \pi$
   THEN $w_{s_i}(t) := w_{s_i}(t) + 2\pi$

   Note that we have $\text{abs}(x) := \sqrt{x^2}$ for $x \in \mathbf{R}$.

After the new codebook vector has been computed, eventually it must also be transformed so that each component $W_{s_i}(t+1)$ is inside the interval $[-\pi, \pi]$. Figure 5.3 shows an one-dimensional example for the first case.

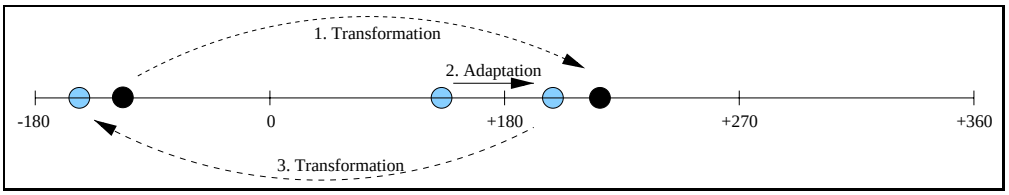


Figure 5.3: **Example: Transformations of the codebook vector (grey) and the input vector (black) to guarantee correct adaptation.**

To use cyclic data within the SOBM algorithm, we need more sophisticated rules, because we have to distinguish between normal and complementary intervals:

If $l_{s_i} < r_{s_i}$, we call $\hat{W}_{s_i} := [l_{s_i}, r_{s_i}]$ a normal interval. But we allow also the case $l_{s_i} > r_{s_i}$. In this case we have $\hat{W}_{s_i} := [-\pi, \pi] \setminus [r_{s_i}, l_{s_i}]$, i.e. $\hat{W}_{s_i}$ is the complementary interval of $[r_{s_i}, l_{s_i}]$.

First we have to refine function $g : [-\pi, \pi]^3 \to [0, 1]$ used within the codebook adaptation rules (see Eq. (3.7)):

Case 1: $a < b$. Set

$$g(a,b,x) := \begin{cases} 1 & \text{if } x \notin [a,b] \wedge d_i(x,a) \leq d_i(x,b) \\ 0 & \text{if } x \notin [a,b] \wedge d_i(x,a) > d_i(x,b) \\ \frac{b-x}{\iota([a,b])} & \text{else} \end{cases}$$

with $\iota([a,b]) := (b-a)$.

Case 2: $a > b$. Set

$$g(a,b,x) := \begin{cases} 1 & \text{if } x \in [b,a] \wedge d_i(x,a) \leq d_i(x,b) \\ 0 & \text{if } x \in [b,a] \wedge d_i(x,a) > d_i(x,b) \\ \frac{2\pi+(b-x)}{\iota([a,b])} & \text{if } x \notin [b,a] \wedge x \geq a \\ \frac{b-x}{\iota([a,b])} & \text{else} \end{cases}$$

with $\iota([a,b]) := 2\pi + (b-a)$.

Next we have to specify the necessary transformations to guarantee a correct adaptation of the codebook boxes:

Cyclic Transformation Rules (SOBM)

If $\hat{W}_{s_i}(t) := [l_{s_i}(t), r_{s_i}(t)]$ with $l_{s_i}(t) > r_{s_i}(t)$ or if $x_i(t)$ is not inside the complementary interval $\hat{W}_{s_i}(t)$, i.e. $x_i(t) \in [r_{s_i}(t), l_{s_i}(t)]$, then we have to consider the earlier defined cyclic transformation rules for the SOM algorithm, with $l_{s_i}(t)$ and $r_{s_i}(t)$ instead of $W_s(t)$. But if $x_i(t)$ is inside the complementary interval $\hat{W}_{s_i}(t)$, i.e. $x_i(t) \notin [r_{s_i}(t), l_{s_i}(t)]$, one has to consider slightly different transformation rules to assure that the boundaries are adapted towards the correct direction:

IF $g(l_{s_i}(t), r_{s_i}(t), x_i(t)) > g(-r_{s_i}(t), -l_{s_i}(t), -x_i(t))$ THEN
  Use the cyclic transformation rules (SOM) for the adaptation of $l_{s_i}(t)$.
  IF $x_i(t) > r_{s_i}(t)$ THEN
    First set $x_i(t) := x_i(t) - 2\pi$, afterwards adapt $r_{s_i}(t)$ directly
    (i.e. without further transformation).
  ELSE
    Adapt $r_{s_i}(t)$ directly.
ELSE
  Use the cyclic transformation rules (SOM) for the adaptation of $r_{s_i}(t)$.
  IF $x_i(t) < l_{s_i}(t)$ THEN
    First set $x_i(t) := x_i(t) + 2\pi$, afterwards adapt $l_{s_i}(t)$ directly.
  ELSE
    Adapt $l_{s_i}(t)$ directly.

If the width of the interval $[l_{s_i}(t), r_{s_i}(t)]$ is nearly 2π, then one observes sometimes the artifact that left and right boundaries interchange so that the interval becomes "too small". In this case the adaptation step has to be skipped and the interval $[-2\pi + \epsilon, 2\pi - \epsilon]$ has to be fixed as the new value of $\hat{W}_{s_i}(t+1)$.

5.1.3 Numerical results: HIV protease inhibitor

The fact that the cleavage of the HIV polyprotein by HIV protease is essential for viral propagation, has made the HIV protease a key target for the design of drugs against AIDS. The recent development of HIV protease inhibitors has dramatically improved the therapeutic outcome for many AIDS patients. Unfortunately, these inhibitors are very expensive and the effectiveness of therapy can encounter problems with drug-resistant viral strains. So there is further strong interest in the development of other classes of HIV protease inhibitors [10]. It is obvious that with a deeper understanding — including knowledge about the dynamic behavior — of the existing inhibitor molecules, it becomes much easier and cheaper to find and to design new inhibitor classes. In the following we present the numerical results of the conformational analysis of the HIV-protease inhibitor VX-478.

The inhibitor VX-478 of the enzyme HIV protease consists of 70 atoms. The molecule was parameterized by the Merck molecular force field (MMFF) [37]. Figure 5.4 shows one possible state (configuration) of the molecule.

