Self-organised clustering as a basis for cognition and machine intelligence

Author(s):
Ott, Thomas Matthias

Publication Date:
2007

Permanent Link:
https://doi.org/10.3929/ethz-a-005516533

Rights / License:
In Copyright - Non-Commercial Use Permitted
Self-organised Clustering as a Basis for Cognition and Machine Intelligence

A dissertation submitted to the

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

Thomas Matthias Ott
Dipl. phys. ETH Zürich
born 03. 04. 1977
citizen of Basadingen TG

accepted on the recommendation of

PD Dr. Ruedi Stoop, Examiner, Supervisor
Prof. Dr. Rodney Douglas, Co-Examiner
Prof. Dr. W.-H. Steeb, Co-Examiner
Prof. Dr. S. Stramaglia, Co-Examiner

2007
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Zusammenfassung</strong></td>
<td>v</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>vi</td>
</tr>
<tr>
<td><strong>Prologue</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>1 Fundamentals of Clustering</strong></td>
<td>5</td>
</tr>
<tr>
<td>1.1 Clustering and Computation</td>
<td>5</td>
</tr>
<tr>
<td>1.1.1 Measuring Computation Performed by Clustering Algorithms</td>
<td>6</td>
</tr>
<tr>
<td>1.1.2 Principle of Optimised Class-Based Predictions</td>
<td>8</td>
</tr>
<tr>
<td>1.1.3 Information Bottleneck Method</td>
<td>11</td>
</tr>
<tr>
<td>1.1.4 Fundamental Clustering as Non-Turing Computation</td>
<td>13</td>
</tr>
<tr>
<td>1.2 Cognitive Aspects of Clustering</td>
<td>16</td>
</tr>
<tr>
<td>1.2.1 Clustering Serves Concept Acquisition: A Toy Example</td>
<td>18</td>
</tr>
<tr>
<td>1.2.2 The Language of ‘Possible Actions or Routes’</td>
<td>24</td>
</tr>
<tr>
<td>1.2.3 Recapitulation: Cognitive Aspects of Clustering</td>
<td>28</td>
</tr>
<tr>
<td>1.3 Neural Models of Concept Acquisition</td>
<td>28</td>
</tr>
<tr>
<td>1.3.1 (I) Concept Acquisition</td>
<td>29</td>
</tr>
<tr>
<td>1.3.2 Symbolic Reasoning (II) and Non-symbolic Processes (III)</td>
<td>38</td>
</tr>
<tr>
<td>1.4 Some Philosophical Reflections</td>
<td>40</td>
</tr>
<tr>
<td><strong>2 Belief Propagation BeP</strong></td>
<td>41</td>
</tr>
<tr>
<td>2.1 Probabilistic Reasoning and Neurodynamics</td>
<td>41</td>
</tr>
<tr>
<td>2.1.1 Markov Random Fields</td>
<td>43</td>
</tr>
<tr>
<td>2.1.2 Belief Propagation for Binary Pairwise Markov Random Fields</td>
<td>44</td>
</tr>
<tr>
<td>2.1.3 Belief Propagation and the Neurodynamics of Hopfield Networks</td>
<td>45</td>
</tr>
<tr>
<td>2.1.4 Biological Plausibility and Non-symbolic Computation</td>
<td>50</td>
</tr>
<tr>
<td>2.2 Convergence of Belief Propagation on Ising Systems</td>
<td>52</td>
</tr>
<tr>
<td>2.2.1 Belief Propagation on Ising Grids: Fundamentals</td>
<td>53</td>
</tr>
<tr>
<td>2.2.2 Belief Propagation on Regular Periodic Grids</td>
<td>55</td>
</tr>
<tr>
<td>2.2.3 Convergence to a Fixed Point</td>
<td>55</td>
</tr>
<tr>
<td>2.2.4 Convergence to a Cycle</td>
<td>58</td>
</tr>
<tr>
<td>2.2.5 Critical Slowing Down</td>
<td>61</td>
</tr>
<tr>
<td>2.2.6 Recapitulation and Conclusions</td>
<td>62</td>
</tr>
</tbody>
</table>
CONTENTS

3 Clustering Algorithms 65
3.1 A very Short Survey of Clustering Algorithms 66
3.1.1 Clustering with Cost Functions 68
3.1.2 Clustering through Self-organisation 71
3.2 Sequential Superparamagnetic Clustering SSC 71
3.2.1 Superparamagnetic Clustering 72
3.2.2 Swendsen-Wang Algorithm and BeP for SSC 73
3.2.3 The Sequential Procedure 74
3.3 SSC Applications 77
3.3.1 A Toy System 77
3.3.2 Classifying Chemical Compounds 78
3.3.3 SSC: Why Does It Work? 81
3.3.4 Visual Scene Analysis Application 84
3.3.5 Spike Sorting with SSC 85
3.3.6 Spike train classification 91
3.3.7 Auditory Scene Analysis 93
3.4 SSC Recapitulation 93

A 95
A.1 Proof of Theorem 1.1 95
A.2 Attractor Clustering in Recurrent Networks 96
A.3 Bethe-Peierls Approximation 97
A.3.1 Bethe Free Energy 98
A.4 Frobenius-Perron Theorem 100

Figures 101
Tables 103
Bibliography 105
Index 110
Acknowledgements 112
List of Publications 113
Curriculum Vitae 114
Zusammenfassung


Im zweiten Teil dieser Arbeit wenden wir uns probabilistischen Modellen des Denkens zu: Im Alltag treffen wir häufig Entscheidungen, die sich auf Wahrscheinlichkeitsabschätzungen stützen (stochastische Inferenz). Solche Abschätzungen lassen sich mathematisch durch Prozesse des Nachrichtenaustausches (message passing) und der lokalen Nachrichtenverarbeitung in rekurrenten Netzwerken modellieren. Belief Propagation ist einer dieser Nachrichtenaustausch-Algorithmen, der für die Berechnung von marginalen Wahrscheinlichkeiten weite Verbreitung gefunden hat. Aufgrund der Ähnlichkeit dieser Prozesse mit Prozessen in Modellen von neuronalen Netzwerken stellen wir die Hypothese auf, dass die neuronale Aktivität in rekurrenten Teilnetzen allgemein als eine stochastische Inferenz verstanden werden

Summary

Structuring our world in terms of categories such as chair and table is fundamental, as it allows for simple assumptions and predictions about the environment: on a chair we can sit, at a table we can eat and from a cup we may drink. According to which criteria do we partition the world into classes and categories? How could we implement similar mechanisms on computers? What potentials for technical applications result from this? This thesis is inspired by basic questions of this kind, in particular with regard to machine learning. In order to find classes within a data set, there is a plethora of clustering algorithms. This PhD thesis intends to contribute to the development of clustering algorithms and to shed light on the cognitive aspects of clustering.

In the first part, we will approach the problem of clustering from an abstract perspective and ask about the cognitive relevance of clustering for humans. Generally, by clustering we denote the partition of a set of entities (e.g., events, observations, objects or data items) into classes. For humans, this operation is understood as a fundamental cognitive act that allows an efficient representation of the world by means of conceptual classes such as ‘cup’. We will speculate on how such acts could be implemented by inventing a few simple models that describe clustering as a self-organised learning process in a neural network. In this approach, we do not aim to describe actual brain structures or brain functions, but we will point out principles for a neural implementation of concept acquisition. Based on these considerations, we will think about potential neural mechanisms of simple symbolic reasoning.

In the second part, we will discuss models of probabilistic reasoning (stochastic inference). Probabilistic reasoning can be captured mathematically as a process of message passing and local computation in recurrent networks. Due to the similarity of these processes to processes described by models of neural networks, we will hypothesise that the neural activity in recurrent neural circuits could generally be understood as stochastic inference. We will motivate this hypothesis by deriving a mathematical relation between continuous Hopfield networks and belief propagation which is a message passing algorithm for stochastic inference. At the end of the second part, we will conduct a thorough analysis of belief propagation on ferromagnetic Ising spin systems, yielding conditions for convergence.

In the third part, we will finally bring together belief propagation and aspects of self-organised learning for the design of a clustering algorithm. We will introduce the sequential superparamagnetic clustering algorithm that has been developed within this PhD project. The algorithm answers the question of how clusters of highly different densities can be detected automatically. Numerous applications in different fields, such as combinatorial chemistry, scene analysis, and neuroscience, will demonstrate the utility of this clustering algorithm as a valuable tool for data analysis.
Prologue

Motivation

When Christopher Columbus discovered the New World, he found himself confronted with numerous things he had never seen before. There were ‘strange’ people, unfamiliar animals and exotic plants. He would not have been able to name a single species in this strange world, and yet Christopher was not lost. He saw that there was a clear order in this world. He could easily distinguish between people and animals or animals and plants. This was a distinction he knew from home. He also found it easy to distinguish between birds and mammals, whereas in the case of the trees he could hardly make any distinction between different species at the beginning. However, after a while, he learnt to see the differences; some trees had small leaves, other trees had spiky leaves - after some days, Christopher could have easily reported on dozens of different types of trees. Additional insight was provided by the natives. They knew the names and the values of many trees and herbs. Some of them attracted Christopher’s attention and, in particular, the attention of those who came after him: the gourmets among them liked the so-called potatoes and the bon vivants appreciated a plant called tobacco.

When Christopher Columbus set foot in the New World, he was not lost because his brain knew how to structure the world. It was able to distinguish between different objects and it could group these objects into categories. Obviously, Christopher used diverse strategies for categorisation. The distinction between animals and plants was made on the basis of the knowledge he already had about these categories. For this categorisation he used an approved classification rule. Some of the plants were shown to him by the natives. They taught him the names of these plants and how to recognise them. Therefore, Christopher learnt these categories in a process of supervised classification learning. Finally, he was able to learn new tree categories without a teacher’s guidance by studying the trees and grouping together similar observations. This type of categorisation corresponds to an unsupervised classification process and is often associated with data clustering. Clustering is the key concept of this thesis. It will serve us as a glue to connect different ideas and questions from seemingly distinct fields of research. At the center of our interest are very general questions, such as: How do people acquire ideas about their environment? How do people structure the continuous stream of sensory information? How are these ideas and these structures represented in their brains? What are the general processing principles that underlie their acquisition? What do they tell us about cognition? Can these principles be implemented in a digital computer or are they fundamentally different from the principles of digital com-
putation? Can we derive corresponding mechanisms that are necessary for an autonomous artificial agent to work? How can the ideas be made useable for technical applications in the form of computer algorithms?

This thesis is inspired by such general questions. It would certainly be a hubris to claim that we can give definitive answers to some of these questions. The more moderate goal of this thesis is to justify and probe clustering as a mechanism, essential for cognition and desirable for ‘machine intelligence’. Our cogitation on the cognitive aspects of clustering will motivate us to think about potential neural mechanisms for concept acquisition and human reasoning, and about guidelines for the design of clustering algorithms. However, our theoretical approach cannot yield final answers for biological questions. The way we take is outlined in the following road map that will lead us through our own New World. This road map is supposed to give an overview of the whole thesis, but it does not necessarily reflect proportionally the importance of all parts. Each chapter is composed in a stand-alone fashion, completed with its own introduction. Hence readers who dislike repetition may start directly with chapter 1.