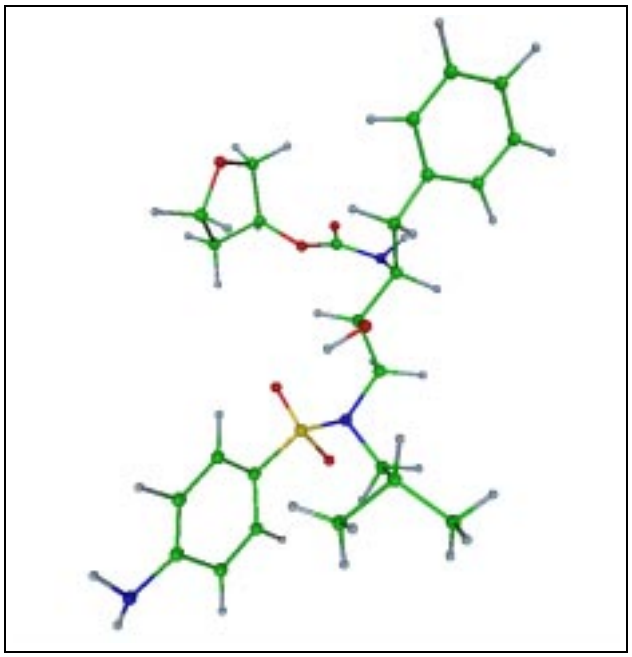


Figure 5.4: **Possible configuration of the HIV-protease inhibitor.**

As noted in Eq. (5.2), the sampling of a thermodynamic distribution at various temperatures within a temperature embedding can be realized by a correlated scaling of time steps and potential [58].

The Hybrid Monte Carlo (HMC) simulations are performed with temperature dependent time steps (fs = femtoseconds)

$$\tau = \frac{1.4}{\sqrt{\frac{300}{T[K]}}} \text{ fs}.$$

Each new configuration is generated by a propagation of the system over a random length between 40 and 80 time steps and each simulation consists of 5 independent Markov chains. For every configuration 34 torsion angles are stored which are sufficient for a rough reconstruction together with the corresponding equilibrium bonds and angles. Convergence of the HMC-simulation is reached, as soon as the Gelman and Rubin quotient R [34, 9] is sufficiently close to the value 1. Note that the choice of what is "sufficiently close to 1" is rather critical, because on the one hand one is interested in fast simulations, but on the other hand a worse convergence bears the risk of sampling not the whole configurational space. In [30] the focus was definitely on fast simulations, leading to a sampling of only parts of the configurational space. Together with a slight different choice of parameters[1] this has led to a detection of conformations even at rather high temperatures. In the following the results of simulations with much better convergence properties are presented, where the Gelman and Rubin quotient accomplishes the rigorous condition $\|1 - R\| \leq 0.05$.

Based on the five Markov chains we have constructed the data set V, the frequency function f and the homogeneity function h_S as described in section 1.1. The computation of the approximate box decomposition of V with respect to f was done automatically via a combination of the SOM and the SOBM algorithm with pruning and early stopping (see section 3.3+3.5). Note that the chosen parameters are comparable with the suggestions in the SOM literature [48]:

1. As an upper bound for the number of partitions Θ_s we have chosen an upper bound $\Bbbk := 600$, what is large enough to guarantee robust results, i.e. nearly equal results, if $\Bbbk$ is changed slightly.

2. The computation of a 25×24 SOM was done by performing $u \cdot \Bbbk$ ordering steps (with $\alpha(0) = 1.0$, $\eta := \eta_{gaussian}$ and $\gamma(0) = 12$) and $u \cdot \Bbbk$ convergence steps (with $\alpha(0) := 0.1$, $\eta = \eta_{bubble}$ and $\gamma(0) = 1$), where $u := 50$ denotes the average number of codebook updates.

[1] In [30] shorter time steps and a propagation of fixed length were used. This has reduced the flexibility of the molecular system.

3. We have initialized the SOBM codebook by using only the codebook vectors w_p with $f(\Theta_{w_p}(V)) \geq 2u$. Then we have performed convergence steps (with $\alpha(0) := 0.005$, $\eta := \eta_{bubble}$ and $\gamma(0) := 1$), until the overlap between the codebook boxes has exceeded 0.1%. We have used the final codebook to derive an approximate box decomposition of V according to Lemma 3.2.2.

Cluster identification

For the cluster identification, we have used our extended multilevel approach. First we look at the results, without decomposition refinement (see Table 5.1):

$\mathcal{T}$[K]	N	k	spectrum	coupling matrix	overlay [%]
900	60000	53	$\underline{1.000}$ 0.830 0.805 0.791	1.000	26.5
700	31000	72	1.000 $\underline{0.930}$ 0.885 0.876 0.860 0.795 0.790	0.924 0.076 0.018 0.982	40.5
700-C_0 RS	60000	65	1.000 0.890 0.820 0.798 0.768	1.000	35.5
700-C_1 RS	42000	92	1.000 0.896 0.875 0.824 0.820	1.000	36.4

Table 5.1: **Hierarchical temperature embedding for HIV protease inhibitor with resimulation at level** $\mathcal{T} = 700K$ (N = number of configurations per Markov chain, k = final number of codebook boxes).

While for $\mathcal{T} \geq 900K$ the Perron cluster analysis only identifies one conformation, one observes a large spectral gap between the second (0.930) and the third

(0.885) eigenvalue of the transition matrix $\mathcal{S}$ at level $\mathcal{T} = 700\,K$. To prove the metastability of the identified clusters C_0 and C_1, a resimulation at the same level was performed. As expected the gap between the 1 and the second eigenvalue grows for both clusters, but there are also large gaps between the second (0.890) and the third eigenvalue (0.820) for the first cluster and between the third (0.875) and the fourth (0.824) eigenvalue for the second cluster. But if one looks again at the original spectrum at level $\mathcal{T} = 700\,K$, one finds another large gap between the fifth (0.860) and the sixth (0.795) eigenvalue. Obviously the configurational space at level $\mathcal{T} = 700\,K$ decomposes into two strongly metastable clusters, but also into five weaker metastable subsets (see Table 5.2).

$\mathcal{T}$[K]	spectrum	coupling matrix					overlay [%]
700	1.000						40.5
	0.930	0.908	0.021	0.024	0.031	0.018	
	0.885	0.014	0.874	0.022	0.001	0.090	
	0.876	0.013	0.018	0.879	0.006	0.085	
	0.860	0.044	0.002	0.015	0.896	0.043	
	0.795	0.004	0.029	0.033	0.006	0.928	
	0.790						

Table 5.2: **Weaker metastability:** Five conformations for HIV protease inhibitor at level $\mathcal{T} = 700K$ (31000 configurations, 72 final codebook boxes).

Next we have refined the decomposition after step (2) and performed step (3), until the decomposition was fine enough. At level $\mathcal{T} = 700\,K$, we have achieved the results presented in Table 5.3.

The number of final codebook boxes has increased, leading to a larger second eigenvalue (0.952), a larger gap size and a better coupling matrix. Additionally the overlay has increased (47.7% in comparison with 40.5%), while the overlap still has remained near zero.