The Road Map for this Thesis

A thesis about clustering needs to answer the natural question What is clustering? before it can proceed to the actual problems. The answer to this question will evolve naturally during the course of chapter 1 from a broad definition to a narrow definition. At the beginning, we will use the term clustering in a very general sense: by clustering we will denote any grouping of observations or things. More precisely, if a set of \( N \) observations (or items, or general entities) is partitioned into groups (or clusters), we will understand this as a clustering process. In this liberal view, we do not bother about how the partition is achieved and how useful it is. It could be based on innate or learnt classification rules or due to some unsupervised learning process, or the partition could even be completely random. In technical terms, any process that can be associated with a map between a set of entities and a set of labels is understood as clustering. Items with the same label define a class or category. Generally, we assume that the labelling step is part of the clustering process. In contrast to the general term clustering, the term classification shall specifically denote a clustering process based on explicit classification rules. As the introductory example demonstrated, clustering is an important cognitive act. Intuitively, we may also associate such an act with computation. In section 1.1.1, we thus ask in what sense clustering shall be understood as computation and what the characteristics of this computation are. Our conclusion will be that clustering is even paradigmatic for the characterisation of computation if the latter is understood as a process in which superfluous information is destroyed. Such a process is called lossy data compression. However, if a clustering process shall be cognitively meaningful, its objective cannot be mere data compression. In sections 1.1.2 and 1.1.3, we will argue that the main goal of clustering is to find classes that allow for an optimal generalisation of features from one instance to all instances of the class. In order to fully understand the purpose to which clustering is dedicated, it must be acknowledged that classes serve as a basis for decisions that are relevant with regard to the environment. We will argue that clustering must thus be seen as a process with real-world input and real-world feedback.
At that point, our road will fork. One road will be concerned with techniques that largely neglect this real-world embedding of clustering. It will be taken up again in chapter 3 on clustering algorithms. Clustering algorithms define clustering as the task of partitioning \( N \) given data vectors into groups, such that vectors that belong to the same group are, in some sense, more similar than vectors from different groups. The other road will insist upon the point of real-world embedding. As a consequence, clustering will be understood in terms of mapping sensory information onto internal representations. In section 1.1.4, this view that clustering serves the acquisition of internal representations will be combined with the notion that clustering provides a paradigm for computation. In this context, one question naturally emerges: how does this notion of computation relate to the traditional notion of computation that is provided by the Turing machine? Our conclusion will be radical: if we really take the real-world embedding seriously, then a clustering process must be understood as a non-Turing computation, i.e., it is a process that is not one-to-one reproducible by a Turing machine. The argument is connected to the understanding that the acquisition of internal representations must be associated with continuous processes, akin to analogue-to-digital conversions. Consequently, we will conjecture that all basic neural processes are continuous and thus of non-Turing type. While computer scientists usually ask the question, how natural processes of information processing can be understood in terms of Turing machines or formal reasoning, our conjecture turns this question upside down. We now have to explain why humans are able to think according to formal rules for symbol manipulations if the underlying processes themselves are not describable in the language of such rules. Section 1.2 will be dedicated to this difficult problem. We will approach it by focussing on a toy world setting. That is, we will point out how a structurally simple world can be described by a formal language. This language emerges as a consequence of the symbolic representation of objects, which is enabled by clustering processes. The emergence of such a formal description can be viewed as a generic first step model to explain the emergence of formal reasoning in general.

In section 1.3, we will address the problem how these clustering processes leading to symbolic representations could be implemented at the level of neurons. We will distinguish between two types of approaches: the ones that correspond to the supervised learning of classifiers and the ones that must be understood as unsupervised learning processes yielding a self-organised grouping. In a broader sense, also supervised processes can be viewed under the paradigm of self-organisation, as learning in the brain is always an autonomous act. We therefore adopt the term ‘self-organised clustering’ for neural classification processes in general and discuss a couple of simple exemplary models. Once it is clear how objects are represented symbolically by means of neural networks, one may look for neural correlates of rules for formal reasoning. We will speculate about what such correlates could look like in section 1.4.

In chapter 2, we will concentrate again on models of non-symbolic information processing in neural networks. We will find them in the form of mechanisms of probabilistic reasoning or stochastic inference, which are naturally associated with continuous or some kind of fuzzy-natured processes. In section 2.1, we will propose to interpret neural activities in terms of message passing between distributed elements that locally perform simple computation. The example of belief propagation demonstrates how the message passing idea can be utilised for stochastic inference. We will show that under some natural constraints
the dynamics of continuous Hopfield networks is, in fact, equivalent to a time-continuous form of belief propagation. The mathematical structure of belief propagation will then be analysed in detail on the basis of Ising spin systems (section 2.2). These insights will turn out to be very useful for accelerating sequential superparamagnetic clustering SSC, a clustering algorithm that is discussed in chapter 3.

Chapter 3 will be concerned with design principles of clustering algorithms. We will distinguish between clustering with cost functions and clustering based on self-organisation. The distinction reflects the difference between top-down and bottom-up approaches for the design of clustering algorithms. We will first discuss a couple of popular top-down algorithms (section 3.1). We will then introduce SSC which is a powerful clustering algorithm that has been developed during the course of this thesis. It exhibits both top-down and bottom-up aspects (section 3.2). However, we will stress the self-organisation perspective, as this is the view that emphasises the principles of computation in nature. In this sense, SSC offers an alternative to the mainstream notion in machine learning that clustering is a top-down optimisation problem. Our road will end with a couple of clustering applications, demonstrating the power and benefit of SSC (section 3.3).

The following list gives a concise summary on the main goals and achievements of this thesis:

- We propose a measure that quantifies the computation performed by a clustering algorithm, closely following Stoop’s notion of computation (section 1.1.1).
- We postulate an abstract principle for the unsupervised learning of categories (section 1.1.2).
- From this, we infer that processes that provide the ‘fundamental categories’ should be considered as non-symbolic (non-Turing) computation (section 1.1.4). We further argue why processes in real neural networks which underlie decision making should also be understood as non-symbolic computation (section 1.2.1).
- We show why formal symbolic computation, akin to Turing computation, could emerge on the basis of fundamental categories that are associated with symbols (section 1.2.2 and section 1.3.2).
- We discuss several strategies for a neural implementation of category learning and we present a couple of simple exemplary models (section 1.3.2).
- We propose belief propagation as an alternative paradigm for the interpretation of neural computation, and demonstrate the equivalence of belief propagation and continuous Hopfield networks (section 2.1).
- As an alternative to pure cost function clustering, we stress the point of self-organisation for clustering algorithms and we develop the SSC algorithm (section 3.2).
- We provide a detailed mathematical treatment of the convergence properties of belief propagation and employ belief propagation for SSC (section 2.2 and section 3.2.2).
- We discuss diverse applications of SSC in different fields, such as neuroscience and combinatorial chemistry (section 3.3).
Chapter 1

Fundamentals of Clustering

1.1 Clustering and Computation

An essential aspect of clustering is the benefit of reduced costs for data storage: instead of keeping every single datum, only a prototype of each class has to be stored. As the original data cannot be retrieved from the prototypes, clustering is a lossy or non-invertible data compression. Throwing away superfluous information in a non-invertible manner is a key property of computation performed by natural cognitive systems such as human brains. In order to clarify this aspect, we develop a simple measure of computation which is essentially inspired by the Stoop measure of natural computation [93, 94] and allows us to quantify the computation performed by clustering algorithms. The basic result is that high computation goes along with high compression and vice versa. Obviously, high computation, in the sense of high compression, is not necessarily equivalent to good computation, in the sense of useful clustering results. In fact, compression to one class yields highest computation, but usually uninteresting clustering results. Therefore, we reason that the main goal of clustering is not high computation (or high data compression, respectively), but the prospect of optimal predictions and generalisations on the basis of the found categories/classes. For instance, if some elements of a class are known to be hostile, danger can possibly be avoided in future if all elements of this class are predicted to be hostile. The relevant question is whether this prediction is suitable or inappropriate. Hence, the quality of clustering, i.e., of the found classes, should be assessed from the perspective of the achieved predictive power in comparison to the achieved efficiency of data compression. Subsequently, we discuss the information bottleneck method which provides a bridge from our general considerations to basic principles of machine learning algorithms. The method starts from the idea that clustering is a trade-off between optimal compression and optimal prediction and finally arrives at the notion of clustering by cost functions. In opposition to this top-down view on clustering, we promote the idea of clustering by self-organisation. At the end of this section, we argue that the concept of clustering must not be reduced to common clustering algorithms, as the viewpoint of class-based prediction is in favour of the paradigm of embedded computation whose inputs are real-world signals. This will bring about a criticism of traditional, i.e., symbol-oriented, concepts of computation and will bridge over to section 1.2.
1.1.1 Measuring Computation Performed by Clustering Algorithms

The goal of clustering is either to identify groups in a set of entities or to group the entities in some way. In the case of humans, we may associate such a process with cognition. In the case of computers, we can argue more technically that clustering is a process that is associated with a mapping $f_c$ between a set of $N$ different data items and a set of $k$ data labels. For now, we consider any algorithm that performs such a mapping as a clustering algorithm, irrespective of the mechanism behind it. As such an algorithm calculates a map, we may claim that it performs computation. What exactly is the nature of this computation? How can it be characterised or measured? It is not our purpose to characterise clustering in terms of complexity classes [88]. We consider this characterisation of computation as rather irrelevant for the cognitive aspect of clustering. The problem is, how to characterise clustering as a process of natural computation.

In [93], R. Stoop argued in favour of the notion that dynamical systems provide a powerful and universal paradigm for natural computation, i.e., a computational process must be interpreted in terms of a dynamical system. He pointed out that the computation performed by a dynamical system should be understood as the reduction of the difficulty of prediction in the statistical sense. Here, the relevant question is: how difficult is it in average to predict the behaviour of the system? According to Stoop’s approach, computation should facilitate the prediction, or reduce the complexity, if the latter is understood as the obstruction against predictability. Consequently, an appropriate measure of complexity was elaborated that quantifies ‘obstruction against predictability’ in a natural manner [95]. In order to quantify the computation performed by a clustering map $f_c$, we conceptually follow the same ideas.

We directly compare the input and the output of the clustering process, though. In terms of dynamical systems, this means that the system that implements the clustering is treated as a black box. We then define a measure of computation $CO$ on the basis of the prediction complexity $\Delta C$, resolved by the clustering process, i.e.,

$$CO := \Delta C \in [0, 1]. \quad (1.1)$$

$\Delta C$ accounts for the notion that – intuitively – the data set appears structured and thus somehow less complex if clusters can be identified. To determine $\Delta C$, we choose a novel approach that can be viewed as a shortcut of the Stoop complexity [95] in the sense of the black box idea. The advantage of our approach is a straightforward implementation and a simple interpretation, but its scope is restricted to clustering processes characterised by a map $f_c$. We determine $\Delta C$ as the difference of the difficulty of predicting a new input before and after clustering, where the difficulty of making predictions is quantified by the entropy. Reduction of complexity then corresponds to the entropy difference before and after clustering (pre- and post-clustering condition).

In the pre-clustering condition, we are given a set $s$ of data represented as vectors in an Euclidian feature space, i.e., $s = \{x_1, x_2, ..., x_N\}, x_i \in \mathbb{R}^m$. We can assume a natural quantisation of the feature space into a discrete set of cells $\{c_1, c_2, ..., c_k\}$ as, practically, such a partition is given by the finite measurement precision achievable by an (external) observer (Fig. 1.1). The entropy for the pre-clustering condition $H_{pre}$ is defined by

$$H_{pre} := -k \sum_i p(c_i) \log_b(p(c_i)), \quad (1.2)$$
1.1. CLUSTERING AND COMPUTATION

Figure 1.1: a) In the pre-clustering condition, a natural partition is given by the finite precision of measurements. b) In the post-clustering condition, fundamental cells are fused to cluster-cells. Each point represents a data vector.

where we sum over all cells \( c_i \). Here, \( p(c_i) = m_i/N \) is the relative occupation of cell \( c_i \), where \( m_i \) is the number of vectors in cell \( c_i \). The constant \( k \) and the choice of the logarithm base \( b \) will turn out to be arbitrary thanks to a suitable normalisation. Hence we omit \( b \) in the following. The entropy is a measure for how much information is resolved on average by observing a single vector \( x_i \), given the whole set \( s \). As an example, in many clustering applications, all individual vectors \( x_i \) will be different within the limits of measurement precision. Consequently, each cell \( c_i \) has at most one occupant. The entropy \( H_{pre} \) thus takes on its maximal possible value

\[
H_{pre}^{max} = k \log(N).
\]  

In the other extremal case, all vectors \( x_i \) coincide. Hence only one cell \( c_1 \) is occupied and we have

\[
H_{pre}^{min} = 0.
\]  

From this example, we observe how the entropy serves as a measure of the difficulty of prediction. In the first case, the prediction of an additional data vector \( x_i \) is maximally difficult as it could be in any cell. In the second case, the prediction is trivial as we expect that \( x_i \) again is in the one cell.