For a temperature embedded simulation at level $\mathcal{T} = 500\,K$ inside the both metastable clusters C_0 and C_1, our cluster method computes 4 $(C_{00}, C_{01}, C_{02}, C_{03})$ and 3 conformations (C_{10}, C_{11}, C_{12}) respectively (see Table 5.3). The seven identified conformations have weights $f(C_i)$ according to Table 5.4.

Figure 5.5 and Figure 5.6 show average configurations for always two out of the seven conformations at $\mathcal{T} = 500\,K$. To allow a better comparison the two average configurations are aligned in a plane defined by three common atoms.

$\mathcal{T}$[K]	N	k	spectrum	coupling matrix	overlay [%]
900	60000	53	$\underline{1.000}$ 0.830 0.805 0.791	1.000	26.5
700	31000	113	1.000 $\underline{0.952}$ 0.898 0.889 0.886 0.830 0.802 0.794	0.934 0.066 0.015 0.985	47.7
500-C_0	60000	101	1.000 0.962 0.949 $\underline{0.945}$ 0.917 0.903 0.896	0.921 0.015 0.040 0.023 0.012 0.920 0.023 0.044 0.034 0.024 0.919 0.023 0.010 0.024 0.012 0.954	51.8
500-C_1	60000	72	1.000 0.952 $\underline{0.942}$ 0.920 0.908 0.891	0.961 0.029 0.010 0.025 0.964 0.012 0.044 0.062 0.894	47.3

Table 5.3: **Hierarchical temperature embedding for HIV protease inhibitor with decomposition refinement** (N = number of configurations per Markov chain, k = final number of codebook boxes).

C_{00}	C_{01}	C_{02}	C_{03}	C_{10}	C_{11}	C_{12}
3.3%	4.1%	3.9%	7.4%	33.8%	40.1%	7.5%

Table 5.4: **Weights of the seven conformations for HIV protease inhibitor at level $\mathcal{T}$ = 500 K**

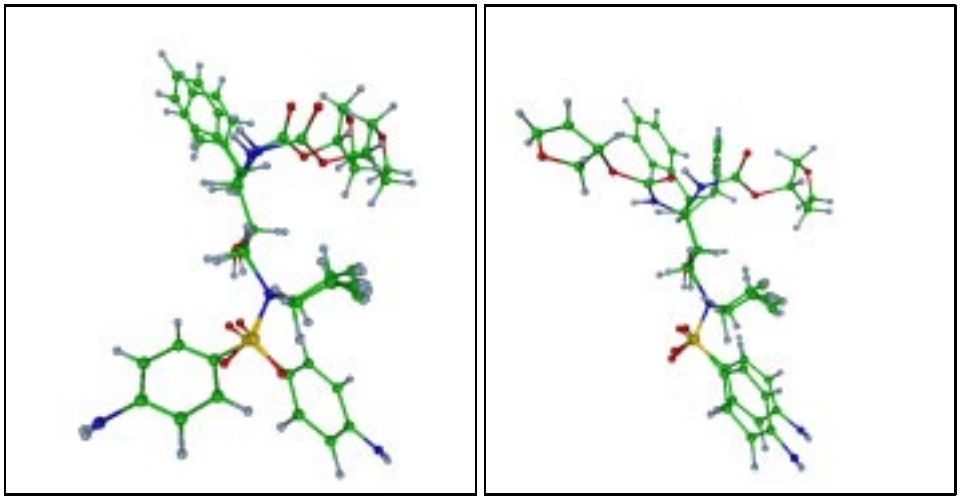


Figure 5.5: **Visualization of conformations of HIV protease inhibitor:** Average configurations for two metastable conformations at temperature level $\mathcal{T} = 500\,K$ (left: C_{00} and C_{02}, right: C_{02} and C_{11}).

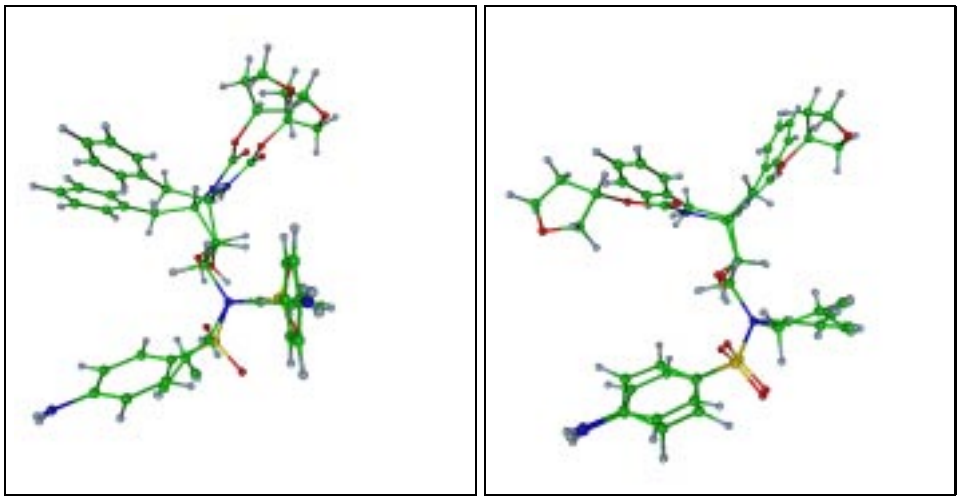


Figure 5.6: **Visualization of conformations of HIV protease inhibitor:** Average configurations for two metastable conformations at temperature level $\mathcal{T} = 500\,K$ (left: C_{01} and C_{03}, right: C_{01} and C_{10}).

For comparison purposes, we have also used mere VQ instead of SOM. In this case Perron Cluster analysis leads to four metastable clusters instead of the three conformations C_{10}, C_{11}, C_{12} at $T = 500K$. Upon careful examination of the results, however, one observes that one of the four clusters is nearly empty — this is the kind of pseudo-clusters already mentioned in chapter 3.