Clustering is the fusion of cells into larger units, in the following termed cluster cells \( cc_i \). In the post-clustering condition, the feature space is segmented into a discrete set of distinct cluster cells \( \{ cc_1, cc_2, ... \} \), and each cell is a union of cells \( c_i \) (Fig. 1.1). The post-clustering entropy is

\[
H_{post} := -k \sum_i p(cc_i) \log(p(cc_i)),
\]  

where we now sum over all cluster cells \( cc_i \) and \( p(cc_i) = m_{cc_i}/N \) is the relative occupation of a cluster cell \( cc_i \). The fusion of two cells \( c_i \) and \( c_j \) with occupation numbers \( m_i \) and \( m_j \) will always lead to a smaller entropy as

\[
\frac{m_i + m_j}{N} \log \left( \frac{m_i + m_j}{N} \right) \geq \frac{m_i}{N} \log \left( \frac{m_i}{N} \right) + \frac{m_j}{N} \log \left( \frac{m_j}{N} \right).
\]

Hence \( H_{post} < H_{pre} \), which means that the difficulty of prediction is smaller in the post-clustering condition. This is in accordance with the intuitive feeling that the prediction is
facilitated if we can recognise clusters in the data (as it is likely that an additional vector $x_i$ is in one of the cluster regions). The reduction of prediction complexity $\Delta C$ is then defined by

$$\Delta C := \frac{H_{pre} - H_{post}}{H_{pre}}. \quad (1.6)$$

We have to state that this measure does not make any statement about the quality of the achieved clustering result. It just quantifies the computation in terms of compression. If $N$ data are grouped into two classes, then the computation performed is always the same, no matter whether the classification is reasonable. Maximal computation is performed if $\Delta C = 1$, i.e., if $H_{post} = 0$ (we exclude $H_{pre} = 0$), which means that all vectors $x_i$ fall into one cluster cell $cc_1$. From this, we infer that computation typically increases with a decreasing number of clusters. If the clustering process yields just one cluster, the computation is always maximal ($\Delta C = 1$) as the prediction in the post-clustering condition is absolutely trivial, regardless of the difficulty of prediction in the pre-clustering condition. This result is in complete contrast to the result that would be obtained if the mutual information [26] was used for measuring computation. In that case, compression to one cluster would correspond to zero computation. However, the result is in accordance with the measure of computation originally proposed in [93] for dynamical systems,

$$CO_s = \frac{1}{C(1, 0) + 1}, \quad (1.7)$$

where $C(1, 0)$ denotes the Stoop complexity [55]. It measures the complexity of a system as the difficulty for an observer of making predictions (choosing an appropriate observable). A clustering process that yields only one cluster can in principle be implemented as a constant map $f = const$ with complexity 0. Hence the system performs maximal computation.\(^1\)

### 1.1.2 Principle of Optimised Class-Based Predictions

In section 1.1.1, any process that defines a map from a set of $N$ different data items to a set of $k$ labels is considered a clustering process. In the literature, the term ‘clustering’ is usually used more specifically for unsupervised category learning. This kind of learning refers to the observation that humans are able to learn a new category of completely novel objects, e.g., a new animal species, without an external supervisor. The problem of unsupervised category learning typically looks as follows. Given, for instance, a set of $N$ animal input items (e.g., pictures) $A = \{a_1, ..., a_N\}$, then different categories $C = \{c_1, ..., c_k\}$ of animals shall be learnt by means of a clustering process whose task is to establish a (possibly fuzzy-natured) mapping $f_c : A \rightarrow C$. As a difficulty, the number of categories $k$ is usually unknown. Naturally, the following questions arise: What are the guidelines along which

\(^1\)However, also the computation for a piecewise constant function (that implements several clusters) is maximal, which identifies a slight conceptual difference to our approach. The Stoop complexity emphasises the difficulty of prediction of (bundles of) phase space orbits, where the exact task is to predict a chosen observable along the orbits. Thus the Stoop measure emphasises the dynamic aspects of a computational process, and it is also applicable if the system does not perform a simple static input-output relation $f_c$. This in contrast to the black box idea of our approach that abstracts from the system itself, but emphasises the result, given by $f_c$. In other words, computation is basically identified with a final result, corresponding to a fixed point of the underlying system.
1.1. CLUSTERING AND COMPUTATION

Figure 1.2: Optimal prediction performance (solid line) is achieved as a trade-off between the generalisation power (dashed line) and the risk of wrong predictions (lower dotted line).

such a clustering shall be achieved? What is the abstract principle upon which the map \( f_c \) should be learnt?

Obviously, the maximisation of computation in terms of our measure (1.6) cannot be the primary goal of a clustering process, because it would be tantamount to throwing away all information about the original data distribution by establishing just one class that can be characterised by a single prototype. Instead, the primary goal of clustering is to enable class-based predictions that stand the ‘test of reality’. In this understanding, a class-based prediction is the generalisation of a certain property from a single instance to the whole class. For instance, if a certain striped black-yellow insect has just stung me, a natural prediction is that any such insect can potentially sting me. Therefore, my further contacts with other instances from the same black-yellow striped insect class should be more cautious. Obviously such a prediction can be more valuable or less valuable. Large classes allow for predictions of high generality, and they additionally provide a sparse coding because many instances can be replaced by one prototype. As a drawback, the risk of wrong predictions is increased. Practically, there must be a trade-off between the generality of a class and the risk of a misclassification (see Fig. 1.2).

We postulate that organisms (‘natural agents’) perform unsupervised category learning by optimising clustering processes with respect to optimal class-based predictions relevant for the survival in the environment and with respect to an as sparse as possible class representation. We refer to this postulate as the principle of optimal prediction and sparse representation.\(^2\)

Generally, we distinguish between embedded prediction optimisation and data-intrinsic prediction optimisation. Embedded optimisation refers to the fact that organisms (or any kind of autonomous agents) exchange with an environment and steadily incorporate external information. Clustering is a never-ending procedure in real-time, as such an agent is constantly and serially sampling input instances. Imagine that the agent characterises these

\(^2\)Our position is in contrast to the notion that learning can be based on purely data-intrinsic principles, such as the minimum description length [5].
items by a feature vector \(a_i\), and it stores the feature vectors in a list. As soon as the list is large enough, it is subdivided into feature vector categories by clustering. Each additional instance is then classified according to the established categories. This allows for predictions about this new instance, which can be evaluated by testing their correctness or behavioural relevance. If a prediction is proven wrong, the clustering solution must be revised. For instance, if an object is judged as friendly, but it then stings the agent, the class assignments may be unsound. Optimisation of the clustering is consequently performed on the basis of prediction validation, enabled by an agent-environment feed-back loop. In the formalism used above, clustering establishes a (time-dependent) map \(f_c(t)\) between an increasing input set \(A = \{a_1,\ldots,a_{N(t)}\}\) and a possibly variable output set \(C = \{c_1,\ldots,c_k\}\). For the clustering, old input items are - ideally - available from storage and new input is frequently added, influencing the mapping \(f_c(t)\). Old assignments might have to be modified or even new categories must be introduced. The New AI community [72] has promoted the idea of embodiment as a natural consequence of embedding. Embodied cognitive science might thus be a good general framework for understanding the basis and importance of clustering by embedded prediction optimisation. In particular, the idea of sensory-motor coordination emphasises the importance of the mutual interaction between an agent and its environment [73].

Common clustering algorithms used for data analysis depart from the problem of finding a useful clustering map from a fixed finite input set \(A\) to a label set \(C\). They restrict their task to identifying classes in \(A\) according to an off-line procedure. Such algorithms usually ignore the feedback loop of the environmental embedding. Class-based prediction optimisation is thus reduced to the task of identifying the ‘most natural clusters’ in the data set, which is a purely data-intrinsic optimisation. In order to perform this optimisation, most clustering algorithms rely on the specification of a cost function. This cost function defines the ‘naturalness’ of a found clustering solution and implicitly contains expectations about the environment in which the algorithm is used. The guiding assumption for the construction of cost functions can be termed similar-property principle. It expresses the experience that structurally similar objects tend to behave similarly in many situations. The cost function thus has to reflect the average structural similarity within groups, achieved for a certain classification. For this, we need to specify a similarity measure.

A short survey of clustering algorithms based on data-intrinsic optimisation follows in chapter 3. At this point, we want to outline how the usage of cost functions can be justified from the more abstract principle of optimal prediction and sparse representation. The key ideas are known as the information bottleneck method IBM [85, 98]. In contrast to this top-down view on clustering, we will emphasise a complementary bottom-up view that understands clustering as a self-organisation process based on local interaction or learning. We will encounter the idea of self-organised clustering in section 1.3.1, where we discuss simple models for neural clustering processes.
1.1.3 Information Bottleneck Method

The information bottleneck method starts from the assumption that a good clustering solution is a trade-off between compressing the data representation and preserving the relevant information for the prediction of another variable. This notion employs a probabilistic or fuzzy-clustering framework. We denote the original signal variable by $X$ and the compressed representation by $\tilde{X}$. Each $x \in X$ is then associated with a cluster representative $\tilde{x} \in \tilde{X}$ with probability $p(\tilde{x}|x)$. The compression quality is measured in terms of the mutual information $I(X;\tilde{X})$. The smaller $I(X;\tilde{X})$, the better is the compression. The relevance variable $Y$ that shall be predicted must be independent of the original signal $X$. Optimising the prediction is equivalent to preserving the relevant information about $Y$, which again is equivalent to maximising the mutual information $I(\tilde{X};Y)$. In [98], $I(\tilde{X};Y)$ is called the meaningful information. While maximising $I(\tilde{X};Y)$, the compression shall be fixed. Equivalently, we may maximise the compression for a fixed amount of meaningful information. The optimal clustering is thus found by minimising the functional

$$L(p(\tilde{x}|x)) = I(X;\tilde{X}) - \beta I(\tilde{X};Y).$$

(1.8)

$\beta$ is the Lagrange multiplier corresponding to the constraint of a fixed amount of meaningful information. $\beta$ plays the role of a resolution parameter that controls the trade-off between optimal compression and prediction. The solution to this minimisation problem can be given as [98],

$$p(\tilde{x}|x) = \frac{p(\tilde{x})}{Z(x,\beta)} \exp(-\beta D_{KL}[p(y|x)|p(y|\tilde{x})]),$$

(1.9)

where

$$Z(x,\beta) := \sum_{\tilde{x}} p(\tilde{x}) \exp(-\beta D_{KL}[p(y|x)|p(y|\tilde{x})])$$

and

$$p(y|\tilde{x}) = \frac{1}{p(\tilde{x})} \sum_x p(y|x)p(\tilde{x}|x)p(x).$$

$D_{KL}$ denotes the Kullback-Leibler divergence [26]

$$D_{KL}[p(y|x)|p(y|\tilde{x})] = \sum_y p(y|x) \log \left( \frac{p(y|x)}{p(y|\tilde{x})} \right).$$

(1.10)

Equation (1.9) defines an implicit solution as the $\tilde{x}$ are still unknown. They can be found in an iterative manner. For details, we refer to [98]. Here, we want to emphasise two aspects of importance for us:

(I) There is a large group of clustering algorithms (e.g., [26, 29, 75, 108]) that are based on the concept of a cost function, or in the language of information theory, an average distortion function,

$$<d(x,\tilde{x})>_{p(x,\tilde{x})} = \sum_x \sum_{\tilde{x}} p(x,\tilde{x})d(x,\tilde{x}).$$

(1.11)

For $d(x,\tilde{x})$, the Euclidian measure $d(x,\tilde{x}) = (x - \tilde{x})^2$ is a common choice. The distortion expresses some average cost that we have to pay for a certain clustering assignment. If

Please note that we adopt the notation from [98], which differs from our notation used in sections 1.1.1 and 1.1.2.
all points are assigned to one cluster centre, the distortion costs are high, but the data compression is optimal. If, in turn, we minimise the distortion, we achieve a worse data compression. Consequently, we have to look for a trade-off between data compression and average distortion. In practice, we want to maximise the data compression for a given distortion, which amounts to minimising the functional

$$F_{I}[p(\tilde{x}|x)] = I(X; \tilde{X}) + \beta <d(x, \tilde{x})>_p(x, \tilde{x}).$$

(1.12)

The solution for this problem is [98]

$$p(\tilde{x}|x) = \frac{p(\tilde{x})}{Z(x, \beta)} \exp(-\beta d(x, \tilde{x})).$$

(1.13)

A comparison with (1.9) suggests to interpret $D_{KL}[p(y|x)|p(y|\tilde{x})]$ as $d(x, \tilde{x})$. Hence, IBM justifies the usage of cost functions from an abstract information theoretical point and suggests the Kullback-Leibler divergence as the natural distortion measure.