Cluster description

Using the corresponding approximate box decomposition (see Figure 5.7 for a projection of codebook boxes computed by the SOBM algorithm on two out of the 34 torsion angles), we have identified 17 discriminating torsion angles for the clustering $\mathcal{C} := \{C_0, C_1\}$ at $T = 700K$. Further we have used the corresponding 113 codebook boxes to determine reduced membership rules of C_0 and C_1. Here is one of these membership rules for cluster C_1:

IF $v_{*,3} \notin [18.9, 151.8]$ AND $v_{*,4} \notin [-169.4, -29.2]$ AND $v_{*,5} \notin [-82.3, 58.5]$ AND $v_{*,6} \notin [29.3, 168.0]$ AND $v_{*,7} \notin [-45.4, 94.6]$ AND $v_{*,8} \notin [-103.7, 29.0]$ AND $v_{*,15} \notin [-36.4, 99.9]$ AND $v_{*,16} \notin [-160.0, -22.5]$ AND $v_{*,17} \notin [-138.5, -10.1]$ AND $v_{*,18} \notin [-52.1, 67.7]$ AND $v_{*,19} \in [-61.8, 177.7]$ AND $v_{*,26} \in [-148.1, 77.0]$ AND $v_{*,27} \in [-158.1, 68.4]$ AND $v_{*,29} \in [-144.4, 89.2]$ AND $v_{*,30} \in [-110.5, 107.4]$ AND $v_{*,31} \in [-152.7, 76.5]$ AND $v_{*,32} \in [-99.3, 121.3]$ THEN $v = (v_{*,1}, \ldots, v_{*,34}) \in C_1$

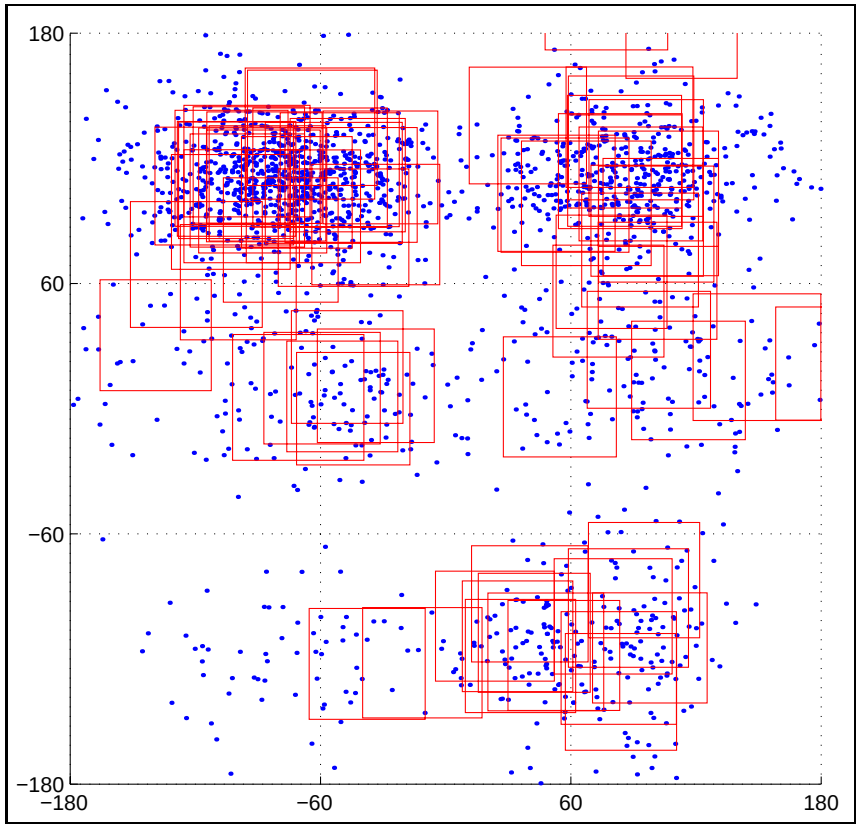


Figure 5.7: **Example: Adaptive box decomposition for HIV protease inhibitor.** Visualized projection of codebook boxes on two out of 34 torsion angles.

5.1.4 Prospect: Virtual screening

Clustering techniques and especially self-organized neural networks have been already used for the analysis of molecular dynamics [43, 41]. But all suggested algorithms have the deficit that they use a geometric cluster model: They try to group geometric conformations to metastable conformations by an investigation of a suitable visualization of the transition probabilities between the geometric conformations. Obviously such a procedure is only possible if the number of geometric conformations is very small, as it is only the case for simple molecules. In contradiction, the method described in the previous subsections is able to compute metastable conformations also for large and complex molecules. Therefore it can be used for a virtual screening of chemical databases.

Example: Virtual screening of CDK inhibitor

Virtual screening of chemical databases is a powerful tool for the identification of derivatives of already known molecules with a function of pharmaceutical interest. Figure 5.8 shows a virtual screening process for the *CDK inhibitor indirubin* in principle: First we have to perform a conformational analysis of indirubin and also of all molecules inside the database, to generate knowledge about their function. Then we have to use suitable matching algorithms (see [52]) to identify molecules inside the database with a similar structure and similar metastable conformations as the indirubin molecule. For a first application of conformational analysis within a virtual screening project see [30].

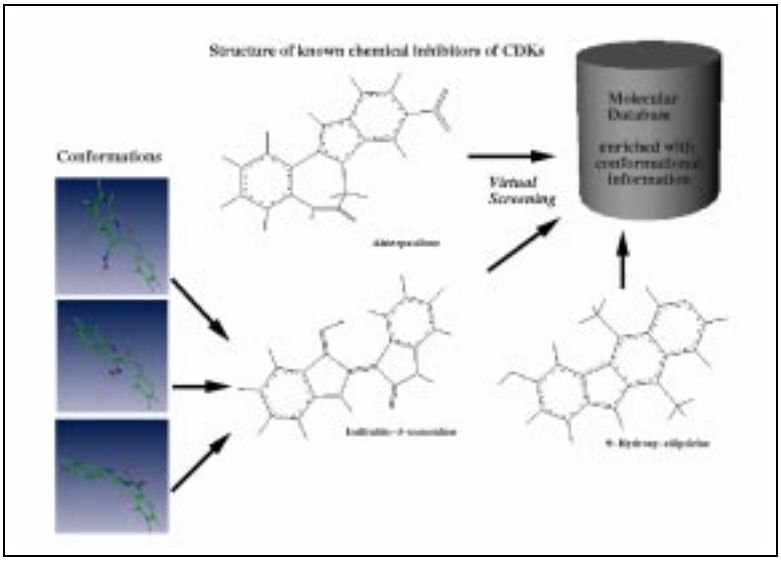


Figure 5.8: **Virtual Screening of CDK inhibitor indirubin.**

5.2 Cluster analysis of insurance customers

Cluster analysis is a powerful tool for insurance companies to get a better understanding of their customer structure, e.g., to design new tariffs or services. In the following we will present a successful applications of our new cluster approach for the analysis of insurance data that has been done in cooperation with RISK-CONSULTING, KÖLN. For a description of a further application see [31].