(II) We can derive equation (1.13) from another perspective that gives us an immediate interpretation of the clustering problem in the language of physics. We start from the intuition that the average distortion should be minimised. Obviously, for a given data distribution a small distortion is achieved if each data point $x$ is fully assigned to the one cluster representative $\tilde{x}$ which is next to $x$ in terms of $d(x, \tilde{x})$. However, in order to optimise the data compression, we have to allow for some randomness, that is, $0 < p(\tilde{x}|x) < 1$ for all $\tilde{x}$. The best case in terms of data compression is the situation of complete randomness, where all cluster centers, as well as the probabilistic assignments are equal. This is equivalent to assigning all points to one center only. The randomness can be measured in terms of the conditional entropy $H = H(\tilde{x}|x)$. This quantity shall be fixed. Following the same trade-off principle as before, we want to minimise the functional [26, 75]

$$F = D - TH,$$

(1.14)

where $T$ is a Lagrange multiplier and $D = <d(x, \tilde{x})>_p(x, \tilde{x})$, $F$ is known from physics as the free energy functional. It is straightforward to interpret all occurring quantities in terms of the statistical physics of a canonical ensemble. The average distortion $D$ corresponds to the average system energy $E$. $H$ is the system entropy and $T = 1/\beta$ is the system temperature. For a set of fixed $\tilde{x}$, $F$ is minimised by the Boltzmann distribution (also called Gibbs distribution) [75]

$$p(\tilde{x}|x) = \frac{1}{Z_x(T)} \exp(-E_x(\tilde{x})/T),$$

(1.15)

where $E_x(\tilde{x}) = d(x, \tilde{x})$ is interpreted as energy and $Z_x(T) := \sum_{\tilde{x}} \exp(-E_x(\tilde{x})/T)$ is the partition function. In practice, the parameters $\tilde{x}$ are still to be determined in order to obtain the actual clustering solution (solution via deterministic annealing, see section 3.1.1). Equation (1.15) illustrates the role of the temperature $T$ as the trade-off control parameter. For small $T$, the distortion term $D$ dominates the functional $F$ and minimising $F$ results in a hard (nonrandom) clustering solution. For large values of $T$, the entropy term $H$ dominates and $p(\tilde{x}|x)$ becomes completely even distributed. As a consequence, all $\tilde{x}$ coincide, which is equivalent to optimal compression with one cluster only.
1.1.4 Fundamental Clustering as Non-Turing Computation

The principle of prediction optimisation (section 1.1.2) is based on the idea that the system that performs the clustering is embedded in a real-world environment. In this spirit, clustering or categorisation is often understood in terms of mappings of sensory input onto internal representations of objects and object classes [73]. Such mappings are vital computational operations to be performed by an autonomously acting agent. We will argue that these operations allow for an understanding of natural computation, which does not necessarily connect to the traditional notion of Turing-like computation. We are thus first referred to the basic question: what do we actually mean by the term ‘computation’?

Computation is a broad term, usually used to denote any kind of information processing. As a consequence of this broadness, there is no agreement on a universal notion of computation. In a general understanding, any physical process that transforms variables can be thought of as a computation, as long as it can be mapped onto mathematical operations that implement some useful functions [40]. In contrast, according to the notion developed in [93], the main characteristic of computation is the reduction of prediction complexity, which can be associated with the destruction of information. Besides these more nature-oriented notions, there is the standard notion provided by computer science. It is exemplified by the concept of Turing machines and by akin concepts (e.g., [88]). Turing machines, although often believed to catch all aspects of computation (strong version of Church-Turing Thesis [88]), are subject to grave assumptions about the nature of computation. In the very core, computation is understood as the manipulation of symbols. Symbols are entities that are virtually dissociated from the real world. This is reflected by the idea that computation is a pure matter of software. Therefore, any physical system that is able to represent and manipulate the symbols required for a specific computation can carry out this computation. In such a process, the symbols and symbol manipulations are also exempt from any meaning, a fact that is sometimes described as abstract and syntactic formality [47]. The manipulations need to be reliable, i.e., predictable and reproducible. Hence, the manipulations must follow clear rules, defining an algorithm. This notion has severe consequences:

(I) Strictly speaking, if a physical process is described on the basis of real numbers, it cannot be understood as computation as such a process cannot be represented in a one-to-one manner on the basis of a finite set of symbols (computers or Turing machines operate on a subset of the rational numbers, they cannot represent a continuous process)
(II) Noise, ubiquitous in biological systems, is not appreciated. It merely plays the role of a disturber.
(III) A computation is a process that is independent of any real-world embodiment. Neither physical matter nor physical time are relevant aspects of computation.
(IV) Computation is understood as a process with a well defined, i.e., in particular reproducible, input and output.

We argue that this notion of computation is incompatible with the core idea of clustering sketched in section 1.1.2, where we emphasised the environmental embedding of autonomous agents. It is clear that this statement is unsustainable if clustering is equated with clustering algorithms. An algorithmic realisation of a clustering process is inevitably built upon the assumption that clustering is a computational process that is consistent with the notion
CHAPTER 1. FUNDAMENTALS OF CLUSTERING

(a) Clustering of feature vectors

(b) Clustering of trajectories

Figure 1.3: Different objects are represented as clearly distinguishable macrostates (cluster regions) in a sensory space or any abstract feature space. Macrostates are recognised by means of a fundamental clustering or fundamental categorisation process.

Our criticism, however, attacks the notion of computation at a more fundamental level. Consider a system (a person, an animal or an artificial agent) that is acting in a real-world environment from which it is receiving input. A general goal of the system is to grasp regularities or invariants of the environment, to make it recognizable and accessible for predictions – an environment without any regularities would not allow for any predictions. As an example, some agent can recognise a certain house because all sensorial encounters with this house exhibit some invariant features. Imagine, for simplicity, that the agent always faces the house from the same side such that, for instance, the house’s shape and colour are reliable invariant features. In an abstract description, these features are represented as points in a real-valued feature space. In reality, the sensorial data, from which shape and colour are inferred, always vary a bit for two different situations (two different sensorial encounters) and so do the inferred values for the features. Hence each sensorial encounter leads to just one point in this feature space. Each point corresponds to a unique event. As the variations of the features are small, the points corresponding to different encounters of the same object make up a point cloud or cluster (see Fig. 1.3 a)).

Definition 1.1 The region of the feature space covered by the cluster made up of points corresponding to different sensorial encounters of the same object is termed a macrostate. In contrast, a single point is called a microstate.

It makes sense to identify the macrostate with the object representation as it stands for the totality of possible sensorial data from the same object. In a more complex setting, a single sensory encounter is understood as a continuous data stream, represented by a whole trajectory in some abstract state space rather than by a single point. In this case, a macrostate corresponds to a bundle of trajectories (see Fig. 1.3 b)). In order to access and use the information stored about an object, the agent must somehow know or recognise the macrostate that represents the object.
Definition 1.2  By a fundamental clustering process we denote any process that allows to identify or to characterise different macrostates whose elements correspond to unique input events.

We could imagine that a fundamental clustering process either labels the cluster regions in the feature space (by some indicators) or establishes a prototype for each region. We can then identify a macrostate with its label or prototype. A fundamental clustering process is not exactly a clustering algorithm. The difference is expressed in the following remark.

Remark: A clustering algorithm establishes a map between a finite set of distinguishable items and a set of labels or item prototypes. In contrast, a fundamental clustering process provides or establishes a map between whole regions of a continuous space and a set of labels or prototypes. The regions must be identified on the basis of microstates that have been experienced in a temporal order.

The subsumption of microstates under a macrostate prototype destroys the original information about the single sensory events, since from the prototype we cannot infer the original microstates. Only the general scheme of an event survives in the form of the prototype. In contrast, microstates cannot be described individually as they correspond to unique, non-identifiable, and non-recurring sensory events. In nature, no sensory situation is likely to reappear in a completely identical manner and each single microstate has zero probability. Consequently, the original input is, strictly speaking, not well-defined as it is as a matter of principle unreproducible. The identification of the input of a fundamental clustering process with a set of symbols $\mathcal{A}$, as it was done in section 1.1.2 for a clustering algorithm, is therefore, strictly speaking, inadmissible. In short, a fundamental clustering process violates point (IV) from the list above. A fundamental clustering process can thus not be considered a symbolic computation in the sense of a computer algorithm. To put it somewhat differently: we argue that real-world input (giving rise to a single microstate) is non-symbolic due to its non-reproducibility. It cannot be named or labelled because it only exists at the very moment of perception. By symbols, we can only label the invariant or reproducible aspects of it. These aspects are expressed in the associated macrostate. Macrostates, or their characterising prototypes respectively, are potentially recurring and hence reproducible. They may thus provide the ‘fundamental symbols’: as macrostates are identified in fundamental clustering processes, such processes enable symbol systems [72] that are a premise for symbol-based computation and cognition (symbolic reasoning).

Definition 1.3  By symbolic reasoning we denote any process that can be described as a rule-based manipulation of symbols. The symbols are supposed to be grounded [72] by denoting some objects.

Taken for granted that a fundamental clustering process is not an act of symbolic reasoning, it is questionable whether we should associate it with any type of computation. If the notion of computation is restricted to the concept of Turing-like computation, a fundamental clustering process lies beyond the realm of computation. In the Turing notion, computation...
and symbolic computation are principally equivalent. However, following Stoop’s notion of computation (see section 1.1.1), a fundamental clustering process must clearly be judged as a computational process: the prediction of a macrostate is ultimately easier than the prediction of a microstate, as a microstate has zero probability. Therefore, the process reduces the difficulty of prediction, which is the characterising aspect of computation. Accepting this characterisation of computation, non-Turing-like processes such as fundamental clustering processes clearly perform computation.

As in our understanding, real-world input per se is non-symbolic, the set of real numbers rather than finite symbol sets (or countable sets such as the rational numbers) constitutes the mathematical structure that can acknowledge the character of real-world input. There is no one-to-one mapping between the set of uncountably many real numbers and a always countable set of symbols. Hence, the totality of real numbers constitutes an essentially non-symbolic structure. It is thus coherent that computational models on the set of real numbers capture essential aspects of computation that cannot be performed by Turing machines [82, 84]. By means of measurements, however, any real world input can be symbolised – measurements are always subject to finite precision. Take the example of measuring the length of an object. In practice, we can only distinguish between a finite set of possible measurement values (coarse-graining [13]). On the one hand, these values can be interpreted as symbols. On the other hand, we may argue that each value constitutes a macrostate as it comprises many indistinguishable microstates (object lengths). According to this, a measurement process is a fundamental clustering process and is thus a non-Turing computation. Imagine again an embedded artificial agent whose peripheral sensors are analogue, but whose internal information processing shall be digital. This requires a peripheral analogue-to-digital (A/D) conversion, which is a process akin to a measurement process. It is therefore a non-symbolic (non-Turing) computation. A/D converters can generally be considered as cradles of symbols as they translate analogue, i.e., identically un reproducible real-world signals into reproducible symbolic signals. Similar A/D conversions could take place in biological systems, ultimately in the brain, providing the fundamental symbols for further symbolic computation. Yet, despite the influential paradigm of symbolic computation, the nature of neural computation may not, or may only be partly symbolic, i.e., it may not be based on an algorithmic manipulation of symbols. The goal of the next section is to shed light on this point, by going through the ideas developed in this section more carefully on the basis of a toy model.

### 1.2 Cognitive Aspects of Clustering

According to the ideas sketched in the last section, fundamental clustering processes underlie the acquisition of symbols and so provide the basis for symbolic reasoning. Symbolic reasoning is often equated with cognitive skills, the summit of which is sometimes identified with formal logic. The goal of this section is to gain an understanding of the role of clustering for cognition by connecting terms that are associated with cognitive skills, such as concept learning, i.e., the acquisition of mental representations of categories [77], planning, 

---

5The assumption that real-world input is, on principle, un reproducible is probably not a sufficient condition to prove the real number or continuous-type character of the world. However, real numbers offer a convenient description tool.
and formal thinking, to fundamental clustering processes.