5.2.1 Modeling

Suppose that each insurance customer can be described by a set of q attributes, e.g., age, sex, occupation. As described in the appendix, we can easily transform the corresponding Ω to a normalized metric space and therefore the customers can be interpreted as points in a set $V \subset \Omega$. Since we want to identify groups of customers, who have similar properties with regard to the different attributes, we have to solve a geometric cluster problem. If the data quality is good, i.e. if we have for each customer valid values for nearly all attributes, we can use a homogeneity measure h_d based on the Euclidean distance function $d = d_{euclid}$. Otherwise we have to use more sophisticated distance measures as, e.g., the Tanimoto measure [48] or measures that use information levels [28]. Since each customer is unique, we use a frequency function f with $f(v) = 1$ for all $v \in V$. If the number of clusters is unknown a priori, we transform h_d into a stochastic homogeneity function $\widetilde{h}_d$ as described in Lemma 4.3.11 so that we can use our extended multilevel approach based on Perron Cluster analysis. Since we cannot be sure that the homogeneity function $\widetilde{h}_d$ corresponds to the same optimal clusters as the original homogeneity function h (see the earlier discussion in connection with Lemma 4.3.11), we have to validate the identified clusters carefully. This is especially necessary, if the artificial construction of $\widetilde{h}_d$ leads to a spectrum with much noise, i.e. a spectrum where the separation between the Perron Cluster and the remaining part is difficult. Obviously an efficient cluster description based on an approximate box decomposition is a helpful tool for cluster validation.

5.2.2 Numerical results: Whiplash Injury Patients

Within our application we have clustered 2153 customers of a German health insurance company with a diagnosis of *whiplash*[2] during the observation years 1996 and 1997. The number of attributes after transformation of Ω into a normalized metric space was 185.

[2]Whiplash (German: Schleudertrauma) is an injury to the cervical spine and its soft tissues caused by forceful flection of the neck, especially that occurring during an automobile accident.

The computation of an approximate box decomposition of V was done with a combination of the SOM and the SOBM algorithm as described in section 3.3. We have used early stopping, but we have not pruned neurons to allow a visual comparison with the results generated by using only the SOM algorithm.

1. As an upper bound for the number of partitions Θ_s we have chosen $\Bbbk := 99$, what is large enough to guarantee robust results, i.e. nearly equal results, if $\Bbbk$ is changed slightly.

2. The computation of a 11×9 SOM was done by performing $100\Bbbk$ ordering steps (with $\alpha(0) = 0.9$, $\eta := \eta_{gaussian}$ and $\gamma(0) = 5$) and $300\Bbbk$ convergence steps (with $\alpha(0) := 0.1$, $\eta = \eta_{bubble}$ and $\gamma(0) = 1$).

3. Using the codebook vectors w_p, we have initialized the SOBM codebook boxes. Then we have performed convergence steps (with $\alpha(0) := 0.005$, $\eta := \eta_{bubble}$ and $\gamma(0) := 1$), until the overlap between the codebook boxes has exceeded the value 0.1%.

In a first trial, we have stopped after step (2). We have used the codebook vectors w_p to determine a decomposition of V and performed a Perron Cluster analysis (see Table 5.5):

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	$\Gamma_{f,h_d}(k=3)$	$\Gamma_{f,h_d}(k=5)$
1.00	0.81	0.72	0.60	0.51	0.38	0.34	0.71	0.60

Table 5.5: **Whiplash Patients:** Perron Cluster analysis using 9×11 SOM.

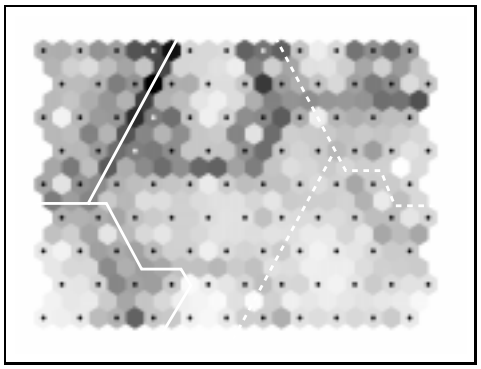


Figure 5.9: **Whiplash Patients:** SOM gray-level visualization including cluster borders computed via Perron cluster analysis (solid border: clusters for $k = 3$, dashed border: two additional clusters for $k = 5$).

The two largest gaps are between λ_3 and λ_4 and between λ_5 and λ_6 respectively. Figure 5.9 shows the borders of the computed clusters within a SOM gray-level visualization[3].

Next we have performed additionally step (3). We have computed an approximate box decomposition of V based on the final codebook boxes and we have used Perron Cluster analysis to determine an optimal clustering:

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	$\Gamma_{f,h_d}(k=3)$	$\Gamma_{f,h_d}(k=5)$
1.00	0.81	0.73	0.62	0.54	0.43	0.35	0.69	0.62

Table 5.6: **Whiplash Patients:** Perron Cluster analysis using 9×11 SOBM.

The algorithm suggests 3 or 5 clusters. Since we have not pruned neurons after step (2), we can visualize the SOBM with gray-levels (see Figure 5.10). The borders computed via Perron Cluster analysis corresponds to the the borders indicated by the dark-shades. Especially the right upper cluster is clearly identified. This cluster contains customers that has been taken over by the insurance company from another company many years ago. It is very interesting that these customers have been grouped together, because we have not used the corresponding attribute within our analysis, i.e. the information "customer has been overtaken" was not given explicitly. Nevertheless there exists a strong relationship between these customers, hidden inside the used attributes. Our cluster algorithm was able to detect these relationship and therefore has generated knowledge.

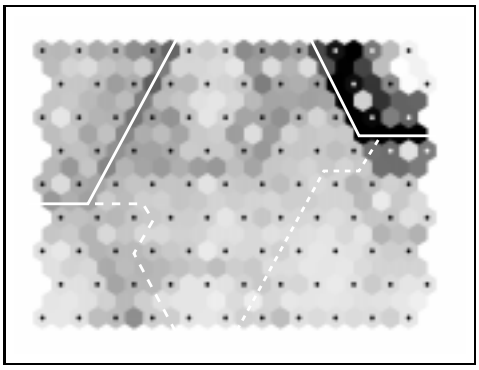


Figure 5.10: **Whiplash Patients:** SOBM visualization including cluster borders computed via Perron cluster analysis (solid border: clusters for $k = 3$, dashed border: two additional clusters for $k = 5$).

[3]SOM gray-level visualization is used to determine the clusters by visual investigation (see [48]). Dark shades represent low homogeneity between the codebook vectors, while light shades represent a high homogeneity . Other techniques for cluster visualization are presented in [61]

Conclusion

This thesis deals with a new and rather general multilevel approach for cluster analysis in high-dimensional data. In contrast to known cluster methods it applies not only for geometric, but also for dynamic cluster problems.