Cognition is attributed to a whole bunch of high-level or mental functions that are concerned with attention, memory, learning, perception and planning [37]. Here, we focus on certain aspects by considering a virtual toy world with a simple structure. It consists of a bounded arena that contains a couple of simple objects (see Fig. 1.4). A robot is the only resident in this world. It is equipped with simple sensors and three elementary cognitive functions: it can (1) discriminate and recognise single objects, it can (2) find categories of similar objects and it can (3) plan routes. For the recognition, discrimination and classification of objects, the robot needs to acquire a concept (internal representation) of each single object and of object classes. Accordingly, for route planning it needs to acquire an internal representation of a route. We will argue that concepts provide a symbolic description or encoding of the environment, allowing for symbolic reasoning. Symbolic reasoning amounts to finding relations between different concepts, which translate into relations between the corresponding symbols. In our toy world setting, a route from object A to object B can be understood – in an abstract topological view – as a relation between A and B. Planning a route is hence an act of symbolic reasoning for which the rules are given by the topological structure of the space. We will find that this act can be formalised by means of purely rule-based symbol manipulations (formal reasoning).

**Definition 1.4** We denote by formal reasoning an act of symbolic reasoning that is purely syntactic.

In contrast, symbolic reasoning assumes that it is clear and known what concepts the symbols refer to. According to the sketch above, we address the following points on the basis of the toy world setting:

- (I) We explain our notion of a concept and discuss the role of clustering for unsupervised concept learning.
- (IIa) We explain our notion of symbolic representations.
- (IIb) We explain how symbolic reasoning can be characterised in the given toy world setting.
- (IIc) We explain our notion of non-symbolic processes and speculate about its relevance for neural computation.
- (III) We finally explain how symbolic reasoning can emerge as a consequence of the structure of the toy world.

At the end of this tour, we will be able to explain why the robot can acquire a scheme for formal reasoning on the basis of non-symbolic processes. This problem of bridging the gap between symbol processing systems and systems that do not perform formal operations on symbols has primarily been addressed by Connectionism (e.g., [7, 87, 99]). For Connectionists, one of the fundamental questions is: how can a process of symbolic reasoning be achieved with the neural network architecture? By neural network architecture, Connectionists usually understand real-valued feedforward networks that are trained in a supervised
process [50]. Most Connectionists implicitly assume that processes of natural computation are essentially non-symbolic and thus symbolic computation must be reducible to terms of these more fundamental processes. In the following, we will develop our own terminology for the problem, based on the idea of fundamental clustering processes. It is convenient to think of such clustering in terms of identifying macrostates in a continuous feature space. This is the picture sketched in section 1.1.4. In section 1.3, we will connect the ideas to the neural level by thinking about possible neural implementations. We use different levels of description that have to be distinguished, for which the scheme in Fig. 1.5 further below in this section may be helpful.

1.2.1 Clustering Serves Concept Acquisition: A Toy Example

In our virtual world setting, we imagine a robot that moves in a two-dimensional noisy environment which contains different objects, such as rectangles and circles or ellipses. No additional features shall be attributed to these objects. The robot can scan these objects by surrounding them (see Fig. 1.4). For this purpose, the robot is equipped with a couple of whisker-like analogue sensors. By scanning an object, the robot records a sensory fingerprint given by the activation pattern of the sensors. A sensory fingerprint corresponds to a nonrecurring real-world event and hence is associated with a microstate in a feature space (see definition 1.1). In the simplest case, the features are just provided by the sensor activity. In this case, a sensory fingerprint is associated with a micro-trajectory in a sensory space. Alternatively, the original signal is transformed by a suitable operation (e.g., a Fourier transform) that yields characteristic features. We do not impose any particular task on the robot, except exploring the world by scanning the objects. For the discovery tour, the robot is supposed to have implemented the three basic cognitive functions mentioned above.
1.2. COGNITIVE ASPECTS OF CLUSTERING

(I) **The role of clustering for unsupervised concept learning:** Concepts are mental representations of categories [77]. In the case of the robot, we equate a mental representation with some physical configuration or process within its ‘brain’. This point of view is adopted as a working hypothesis for the case of animal or human brains; we do not want to engage in the discourse on the mind-body problem [86] and its philosophical consequences concerning free will, consciousness etc. More precisely, we postulate that the mental representation of categories conforms with a macrostate description, as sketched in section 1.1.4. In other words, some sets of internal states of the ‘robot’s brain’ are describable as macrostates in a feature space. These sets correspond to categories at the semantic level. The elements of these categories refer to microstates in the feature space and to sensory fingerprints in the physical space.\(^6\) The robot is obviously confronted with different kinds of categories. Firstly, a single object can be described as a category whose elements correspond to all sensory fingerprints that belong to this object (low-level). Secondly, the groups of circles and rectangles or the groups of small and large objects form categories of single objects that are similar (high-level). The function of a fundamental clustering process is to make the fundamental categories accessible (e.g., by determining a prototype).

**Definition 1.5** If the characterisation of a concept can be described as a fundamental clustering process in a feature space, then we shall speak about a fundamental concept. By fundamental category, we denote the corresponding semantic entity (the distinguishable thing we mean by, e.g., ‘this cup’).

Hence, the elements of a fundamental category correspond to microstates in the feature space, which in turn correspond to unique non-recurring events (sensory fingerprints) in the physical space (Fig. 1.5). Not every category is necessarily a fundamental category. Moreover, it is not clear a priori what the fundamental level of categories is. Either the ‘high-level’ categories can be obtained by clustering the ‘low-level’ categories, or the ‘low-level’ categories can be obtained by segmenting the ‘high-level’ categories. In this picture, clustering and segmentation show up as inverse operations. This also means that the entities that we recognise as the ‘natural objects’ (e.g., a certain rectangle) do not have to be the fundamental categories. It is conceivable that a category that corresponds to a natural object, itself is composed of more fundamental subcategories. To illustrate this thought, let us again assume that every sensory fingerprint can be represented as a point in a real-valued feature space. If we apply a coarse-graining, i.e., each point is described by a vector of finite precision, we impose fundamental categories at an early stage, since infinitely many points are described by the same vector. A natural object could then be obtained by constructing a ‘higher-level’ category, composed of many of these subcategories. This could be achieved by any suitable clustering algorithm and is not a fundamental clustering process.

In summary, concepts correspond to sets of internal states of the robot’s brain, which stand for categories. We have to distinguish a concept from a category as we discriminate between the representing state (physical level) and the represented content (semantic level). Additionally, at the intermediate level of description, we have the characterisation of the physical

---

\(^6\)A concept thus is a set of brain states that correspond to a feature space region. In this respect, our notion of a concept differs from the notion in [77]. The authors would probably argue that the brain state which represents the prototype, is the concept. In our notion, this state must be interpreted as the representative of the concept (see Fig. 1.5). We will call it ‘indicator token’ or ‘token’.
state in terms of a feature or state space. As we assume an unambiguous relationship between the representing state and the represented content, we will now – for simplicity– use the terms ‘concept’ and ‘category’ interchangeably.

Besides the concepts that encode objects such as rectangles, there is another class of concepts that are of importance for the robot: operational concepts. We define an operational concept as an internal representation of a possible action. For our robot, the set of possible actions is confined to movements, such as ‘from A to B’. Here, A and B are symbols denoting different fundamental categories, which could be single rectangles or circles. Operational concepts can be considered higher level concepts as they are based on object concepts. Alternatively, we interpret them as relations between object concepts.

(IIa) Symbolic representations: A fundamental category can be labelled by a symbol. How can we achieve the labelling? The robot could be able to represent a specific object by a prototype vector in the feature space; this assumption is called ‘prototype theory’ [3]. In this case, a fundamental clustering process assigns each microstate to its prototype. The physical entities that realise the prototype are called tokens [13]. Several tokens can stand for the same prototype. At the physical level, a sensory fingerprint is mapped onto one of these tokens. Alternatively, a category might be indicated by a class label. A sensory fingerprint is then mapped onto a physical label or indicator state. Unless in the prototype case, any information about the characteristics of the original fingerprint is lost. By default, we will stick to the prototype idea. If the prototypes are clearly distinguishable, which also requires that the corresponding tokens are clearly distinct, we can define a map between the prototypes and symbols from a given alphabet. This map also translates into a map between the tokens and the symbols. In this way, a symbol is said to represent a category via the identification of the category with the prototype. Conversely, we can also say that a token realises a symbol (see Fig. 1.5). It may not be clear-cut when two different prototypes can be considered as clearly distinguishable. However, it is clear that if two prototypes are very similar, they should rather be judged as two instances of the same ideal prototype. This might even be the normal case: as all physical media are subject to noise, the physical correlates of prototypes must correspond to clearly distinguishable clusters of equivalent microstates or micro-trajectories. Hence a realistic prototype is not a pure vector, but a macrostate (which, however, is much compacter than the original macrostate). Consider this example: the robot may only have one sensor with the two possible output states ‘active’ or ‘inactive’. The sensor is activated if it touches an object. A sensory fingerprint shall thus be the mean voltage after scanning an object. This corresponds to a macrostate in a one-dimensional feature space; the mean voltage is the only feature. If the value is above a threshold, the output state is ‘active’, else it is ‘inactive’. In practice, this shall be realised by normalising the mean voltage such that it is either 1 (if an object has been touched) or 0. With this primitive sensor, the robot cannot distinguish between objects and the only concept is that of objects in general. The sensor states may be viewed as bit-values. These values are symbolic prototypes or, more precisely, indicators. They are realised as macrostates as the normalised mean voltage $V_n$ does not need to be perfectly equal for different sensory fingerprints. In fact, due to noise, $V_n$ will slightly fluctuate around either 0 or 1. In other words, the pure prototypes 1 or 0 are never actually physically realised, but the corresponding macrostates or microstate clusters are clearly distinguishable from each other. Hence, there is a well-defined mapping from the set of prototypes to as set of symbols.
1.2. COGNITIVE ASPECTS OF CLUSTERING

Operational concepts can be ‘symbolised’ in an analogue way: many actions have an all-or-nothing characteristic – either the action is executed in a stereotype way or it is not executed at all. Therefore, our robot may have a ‘mental’ prototype-like idea of what it means to go from A to B.

In summary, a symbolic representation of a category is defined by a mapping between the token that represents the category (respectively the corresponding prototype) and a symbol.

(IIb) Symbolic reasoning: In the toy world setting, symbolic representations are particularly simple as only two classes of categories occur. For example, the symbol $A$ may denote the class of all possible sensory fingerprints that belong to one object (object concepts), and the symbol $AB$ may denote the class of all direct routes from object $A$ to object $B$ (operational concepts). Symbolic reasoning refers to processes that can be appropriately conceived at the level of symbols. As the symbols denote categories, such processes connect concepts with each other. The rules for these processes must not be arbitrary, but should reflect some structural conditions of the robot’s world. As an example, if $A$ denotes a category, then...
the rules may tell us under what conditions a $B$, that stands for a second category, can be derived. This simple rule is an input-output relation that assigns an output symbol $B$ to an input symbol $A$. If $A$ and $B$ refer to objects, then we may say that the rule expresses that object $B$ can be reached from object $A$ on a direct path. Hence, the existence of the path defines the structural condition upon which the rule is based. Alternatively, we can interpret the relation itself as a representative of the operational concept ‘route from $A$ to $B’$.

Rules for symbolic reasoning must possess correlates at the level of the feature space description and at the level of the tokens. This is best explained by means of another simple example: take the logical NOT-function. It maps the symbol 1 onto 0 and vice versa. Hence, at the level of physics, the appearance of a token that codes for 1 forces the appearance of a token that codes for 0. The 0-input token and the 0-output token may be distinct physical states, but they can be represented by one prototype in the feature space. So the rule can be interpreted as a mapping between prototypes. For the execution of the rules, the symbols do no longer need to be grounded by referring to some category. We only need an unambiguous correspondence of tokens and symbols and a physical implementation of the rules (formal reasoning). This is the top-down architecture principle of computers.

(IIc) Non-symbolic processes: We have seen that the robot can acquire a symbolic description of its toy world by acquiring different concepts. This potentially allows for symbolic reasoning. The reason we have focused on symbolic reasoning is that it has ever served as a paradigm for human reasoning in general. Can we therefore conclude that any human reasoning is inevitably based on symbol manipulations (as computers are based on the manipulation of bits)? Our general answer is ‘no’. In section 1.1.4, we already argued that the fundamental clustering processes cannot be based on operations on symbols. They rather provide the symbols:

**Hypothesis 1:** Fundamental clustering processes are not imitable by formal operations on symbols. They are some kind of non-symbolic computation, serving the acquisition or allocation of fundamental concepts.

In other words, the dynamics are not governed by rules that describe the stepwise succession of clearly distinguishable states that can be labelled by symbols. In principle, though, all relevant computation in the brain could be symbolic and the non-symbolic processes could be unique to some trivial peripheral analogue-digital conversions. For the robot in the toy world, this is certainly a reasonable view if it is equipped with a digital ‘brain’. For real brains, we propose an alternative view:

**Hypothesis 2:** There are cognitively relevant processes in human brains that are not one-to-one representable by formal operations on symbols.

In particular, we anticipate this hypothesis for processes of decision making. How can we justify hypothesis 2? What does it imply? Many decisions humans make seem to have the character of stochastic inference or probabilistic reasoning. The result of such an inference process is a degree of belief in a decision. The underlying mechanisms (see chapter 2), need not rely on symbol manipulations, although we may be able to approximately emulate them.
by ‘symbolic computation’ (e.g., by a computer). Our statement is based on two arguments:
1) Probabilistic reasoning deals with probabilities that are more related to something continuous than discrete. Therefore, for the description of probabilistic reasoning, continuous dynamical systems seem to be more appropriate than symbol manipulating systems, such as Turing machines.
2) The real world is inherently noisy. A certain process will thus never be repeated in an absolutely identical manner. For instance, the firing of a neural circuit performing an inference will never be absolutely identical for two different episodes. A process thus seems to correspond to a unique micro-trajectory in the state space. One may argue that noise inevitably requires a coarse-graining or state clustering of the real-valued state space, as noise renders the idea of infinite precision of a process useless. This requires that brains introduce a partition of the state space, providing the macrostates that refer to the basic tokens upon which computation is based. Computation then amounts to implementing mapping rules between macrostates, and a process is described accurately as a sequence of macrostates. Some authors argued that spikes can be associated with macrostates [13]. Since they are discrete (all-or-nothing) events, they could serve as basic tokens of a neural code. We do not adopt this view as it requires that there is some very high frequency clocking signal in the brain that provides time bins which define the macrostates along the time axis. If there was a time binning, a spike train could be translated into a binary bit string that contains the full information necessary for neural computation. However, to our knowledge there is no evidence that there is a neural high frequency clocking.7 Neither is there clear evidence why the discrete character of spikes should play a crucial role for computation (although, for spike-timing codes, it is relevant that spikes are temporally localised events. For rate codes, spikes could just serve the purpose of optimal signal transduction). Even if certain quantities involved in the process are quantised and thus discrete, space and time are commonly still thought to be continuous quantities.8 The state space of a natural system therefore is essentially continuous, and information processing is directly concerned with microstates. Nevertheless, it seems plausible that instances of the same inference task form a bundle of trajectories in the state space, which could be recognised as a macrostate or a process scheme. This picture is in accordance with the probabilistic result of an inference process, for which it does not matter whether the process is a bit noisy and not identically reproducible. We summarise these considerations in the following definition:

**Definition 1.6** A non-symbolic process is a continuous sequence of microstates that is, as a matter of principle, unreplicable.

Mathematically, we may capture certain aspects of non-symbolic processes by means of stochastic differential equations: i) The ‘evolution of the process’ is not algorithmic, in the sense that it can be exactly captured by a finite list of operations on symbols. The rules for the evolution can be coded implicitly in the system. The performed computation is therefore inevitably bound to the system structure (embodied computation). ii) A single process is an unreplicable trajectory in the state space. iii) There might exist a reasonable stochastic

---

7It must be in the order of a typical neural refractory period, which corresponds to about 1000 Hz, whereas the fast Gamma-rhythm is only about 80 Hz [37].

8Some authors argue that the discrete aspects of quantum mechanics render the whole world Turing-like (see, e.g., [4] for counter-arguments).
description of whole bundles of trajectories that allows to identify process prototypes or schemes. iv) A single trajectory can possibly be emulated by symbolic computation, just as differential equations can be integrated numerically.

(III) Rules for symbolic reasoning reflect the structure of the toy world: Non-symbolic processes are intrinsically fuzzy-natured as there are no clearly defined and distinguishable macrostates. This fuzziness renders non-symbolic processes ideal to implement skills of flexible reasoning or decision making on the basis of noisy or unreliable units such as neurons. Symbolic representations, in contrast, offer an efficient way of representing and interchanging information about objects. Two robots in the toy world can easily ‘talk about’ an object if they agreed to assign the same symbol to it. Similarly, an act of symbolic reasoning can easily be communicated. How can the robot acquire rules for symbolic reasoning?

Ultimately, a symbolic description of the world is a structuring operation that induces further structure: since the robot can distinguish between different categories, it makes sense to consider relations between them. For example, objects can be subsumed in classes of similar objects which again can be unified in classes of classes etc. These relations reveal hierarchical structures. Furthermore, topological structures emerge as a consequence of the robot’s ability of acquiring operational concepts (planning routes). For instance, the robot may plan to go from object A to object B. If the plan is topologically feasible, we say that it represents a possible action. From an abstract viewpoint, possible actions are understood in terms of relations between the objects; ‘going from A to B’ is translated into the relation \((A,B)\). By means of clear rules, several relations can be combined to a new one. For instance, from \((A,B)\) and \((B,C)\), we derive \((A,B,C)\). In words, if ‘going from A to B’ and ‘going from B to C’ are possible actions, then the combination ‘going from A to C via B’ is also a possible action. The set of all possible actions defines a language on the alphabet of the symbols of distinguishable objects (e.g., \(A,B,C\)). This language is characterised by an underlying grammar (akin to the grammar of the symbolic dynamics of a dynamical system [17]). If the robot can reveal this grammar and finds out the generating rules, it can manipulate the symbols (and thus generate possible actions) without referring to their interpretation. In this sense, the ‘abstract formality’ of concepts induces a ‘syntactic formality’ which is a purely rule-based manipulation of symbols. This is the essence of formal reasoning. For now, we do not go into how the grammar could be learnt, but only explore the structure of this language.

### 1.2.2 The Language of ‘Possible Actions or Routes’

When the robot explores its environments it will realise that the toy space has some structure. This structure expresses a relation on the set of objects: the relation of object neighbourhood (or direct reachability). If we ignore metric aspects of the space, the structure can be visualised by an undirected connectivity graph (see Fig. 1.6). We draw a connection between two objects A and B, if A is reachable from B on a direct route without passing any other object, and vice versa. Mathematically, this structure is captured by a set of nodes \(K = \{n_1,...,n_N\}\), where each node \(n_i\) represents one of the \(N\) objects, and a symmetric adjacency matrix \(A\), where \(A_{n_i,n_j} = 1\) if the two nodes are connected and \(A_{n_i,n_j} = 0\) otherwise (here the indices \(n_i\) and \(n_j\) shall denote the corresponding row or column of \(A\); \(A_{n_i,n_j}\))
and $A_{ij}$ thus are equivalent). In the following, we introduce a couple of technical terms and definitions:

**Definition 1.7** A map $f_s : K \rightarrow C := \{a_1, ..., a_k\}$ is called a symbol map, if $C$ is a set of symbols and $f_s$ assigns a symbol to each node $n_i$.

It is natural to assume a one-to-one mapping between $K$ and $C$, which means that each object is characterised by one symbol and $N = k$. The robot can then describe a possible route as a set of concatenated symbols, e.g., $a_1a_4a_3a_1$. We may say that the robot symbolically represents a possible action. In principle, two or more nodes could also be mapped onto the same symbol. This means that in this case, the robot does not distinguish between the corresponding objects, or in other words, these objects are clustered to one class. As a consequence, the structure of the space looks different to the robot (see Fig 1.7). We will work out the substance of this statement below. For a fixed route length $l$, we define:

**Definition 1.8** The set of all possible routes of length $l$ is $P_l = \{b_1, ..., b_l | b_i \in C, A_{f^{-1}_s(b_i)f^{-1}_s(b_{i+1})} = 1\}$.

Not all possible concatenations of symbols from $C$ are possible routes. The condition is that the nodes corresponding to two successive symbols are connected. This is indicated by $A_{f^{-1}_s(b_i)f^{-1}_s(b_{i+1})} = 1$, where we may or may not allow for self-connections. We can now define a special formal language:

**Definition 1.9** The language of possible routes is $L_{A,f_s} = \bigcup_l P_l$.

A formal language [76] is a subset of the set of all finite-length sequences whose elements are drawn from a finite alphabet $C$. For certain languages, a formal generative grammar can be specified, which is a set of rules with which all sequences of the language can be derived. The grammars are classified according to the Chomsky hierarchy [12] whose classes (regular, context-free, context-sensitive, unrestricted) correspond to different classes of computational power [76]. Naturally, the question arises to which class the grammar of $L_{A,f_s}$ belongs. We can prove the following theorem (Appendix):

![Figure 1.6: A connectivity graph visualises the topological aspects of a certain toy space. Each dot represents an object (or object class) and a connection stands for reachability.](image)
Here, \( M_{a} \)

**Definition 1.10** A concatenating grammar on \( C \) is a regular grammar with the production rules:

1) \( \epsilon \rightarrow a_{1} \), where \( \epsilon \) denotes the empty string and \( a_{1} \in C \).

2) \( a_{1}...a_{k} \rightarrow a_{1}...a_{k+1} \), where \( a_{k+1} \in C \) and \( M_{a_{k}a_{k+1}} = 1 \).

Here, \( M \) is some (symmetric) connectivity matrix that defines whether a symbol \( a_{i} \) is reachable from \( a_{j} \) and vice versa. Note that \( M \) is, unlike the matrix \( A \), directly defined on the alphabet \( C \). A language that can be described by a concatenating grammar is easy to implement. Firstly, we do not need to introduce non-terminal symbols (for a definition, see Appendix) and secondly, the main rule (2) possesses a first order Markov property: whether a symbol can be added to a string only depends on the last symbol of the string.

To keep the description of its world simple, the robot should prefer a small set of symbols, but it should also prefer a simple grammar to plan its routes. Interestingly, for general spaces there is an antagonistic interplay between the size of the alphabet and the simplicity of the grammar. The grammar is simple for a large alphabet (if \( f_{s} \) is one-to-one), becomes more complicated for a smaller alphabet and is again simple for a very small alphabet (if \( f_{s} \) projects to one symbol only; the grammar is ultimately simple). This is expressed by the following theorem:

**Theorem 1.1** \( L_{A,f_{s}} \) is described by a regular grammar.

What happens if \( f_{s} \) is not injective? In this case, the robot does not distinguish between all single objects, but only between classes of objects (say: ellipses and rectangles). At a first glance, the world is seemingly less complicated (see Fig.1.7). There is, however, a ‘hidden complexity’ because the spatial structure behind each symbol does not just disappear. Hence, the ersatz graph in Fig.1.7 b) is actually wrong. To understand this ‘hidden complexity’, we define:

**Definition 1.10** A concatenating grammar on \( C \) is a regular grammar with the production rules:

1) \( \epsilon \rightarrow a_{1} \), where \( \epsilon \) denotes the empty string and \( a_{1} \in C \).

2) \( a_{1}...a_{k} \rightarrow a_{1}...a_{k+1} \), where \( a_{k+1} \in C \) and \( M_{a_{k}a_{k+1}} = 1 \).

Here, \( M \) is some (symmetric) connectivity matrix that defines whether a symbol \( a_{i} \) is reachable from \( a_{j} \) and vice versa. Note that \( M \) is, unlike the matrix \( A \), directly defined on the alphabet \( C \). A language that can be described by a concatenating grammar is easy to implement. Firstly, we do not need to introduce non-terminal symbols (for a definition, see Appendix) and secondly, the main rule (2) possesses a first order Markov property: whether a symbol can be added to a string only depends on the last symbol of the string.

To keep the description of its world simple, the robot should prefer a small set of symbols, but it should also prefer a simple grammar to plan its routes. Interestingly, for general spaces there is an antagonistic interplay between the size of the alphabet and the simplicity of the grammar. The grammar is simple for a large alphabet (if \( f_{s} \) is one-to-one), becomes more complicated for a smaller alphabet and is again simple for a very small alphabet (if \( f_{s} \) projects to one symbol only; the grammar is ultimately simple). This is expressed by the following theorem:
1.2. COGNITIVE ASPECTS OF CLUSTERING

(a) Spatial arrangement of objects.  (b) Corresponding Markov graph with symbols $a$ and $b$.