To guarantee the applicability to large cluster problems, the cluster identification is done via a decomposition based representative clustering method. If the underlying decomposition is fine enough, this method allows a problem reduction without destroying the original cluster structure. Furthermore, an efficient cluster description becomes possible if one uses a special decomposition variant, called approximate box decomposition. The computation of a suitably fine decomposition is done by a self-organized neural network.

Upon using the theory of Perron Cluster analysis, the general multilevel cluster approach can been extended: For cluster problems with a stochastic homogeneity function it allows to compute a correct set of clusters, even if their number is unknown a priori. Since traditional cluster methods need the number of clusters as an input, this is a significant improvement. Furthermore, the extended approach allows for the first time a conformational analysis of large biomolecules in combination with hierarchical temperature embedding.

On the computational complexity side, the computation of a suitably fine decomposition is still the *bottleneck*. Especially for an application within commercial virtual screening projects, a speed-up will be necessary. In this respect, parallelization and an improved convergence of the SOM/SOBM algorithm seem to be promising.

Appendix

Extension of Ω to a normalized metric space

Let $\mathcal{A} := \{A_1, \ldots, A_q\}$ be a set of not necessarily ordered domains and define $\Omega := \bigotimes_{i=1}^{q} A_i := \{(a_1, \ldots, a_q)^T \,|\, a_i \in A_i\,, i = 1, \ldots, q\}$. Further let $V \subset \Omega$ be any finite subset of Ω.

We suppose that any attribute A_i is finite or at least bounded. Otherwise we replace it by $A_i(V) := \{x \in A_i \,|\, (\exists\, v = (v_{*,1}, \ldots, v_{*,q})^T \in V)\, v_{*,i} = x\}$. We define an unique projection π from Ω into a normalized metric space, as follows:

1. Let A_i any attribute of Ω with $A_i = \{x_{i,1}, \ldots, x_{i,m_i}\} \nsubseteq \mathbf{R}$, $m_i \in \mathbf{N}$. For $1 \leq j \leq m_i$ set $A_{i,j} := \{0, 1\}$ and define $\pi_i : A_i \longrightarrow \bigotimes_{j=1}^{m_i} A_{i,j} \subset \mathbf{R}^{m_i}$ via

$$\pi_i(x_{i,j}) := (\delta_{i,1}, \ldots, \delta_{i,m_i})^T \;\text{ for}\, j = 1, \ldots, m_i$$

   with

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else.} \end{cases}$$

2. Let A_i any attribute of Ω with $A_i \subset [l_i, r_i] \subset \mathbf{R}$ and $l_i, r_i \in \mathbf{R}$. Set $A_{i,1} := [0, 1]$, $m_i := 1$ and define $\pi_i : A_i \longrightarrow A_{i,1} \subset \mathbf{R}$ via

$$\pi_i(x) := \frac{x - l_i}{r_i - l_i} \;\text{ for}\, x \in A_i.$$

Then $\pi := (\pi_1, \ldots, \pi_q)^T$ is a projection from Ω into a $\dot{q} := \sum_{i=1}^{q} m_i$ dimensional normalized subspace $\Omega_{\mathbf{R}} := \bigotimes_{i=1}^{q} \bigotimes_{j=1}^{m_i} A_{i,j} \subset \mathbf{R}^{\dot{q}}$.

Obviously we have: $\pi(v) = \pi(w) \iff v = w$ for all $v, w \in \Omega$.

Symbols

General Notation

$\|M\|$	number of objects in a finite set M
$\|\| \cdot \|\|$	Euclidean distance

Sets

$\mathbf{N}$	natural numbers
$\mathbf{R}, \mathbf{R}_0^+$	real numbers, positive real numbers including zero
$A_j, \mathcal{A}$	attribute, finite set of attributes
Ω	direct product of attributes
V	data set
$C_i, \mathcal{C}$	cluster, k-cluster set (finite set of disjoint clusters)
$\wp(\Omega)$	power set of Ω
I, J	index subset
$\mathcal{A}(J)$	reduced set of attributes (only A_j with $j \in J$)
$\Omega(J)$	direct product of attributes in $\mathcal{A}(J)$
$V(J)$	canonical projection of V on $\Omega(J)$
Θ_s	partition
Θ	decomposition (finite set of disjoint partitions)
B_j	subset of attribute A_j
B, Δ_s	box
Δ, Δ_I	set of boxes, reduced set of boxes (only Δ_s with $s \in I$)
W	codebook
$\mathcal{C}(W)$	compressed clustering
$\hat{\mathcal{C}}$	extended clustering
Θ_W	decomposition based on SOM codebook
$\hat{W}_s$	codebook box

Matrices

$\mathcal{S}$	stochastic matrix
$\hat{\mathcal{S}}$	coupling matrix
$\mathcal{D}$	weighting matrix

Variables

q	dimension of Ω
v, v_i	data object in V
n	number of data objects in V
k	number of clusters
$v(J)$	projection of v on $\Omega(J)$
n_k	number of decomposition partitions
w_s	codebook vector
T, L	time steps
$\mathbb{k}$	upper bound of n_k
z_s	grid position of neuron s
l_i, r_i	left and right boundaries of interval in $\mathbf{R}$
X	random variable
u	average number of codebook updates
λ_i, Y_i	eigenvalue, eigenvector

Functions

f	frequency function
h	homogeneity function
$h_{max}(V)$	maximal value of homogeneity function in V
$\Gamma_{f,h}$	weighted intra-cluster homogeneity
d	distance function
h_d	homogeneity function based on distance function
S	conditional transition probability function
$\hat{S}$	set extension of S
h_S	homogeneity function based on transition probability function
χ_M	characteristic function of set M
r	membership rule (set)
$\vartheta_{f,h}$	decomposition error
$\check{f}$	compressed frequency function
$\check{h}$	compressed homogeneity function
$\hat{f}$	set extension of f
$\hat{h}$	set extension of h
ρ	probability density function
P_ρ	probability function corresponding to ρ
α	learning rate
γ	neighborhood radius function
η	grid distance function
$E(X)$	conditional expectation value of X
P	weighted homogeneity function
$\hat{P}$	set extension of P

Bibliography

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 94–105, 1998.

[2] M.P. Allen and D.J. Tildesley. *Computer Simulations of Liquids*. Clarendon Press, Oxford, 1990.

[3] N. Allinson, H. Yin, L. Allinson, and J. Slack (editors). *Advances in Self-Organizing Maps*. Springer, 2001.

[4] A. Amadei, A.B.M. Linssen, and H.J.C. Berendsen. Essential dynamics of proteins. *Proteins*, 17, 1993.

[5] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[6] J. Bezdek and N. Pal. Some new indexes of cluster validity. *IEEE Trans. Syst. Man. Cybern.*, 28:301–315, 1998.

[7] Ch.M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[8] H. Bothe and R. Rojas (editors). *Proceedings of the 2nd International ICSC Symposium on Neural Computation*. ICSC Academic Press, 2000.

[9] S. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *J. Comp. Graph. Stat.*, (7):434–455, 1998.

[10] D. R. Corey. Design and engineering of proteins as therapeutic agents. In *[63]*, pages 187–204.

[11] M. Cottrell, E. de Bodt, and M. Verleysen. Kohonen maps versus vector quantization for data analysis. In *Proc. of European Symposium on Artifical Neural Networks (ESANN)*. D-Facto, Brussel, 1997.

[12] M. Cottrell, J. Fort, and G. Pages. Theoretical aspects of the som algorithm. *Neurocomputing*, (21):119–138, 1998.

[13] P. Deuflhard and A.Hohmann. *Introduction to Scientific Computing*. Springer, 2nd edition, 2002.

[14] P. Deuflhard, M. Dellnitz, O. Junge, and Ch. Schütte. Computation of essential molecular dynamics by subdivision techniques. In *[15]*.

[15] P. Deuflhard, J. Hermans, B. Leimkuhler, A.E. Mark, S. Reich, and R.D. Skeel (editors). *Computational Molecular Dynamics: Challenges, Methods, Ideas. Lecture Notes in Computational Science and Engineering*, volume 4. Springer, 1998.

[16] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in nearly uncoupled markov chains. *Linear Algebra and its Applications 315*, pages 39–59, 2000.

[17] M. Dittenbach, D. Merkl, and A. Rauber. The growing hierarchical self-organizing map. In *Proc. International Joint Conf. on Neural Networks (IJCNN), Como, Italy*, volume 6, pages 15–19. IEEE Computer Society, 2000.

[18] B.S. Duran and P.L. Odell. *Cluster Analysis*. Springer, 1974.

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. Incremental clustering for mining in a data warehousing environment. In *Proc. 24th Int. Conf. on Very Large Databases (VLDB 98), New York City*, pages 323–333, 1998.

[20] B.S. Everitt. *Cluster Analysis*. Arnold, 3rd edition, 1993.

[21] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (editors). *Advances in Knowledge Discovery and Data Mining*. AAAI Press / The MIT Press, California, 1996.

[22] A. Fischer, F. Cordes, and Ch. Schütte. Hybrid Monte Carlo with adaptive temperature choice: efficient conformational analysis of RNA. *Comput. Phys. Comm.*, 121-122:37–39, 1998.

[23] A. Fischer, Ch. Schütte, P. Deuflhard, and F. Cordes. Hierarchical uncoupling-coupling of metastable conformations. In *[57]*. Available as ZIB-Report 01-03 via http://www.zib.de/bib/pub/pw.

[24] N.I. Fisher. *Statistical Analysis of Circular Data*. University Press, Cambridge, 1993.

[25] C. Fraley and A. E. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *Computer Journal*, (41):578–588, 1998.

[26] B. Fritzke. Growing grid - a self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 2(5):9–13, 1995.

[27] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[28] T. Galliat. Clustering data of different information levels. Preprint SC-99-42, Konrad–Zuse–Zentrum, Berlin. Available via http://www.zib.de/DataMining, 1999.

[29] T. Galliat and P. Deuflhard. Adaptive hierarchical cluster analysis by self-organizing box maps. ZIB-Report 00-13, Konrad–Zuse–Zentrum, Berlin. Available via http://www.zib.de/bib/pub/pw/, 2000.

[30] T. Galliat, P. Deuflhard, R. Roitzsch, and F. Cordes. Automatic identification of metastable conformations via self-organized neural networks. In *[57]*. Available as ZIB-Report 00-51 via http://www.zib.de/bib/pub/pw/.

[31] T. Galliat, W. Huisinga, and P. Deuflhard. Self-organizing maps combined with eigenmode analysis for automated cluster identification. In *[8]*, pages 227–232.

[32] V. Ganti, R. Ramakrishnan, J. Gehrke, A. Powell, and J. French. Clustering large datasets in arbitrary metric spaces. In *Proc. 15th International Conf. on Data Engineering, Sydney, Australia*, pages 502–511. IEEE Computer Society, 1999.

[33] M. Garey and D. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. Freeman, New York, 1979.

[34] A. Gelman and D.B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, (7):457–511, 1992.

[35] A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[36] D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. In *Proc. 24th Int. Conf. on Very Large Databases (VLDB 98), New York City*, pages 311–323, 1998.

[37] T.A. Halgren. Merck molecular force field.i-v. *J. Comp. Chem.*, 17(5&6):490–641, 1996.

[38] T. Heskes. Energy functions for self-organizing maps. In *[50]*, pages 303–316.

[39] W. Huisinga, C. Best, R. Roitzsch, C. Schütte, and F. Cordes. From simulation data to conformational ensembles: Structure and dynamic based methods. *J. Comp. Chem.*, 20(16):1760–1774, 1999.

[40] M. Van Hulle. *Faithful Representations and Topographic Maps*. John Wiley Sons, Inc., 2000.

[41] M. Hyvönen, Y. Hiltunen, W. El-Deredy, T. Ojala, J. Vaara, P. Kovanen, and M. Ala-Korpela. Application of self-organizing maps in conformational analysis of lipids. *J. Am. Chem. Soc.*, 123(5):810–816, 2001.

[42] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[43] M. Karpen, D. Tobias, and C. Brooks. Statistical clustering techniques for the analysis of long molecular dynamics trajectories. *Biochemistry*, 32(2):412–420, 1993.

[44] S. Kaski. *Data Exploration Using Self-Organizing Maps*. PhD thesis, Helsinki University of Technology, 1997.

[45] L. Kato. *Pertubation Theory for Linear Operators*. Springer, 1995.

[46] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, 1990.

[47] T. Kohonen. Comparison of som point densities based on different criteria. *Neural Computation*, (11):2081–2095, 1999.

[48] T. Kohonen. *Self-Organizing Maps*. Springer, 3rd edition, 2001.

[49] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, Beverly Hills, CA, 1978.

[50] E. Oja and S. Kaski. *Kohonen Maps*. Elsevier, Amsterdam, 1999.

[51] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proc. 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann, San Francisco, 2000.

[52] I. Rigoutsos, D. Platt, A. Califano, and D. Silverman. Representing and matching of small flexible molecules in large databases of 3d molecular information. In *[62]*.