Figure 1.8: For this setting, the possible routes cannot be described by a concatenating grammar.

**Theorem 1.2** For a symbol map $f_s$, the following statements hold:

a) If $f_s$ is one-to-one, then $L_{A,f_s}$ can always be described by a concatenating grammar.

b) If $f_s$ is not injective, then there is no guarantee that $L_{A,f_s}$ can be described by a concatenating grammar.

c) If $f_s$ is all-to-one (only one symbol), then the grammar of $L_{A,f_s}$ is trivial.

Obviously, if $f_s$ is all-to-one, the grammar must be trivial as all sequences consist of one symbol only. We also see immediately that for the case ‘$f_s$ is one-to-one’, there is a concatenating grammar as we can identify $M = A$. However, the language that corresponds to the non-injective setting in Fig. 1.8 does not have a concatenating grammar. This is because only an odd number of the symbol $a$ between two $b$s is possible. For example, $bbaaaab$ or $bbaaaaab$ are possible routes, but $baab$ is not possible. Therefore, it does not only depend on the last symbol of the string $s$ whether a new $b$ can be added to $s$ (a similar argument holds for the setting in Fig. 1.7).

Given a certain space, what symbol map $f_s$ should the robot eventually choose? This question does not have a clear-cut answer. If the sparseness of the representation was the only criterion, then the robot should always prefer an all-to-one mapping. This might be a good solution, if all objects have absolutely identical properties. In reality, there is a variety of different objects – perhaps hostile and friendly ones – and it makes sense to distinguish between them to optimise the survival perspectives. In addition, if $f_s$ is not all-to-one, it can even make sense to distinguish between two identical objects (e.g., the two $a$s or $b$s in Fig. 1.8) in order to simplify the grammar of $L_{A,f_s}$. This demonstrates a) that two identical objects can still be perceived as different due to their different ‘topological position’ and b) that an optimal representation of the space is usually obtained in a trade-off between the number of distinguishable objects and the simplicity of the grammar.
1.2.3 Recapitulation: Cognitive Aspects of Clustering

In this section, we adopted the view of Connectionists that basic neural processes are not based on symbol manipulations. We therefore have to explain how symbolic reasoning is possible on the basis of non-symbolic processes. We developed the notion that fundamental clustering processes serve the purpose of concept acquisition and allow for a symbolic representation of objects. By means of a simple toy world model, we demonstrated how rules for symbolic and formal reasoning can result as a consequence of the symbolic encoding of objects and as a consequence of their topological relations. This section provided an abstract view on the cognitive aspects of clustering. The biological relevance of these considerations remains vague. In the next section, we will try to fill out this gap by discussing a couple of models for neural mechanisms of concept acquisition.

1.3 Neural Models of Concept Acquisition

In the real world, mental representations and cognitive functions are implemented in some hardware. While in the case of robots the implementation is based on electronic circuits, humans and animals possess a ‘neural implementation’. To reveal the main features of this implementation is the core challenge of theoretical neuroscience. According to Connectionists, three major problems have to be solved [99]. First, the problem of flexible human reasoning: what mechanisms underlie the human style of inference? Second, the problem of neural plausibility: how does symbolic behaviour emerge from the activity of neural elements? Third, the problem of self-organisation: how do reasoners construct appropriate representations based on experience?

How do these questions fit into the picture we have developed so far? We identified two principal ways of computation: symbolic computation, which ultimately culminates in purely formal symbol manipulation, and non-symbolic computation, which are processes based on a continuous state space. Symbolic computation raises two issues that overlap with the problems raised by Connectionism:

(I) How are concepts acquired and represented in neural networks?
(II) How is symbolic reasoning realised in neural networks?

The acquisition of fundamental concepts has been associated with non-symbolic clustering processes. We also hypothesised that stochastic inference processes may be non-symbolic. This leads to the question:

(III) How are non-symbolic processes implemented at the neural level?

Methodologically, we follow the Connectionism paradigm by trying to identify plausible mechanisms at a relatively schematic level, rather than identifying certain actual brain structures responsible for cognitive processes.
1.3. NEURAL MODELS OF CONCEPT ACQUISITION

1.3.1 (I) Concept Acquisition

In our notion, the acquisition of concepts is equivalent to learning certain categories or classes. The problem of conceiving the world by means of concepts thus amounts to finding and performing classifications. In the following, we want to focus on the assumption that concepts have to be learnt. We thus do not discuss the case, where the representation of categories is based on innate (genetically predetermined) processes.

This task of finding and performing classifications divides into two aspects. First, finding classifications is the task of identifying categories by means of a set of unclassified samples. Technically speaking, for a given data set \( A \), a map \( f_c \) onto a set of class labels \( C \) has to be found. Second, performing classifications is the assignment of a data item \( a \) to an afore established class according to some learnt rule or mechanism. The distinction between finding classifications and performing classifications reflects two distinct classification approaches discussed in the literature (e.g., [77]). That is, the approach of unsupervised data clustering and the approach of classifiers that are trained by a supervisor. While in the literature classifiers are often considered as realistic models for neural classification processes [33], clustering algorithms are less frequently treated as a basis for realistic neural models. Nevertheless, they could give a hint on neural mechanisms of concept acquisition. We now sketch different models that could explain neural classification processes. We, however, do not comment on the plausibility of these models – this judgment lies beyond the methodological nature of this thesis. Empirical evidences have been found for diverse mechanisms, so that neural circuits could in principle implement all of the following ideas.

Read-out neurons as simple neural classifiers?

A traditional classifier is characterised by a non-invertible map \( f \) from a continuous input space to a discrete output space. The simplest classifier, the linear perceptron [26], separates a \( m \)-dimensional real-valued input space into two classes (denoted by 1 and \(-1\)). Mathematically, the classification corresponds to a hyperplane separation in the input space, described by the equation

\[
 f(x) = sgn\left(\sum_{i=1}^{m} w_i x_i + b\right),
\]

where \( x = (x_1, \ldots, x_m) \) is the input, \( b \) is a bias, the \( w_i \)'s are interpreted as synaptic weights, and \( sgn \) is the signum function. Concerning its structure, the perceptron is a feedforward network with input channels that encode some features of an input signal and a read-out neuron that encodes the classes. In the terminology of section 1.2, the two separated parts of the input space are (very large) macrostates and the read-out values are the class indicators. The position of the hyperplane is learnt on the basis of a training set that contains data with known class assignments. Learning is equivalent to tuning the synaptic weights. The advantage of classifiers is that they offer a fast classification of new input data. As a disadvantage, this kind of learning is absolutely supervised. Hence, such classifiers either rely on some explicit supervisor signal that can ‘teach’ them or they rely on some reinforcement signal that serves as an implicit supervisor. This renders the question how classifiers in real brains are tuned nontrivial. In order to bridge the gap between abstract perceptrons and biologically more realistic models, the states 1 and \(-1\) are interpreted as two different firing states of the classifier neuron (e.g., active and silent). In reality, a biological neuron will
never be a system with exactly two clear-cut states. On the one hand, in the silent state, a spike can be emitted spontaneously from time to time, due to various noise sources. On the other hand, in the active state, the firing rate will always vary a bit around a mean value. The two states 1 and $-1$ are thus best characterised as macrostates. This is in accordance with the idea of a macrostate characterisation of prototypes or indicators, proposed in section 1.2. As the indicators represent categories, the read-out neuron encodes a category by its activity. A neuron with this behaviour is often called a *grandmother neuron*. An interesting model referring to read-out neurons are ‘liquid state machines’ [57]. Here, read-out neurons can play the role of (possibly nonlinear) classifiers, fed by a large pool of recurrently connected neurons. The pool blows up the dimensionality of the input space, which facilitates the linear separation for certain classification problems. Additionally, the pool serves as an information storage. Liquid state machines offer a model for real-time processing, which is an essential aspect of biological neural networks [45]. However, classifications based on a single read-out neuron are not robust: if the neuron fails or dies, the classification ability is lost, unless there are other copies of the same classifier.

**Tokens obtained by phase locking?**

Another classification approach is provided by the idea of phase locking of biological neurons [92]. Two – in a biological sense – strongly coupled neurons, driven by two input currents $I_1$ and $I_2$, can engage in phase locking, defining a map from the input currents to a periodicity $l$, i.e.,

$$f : I_1, I_2 \rightarrow l.$$  \hspace{1cm} (1.17)

This is a map from a two-dimensional continuous input space to a discrete output space, exhibiting intriguing optimal coding properties similar to Huffman coding [92]. Phase locking has been observed in vitro and could serve as a kind of A/D-converter mechanism in the brain that provides the basic tokens for processes of symbolic reasoning. Similar to the above-discussed classifiers, the mapping $f$ must be learnt by adjusting the synaptic weight(s). The principles of this learning remain to be explored. In practice, we often have to deal with unsupervised category learning that does not seem to rely on an explicit supervisor or reinforcement signal, but seems to be related to some kind of self-organisation process. What kind of neural mechanisms could underlie such a process?

**Clustering algorithms as models for concept acquisition?**

In machine learning, unsupervised category learning is provided by clustering algorithms. We may thus use clustering algorithms as potential guidelines for processes of unsupervised category learning in brains. However, many clustering algorithms are not directly suited for a neural implementation as they are formulated as an off-line procedure. That is, all data items are given at once, whereas in reality, data comes in over time (on-line clustering). Hence, in order to use a clustering algorithm as a paradigm for unsupervised category learning, the algorithm has to be translated into an on-line procedure (an example is given by on-line k-means [111]). By trying to find a neural implementation or interpretation of various clustering algorithms – some of them are reviewed in a short survey in section 3.1 – we identified different ‘design principles’: 
1.3. NEURAL MODELS OF CONCEPT ACQUISITION

1) **Radial basis function or delay lines approach:** This model shall only briefly be mentioned. Details are found in [31, 58]. In this model, clustering is performed on the basis of a feed-forward network with an unsupervised SOM-like (self-organising maps [42]) learning feedback. The input layer encodes the features of a data item by means of a spike timing code. The output layer consists of RBF-like neurons (radial basis functions [26]) that encode the classes by means of a winner-take-all mechanism. This approach is a combination of (nonlinear) feedforward classifiers and ideas of unsupervised learning.

2) **Attractor clustering in recurrent networks:** Models of attractor computation have a long standing tradition in theoretical neuroscience [27, 30, 78, 83], with increasing experimental evidence [105]. We quickly outline the concept on the basis of the simplest case, the point attractor paradigm, which has been established by J.J. Hopfield [30]. We assume that our system under consideration (e.g., a neural circuit) is characterised by a \( m \)-dimensional state space variable \( x \) whose evolution is described by a system of autonomous ordinary differential equations, i.e.,

\[
\frac{dx(t)}{dt} = F(x(t)).
\]  

(1.18)

We further assume that the state space decomposes into a finite set of basins of attraction. Within a basin of attraction, each initial point \( x(0) \) evolves towards a point attractor \( x_a \). By interpreting \( x(0) \) as input and \( x_a \) as output, we can naturally define a map \( f \) from the state space to the set of attractors,

\[
f : x(0) \in \mathbb{R}^m \rightarrow x_a \in \mathbb{A}.
\]  

(1.19)

In the terminology developed in section 1.2, a basin of attraction can be identified with a macrostate, and a point attractor is identified with a prototype. The task of finding (better: learning) a reasonable \( f \) amounts to establishing the set of attractors \( \mathbb{A} \) and their basins on the basis of a sequence of inputs \( \xi \).

The idea is best explained by means of an example: for the discrete variant of the Hopfield model [55], a learning of attractors is possible by imposing a simple Hebbian-like learning rule. Consequently, concept acquisition is described as a bottom-up self-organisation. The influence of the input patterns can be described figuratively as ‘digging a hole (attractor) into an energy landscape’. Fig. 1.9 demonstrates that in this model, point attractors are established by extracting prototypes from a sequence of structured input. The network consists of \( m \) binary elements (state 1 = black, state -1 = white) that are symmetrically connected (\( w_{ij} = w_{ji} \) denotes the synaptic strength between neuron \( i \) and \( j \) and \( w_{ii} = 0 \)). The dynamics is given by a (either synchronous or asynchronous) update rule:

\[
x_i(t + 1) = \begin{cases} 
1 & \text{if } h_i > 0 \\
-1 & \text{if } h_i < 0 \\
x_i(t) & \text{if } h_i = 0,
\end{cases}
\]  

(1.20)

where \( x_i \) is the state of neuron \( i \) and \( h_i := \sum_{j=1}^{m} w_{ij} x_j \), \( i = 1, 2, ..., m \).