[53] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[54] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps*. Addison-Wesley, 1992.

[55] R. Rojas. *Neural Networks - A Systematic Introduction*. Springer, 1996.

[56] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transcations on Computers*, C-18(5):401–409, 1969.

[57] T. Schlick and H. H. Gan (editors). *Computational Methods for Macromolecules: Challenges and Applications — Proc. of the 3rd Intern. Workshop on Algorithms for Macromolecular Modelling, New York, 2000.* Springer, 2002, In Press.

[58] Ch. Schütte. *Conformational Dynamics: Modelling, Theory, Algorithm, and Application to Biomolecules*. Habilitation Thesis, Dept. of mathematics und computer science, Free University Berlin, 1998. Available as ZIB-Report SC-99-18 via http://www.zib.de/bib/pub/pw/.

[59] Ch. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard. A direct approach to conformational dynamics based on hybrid Monte Carlo. *J. Comput. Phys., Special Issue on Computational Biophysics*, 151:146–168, 1999.

[60] R. Varga. *Matrix iterative analysis*. Springer, 2nd edition, 2000.

[61] J. Vesanto. Som-based data visualization methods. *Intelligent Data Analysis*, (3):111–126, 1999.

[62] J. Wang, B. Shapiro, and D. Shasha. *Pattern Discovery in Biomolecular Data*. Oxford University Press, 1999.

[63] S. Wu-Pong and Y. Rojanasakul. *Biopharmaceutical Drug Design and Development*. Humana Press, 1999.

[64] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, pages 103–114. ACM Press, 1996.

Zusammenfassung

Als Cluster Analyse bezeichnet man den Prozess der Suche und Beschreibung von Gruppen (Clustern) von Objekten, so daß die Objekte innerhalb eines Clusters bezüglich eines gegebenen Maßes maximal homogen sind. Die Homogenität der Objekte hängt dabei direkt oder indirekt von den Ausprägungen ab, die sie für eine Anzahl festgelegter Attribute besitzen. Die Suche nach Clustern läßt sich somit als Optimierungsproblem auffassen, wobei die Anzahl der Cluster vorher bekannt sein muß. Wenn die Anzahl der Objekte und der Attribute groß ist, spricht man von komplexen, hoch-dimensionalen Cluster Problemen. In diesem Fall ist eine direkte Optimierung zu aufwendig, und man benötigt entweder heuristische Optimierungsverfahren oder Methoden zur Reduktion der Komplexität. In der Vergangenheit wurden in der Forschung fast ausschließlich Verfahren für geometrisch basierte Clusterprobleme entwickelt. Bei diesen Problemen lassen sich die Objekte als Punkte in einem von den Attributen aufgespannten metrischen Raum modellieren; das verwendete Homogenitätsmaß basiert auf der geometrischen Distanz der den Objekten zugeordneten Punkte. Insbesondere zur Bestimmung sogenannter metastabiler Cluster sind solche Verfahren aber offensichtlich nicht geeignet, da metastabile Cluster, die z.B. in der Konformationsanalyse von Biomolekülen von zentraler Bedeutung sind, nicht auf einer geometrischen, sondern einer dynamischen Ähnlichkeit beruhen.

In der vorliegenden Arbeit wird ein allgemeines Clustermodell vorgeschlagen, das zur Modellierung geometrischer, wie auch dynamischer Clusterprobleme geeignet ist. Es wird eine Methode zur Komplexitätsreduktion von Clusterproblemen vorgestellt, die auf einer zuvor generierten Komprimierung der Objekte innerhalb des Datenraumes basiert. Dabei wird bewiesen, daß eine solche Reduktion die Clusterstruktur nicht zerstört, wenn die Komprimierung fein genug ist. Mittels selbstorganisierter neuronaler Netze lassen sich geeignete Komprimierungen berechnen. Um eine signifikante Komplexitätsreduktion ohne Zerstörung der Clusterstruktur zu erzielen, werden die genannten Methoden in ein mehrstufiges Verfahren eingebettet. Da neben der Identifizierung der Cluster auch deren effiziente Beschreibung notwendig ist, wird ferner eine spezielle Art der Komprimierung vorgestellt, der eine Boxdiskretisierung des Datenraumes zugrunde liegt.

Diese ermöglicht die einfache Generierung von regelbasierten Clusterbeschreibungen. Für einen speziellen Typ von Homogenitätsfunktionen, die eine stochastische Eigenschaft besitzen, wird das mehrstufige Clusterverfahren um eine Perroncluster Analyse erweitert. Dadurch wird die Anzahl der Cluster, im Gegensatz zu herkömmlichen Verfahren, nicht mehr als Eingabeparameter benötigt. Mit dem entwickelten Clusterverfahren kann erstmalig eine computergestützte Konformationsanalyse großer, für die Praxis relevanter Biomoleküle durchgeführt werden. Am Beispiel des *HIV Protease Inhibitors VX-478* wird dies detailliert beschrieben.

Lebenslauf

Persönliche Daten

Name:	Galliat
Vorname:	Tobias
geboren am:	17.11.1972 in Köln
Familienstand:	ledig
Konfession:	röm.-kath.
Staatsangehörigkeit:	deutsch

Ausbildung

1979 - 1992	Schulbesuch (Abitur: 06/1992)
10/1992 - 09/1999	Informatikstudium mit Nebenfach BWL, FernUniversität Hagen (Vordiplom: 08/1995, Diplom: 08/1999)
10/1993 - 09/1998	Mathematikstudium, Universität zu Köln (Vordipolm: 10/1995, Diplom: 07/1998)
10/1996 - 12/1996	Auslandstrimester an der Universität Paris-Orsay, Frankreich, im Rahmen eines Erasmus-Stipendiums

Beruflicher Werdegang

07/1992 - 09/1993	Zivildienst, “Referat für interreligiösen Dialog” im Erzbistum Köln
07/1995 - 09/1996	Studentische Hilfskraft, Risk-Consulting, Prof. Dr. Weyer, Köln
01/1997 - 12/1998	Studentischer Mitarbeiter, Risk-Consulting, Prof. Dr. Weyer, Köln
01/1999 - 09/1999	Wissenschaftlicher Mitarbeiter, Risk-Consulting, Prof. Dr. Weyer, Köln
04/1999 - 09/1999	Übungsbetreuung und Vorlesungsvertretung zum Thema “Neuronale Netze” an der Universität zu Köln (zusammen mit Herrn Prof. Dr. Weyer)
seit 10/1999	Wissenschaftlicher Angestellter, Konrad-Zuse-Zentrum für Informationstechnik Berlin