After convergence, a new input \( \xi^{n+1} = (\xi_1^{n+1}, ..., \xi_m^{n+1}) \) is received and the weights are

9The continuous Hopfield network would be more appropriate to model a fundamental clustering process because the discrete variant is clearly Turing-compatible. We chose the latter for simplicity.
modified according to a Hebb learning rule\textsuperscript{10}:

\begin{equation}
    w_{ij}^{n+1} = \frac{1}{m} \Phi(mw_{ij}^n + \beta \xi_i^{n+1} \xi_j^{n+1}).
\end{equation}

For $\Phi$, we use a monotonically increasing odd function, as for instance: $\Phi(x) = \tanh(x)$. Here, $\beta$ regulates the learning impact. As $\Phi$ is bounded, it assures that old input is constantly fading away. This is important in order to prevent catastrophic forgetting, i.e., the problem that many spurious attractors will prevent any useful prototype association if the number of prototypes to be stored becomes to high \cite{96}.

In the picture sketched here, attractors correspond to arbitrary activity patterns of the whole network. Winner-take-all networks (e.g., \cite{61} and references therein) can implement

\textsuperscript{10}For the actual simulation, the synaptic dynamics is a bit more complicated (see Appendix).
an alternative form of attractor networks. To illustrate this idea, we consider again a relatively simple toy model, for which we neglect the unsupervised learning aspect. The model consists of two populations of simple binary neural elements $s_i \in \{-1, 1\}$. There are symmetric excitatory connections $J_{ij} = J_{ji} = 1$ between the neurons within each population. Neurons of different populations interact via inhibitory fields, i.e., each population induces an inhibitory field $I$ felt by the neurons of the other population (the situation is depicted in Fig. 1.10 a)). The field strength $|I|$ is proportional to the activity of the population (proportional to the number of $+1$). The underlying dynamics is discrete and sequential, i.e., at each update step $t$ only one single, randomly drawn neuron will be updated according to

$$s_i(t + 1) = \text{sgn} \left( \tanh \left( \frac{h_i(s(t))}{T} \right) + \eta_i(t) \right).$$  \hfill (1.22)

$\eta_i(t)$ is an independent random number (representing threshold noise) uniformly drawn from the interval $[-1, 1]$. $T > 0$ controls the noise impact. $h_i(s(t))$ is the local field calculated after

$$h_i(s(t)) = \sum_j J_{ij} s_j + b_i - I_{\text{other}}.$$  \hfill (1.23)

The sum $\sum_j$ is over all connections of neuron $i$ within its population, $b_i > 0$ is a bias or external input that breaks the symmetry and makes the state $+1$ the preferred one, and $-I_{\text{other}}$ is the inhibitory field induced by the other population. The model is an Ising spin model or could also be treated as a special Hopfield network. The Metropolis-like dynamics

Figure 1.10: a) A winner-take-all network with two neural populations (see text). b) The shape of a minimum of the energy landscape corresponding to the activity of one population (see text) depends on the activity of the other population. In the final attractor state, one minimum is enhanced and the other one is suppressed (dashed line).
exhibits a Boltzmann (Gibbs) distribution as the steady-state distribution \[63\], with the energy function,

\[
E = \left( -\frac{1}{2} \sum_{i,j} s_i s_j - \sum_i b_i s_i + \sum_i I_2 s_i \right) + \left( -\frac{1}{2} \sum_{k,l} s_k s_l - \sum_k b_k s_k + \sum_k I_1 s_k \right),
\]

where we have split the energy into the contributions from population \(p_1\) and population \(p_2\). A final attractor of the network corresponds to a state for which one population (the winner population) is in the firing state 1 and the activity of the other population is suppressed (state \(-1\)). The dynamics modulates the energy landscape as the inhibitory fields \(I_1, I_2\) are changed. For two equivalent populations and symmetric initial conditions, both populations are equally likely to win. This is indicated by two symmetrical minima in the energy landscape (see Fig. 1.10 b)). If, however, there are slight inhomogeneities in the initial conditions or in the biases, only one minimum survives, whereas the other minimum vanishes. This minimum constitutes the final attractor state (see Fig. 1.11). The whole process can again be interpreted in terms of a mapping between the initial condition and the attractor. The mapping is interpreted as a categorisation, where the attractor plays the role of an indicator. Due to the noise, the indicator must be associated with a set of states, which demonstrates its macrostate character.

3) **Grandmother cell clustering:** In this paradigm, an object is supposed to be encoded in the sparsest possibly way – by a grandmother cell \[16\]. Clusters are then formed by the coordinated activity of groups of neurons, reflecting categories of similar objects. In the simplest case, clusters correspond to groups of synchronised neurons. There are two interesting remarks. First, grandmother cells already represent some fundamental concepts as they stand for objects. This is in contrast to 1) and 2), where the neurons represent...
features or some feature-like quantities derived from sensorial activity. Clusters are obtained by binding together objects through synchronisation. Second, synchronisation has also been proposed as a means to solve the general ‘binding problem’ (binding together different modalities of the same object). The suggestion is discussed controversially [13]. Here, we illuminate the idea of grandmother cell clustering on the basis of a simple I&F (integrate-and-fire) model with short-range excitation. The model does not primarily reflect a realistic biological situation, but reveals some general principles of network self-organisation [101]:

- Global order can arise from local interaction.
- There is an interplay between network activity and network connectivity.
- Initial inhomogeneities in synaptic weights tend to self-amplify.
- Modifications in synaptic weights tend to cooperate.

In our model we consider $N$ simple leaky I&F-neurons whose membrane potentials $u_i$ are described by

$$C \frac{du_i}{dt} = I_i(t) - \frac{u_i}{R}. \quad (1.25)$$

If $u_i(t)$ reaches the threshold $\theta$, a spike is emitted and $u_i$ is reset to 0. The input current $I_i(t) = I^i_{ext} + I^i_{int}$ consists of a (time-independent) external current $I^i_{ext} = I_{ext}$ and the contributions of the presynaptic neurons

$$I^i_{int} = \sum_{j=1}^{N} J_{ji}\sum_{k} \delta(t - t^k_j), \quad (1.26)$$

where $t^k_j$ is the emission time of the $k$th spike at neuron $j$ and $J_{ji}$ is the synaptic efficacy.

The simplicity of the used IF model allows for an exact integration. If the potentials at time $t$ are given as $(u_1, u_2, ..., u_N)$, the time interval to the next spike is given by $T_k = \min_j \{T_j\}$, where

$$T_j = RC \ln \left( \frac{u_j(t) - I_{ext} R}{\theta - I_{ext} R} \right). \quad (1.27)$$

The resulting potentials for $j \neq k$ at time $t + T_k$ are

$$u_j = I_{ext} R(1 - \exp \left( \frac{-T_k}{RC} \right)) + u_j(t) \exp \left( \frac{-T_k}{RC} \right) + J_{kj} R.$$

As there is no transmission delay in this model, an outgoing spike can provoke a whole bunch of simultaneous spikes at further sites. This gives the possibility of quick synchronisation among tightly coupled neurons. One neuron, however, can only fire once at a given time. The internal parameters were chosen according to $RC = 8$ ms and $\theta = 16$ mV. The external current was chosen as $I_{ext} R = 25$ mV. The initial network connectivity is given by weak (symmetric) excitatory connections between each neuron and its $k$ nearest neighbours. The distance between two neurons is supposed to reflect the similarity of the objects that are represented by the cells (topographic map or data structure, see Fig. 1.12). The grand-mother cells of similar objects lie close to each other. The network’s task is to develop its structure, ending with clearly distinguishable networks components that internally have a
strong connectivity and thus lead to synchronised activity of the involved neurons. The initial connection strengths $J_{ij} = J_{ji}$ are given by a decreasing function of the neural distance $d_{ij}$, reflecting the data structure, i.e.,

$$J_{ij} = J_{ji} = \exp\left(-\frac{d_{ij}^2}{a^2}\right).$$  \hspace{1cm} (1.28)

A good choice for the parameter $a$ is related to the average distance between connected sites $d$. E.g., for $k = 10$, we take $d/4$. The potentials $u_i$ and $u_j$ of two connected neurons tend to be more correlated if the connection $J_{ij}$ is stronger. This can be used to ‘breed’ clusters by means of Hebbian-like learning. In this paradigm, a synapse is strengthened if the firing at the presynaptic and the postsynaptic neuron is more or less coincident. As the synapses are symmetric, we define a simple learning rule in the following way

$$J_{ij} = J_{ji} \rightarrow \min\{2J_{ij}, 1\}, \quad \text{if} \quad G_{ij}^\tau(t) = 1,$$  \hspace{1cm} (1.29)

where $G_{ij}^\tau(t) = 1$ only if both of the two neurons $i$ and $j$ have emitted a spike within $[t-\tau, t]$. $	au$ defines the width of the learning window and is chosen in relation to the interspike interval of independently spiking neurons $T = RC\ln(I_{ext}R/(I_{ext}R - \theta)) = 8.17$ ms. If $	au$ is too small, learning is slow. If $	au$ is too large, learning is too fast and quickly leads to global synchronisation. For the given parameters, we chose $\tau = T/4 = 2$ ms.

The effect of Hebbian learning for a toy system can be observed in Fig. 1.13 b). The neurons completely synchronise in clusters after about $100T \approx 200$ ms. This is in stark contrast to the situation without learning (Fig. 1.13 a)). In Fig. 1.14, the evolution of the corresponding connectivity structure is illustrated. While at the very beginning only a few strong connections are present (the seed of the clusters), after 200 ms the network has developed three large connectivity components, reflecting the three conceptual clusters.

Clustering is certainly unlikely to be neurally realised according to this simple model. For realistic situations, more detailed models must be elaborated (e.g., the strict grandmother

\[ \text{Figure 1.12: A two-dimensional data distribution with three cluster structures. A data point corresponds to an object described by two features. In the grandmother cell paradigm, the data distribution is translated into a network whose synaptic weights reflect the similarity of objects.} \]
Figure 1.13: Spike diagram a) without learning and b) with learning. Time in ms.

Figure 1.14: Structure of strong connections with $J_{ij} > 0.2$ at time a) $t = 0$ ms and b) $t = 300$ ms.
coding is unrealistic because it is not robust). Conversely, the self-organisation principles behind the model turn out to be fruitful for the design of powerful clustering algorithms. The sequential superparamagnetic clustering algorithm, see chapter 3, shows in fact a great resemblance. Other clustering algorithms can also be interpreted according to the grandmother clustering scheme [63]. Such an interpretation may inspire further developments in clustering technology.

Closing the feedback loop

According to the idea that (neural) clustering processes are based on the principle of embedded prediction optimisation (section 1.1.2), the clustering result is validated for improvements. In the unsupervised learning paradigm, the validation (or reinforcement) signal is not fed back to the clustering itself. Instead, the input or initial condition of the clustering is modified. In the case of the delay lines approach and in the case of the attractor clustering approach, the validation effects a modification of the input structure. This could partially happen via a tuning of the sensors (as in the case of the cochlea [38]). In the case of the grandmother clustering model, the validation must effect the initial synaptic weights that serve as input in this approach. The picture of a closed feedback loop is in accordance with the two-stage picture for clustering algorithms that we will discuss in chapter 3.

1.3.2 Symbolic Reasoning (II) and Non-symbolic Processes (III)

(II) How could symbolic or formal reasoning arise from neural activity? What kind of neural processes could provide the basis for, e.g., the mental act of logical deduction?

In [34], a sketch of a neural framework is presented for the formation of simple predicates, which are postulated as basic modules for more complex semantic structures. Hence, this study is an attempt to trace back logic to a neural basis. To summarise the main idea in [34], the logical formula PREDICATE(x), which denotes a predicative relation, is a simplifying schematic representation of the integration by the brain of two broadly separable processes. One process is the delivery by the senses (visual and/or auditory) of information about the spatial location of an object. The other process is the analysis of the object by the perceptual (visual or auditory) recognition subsystems in terms of its properties. The problem of finding a neural mechanism for the combination of a predicate (property) and an object thus amounts to solving the neural ‘binding problem’.

Conceptually, we do not follow this article, but we bring together what we have inferred about symbolic and formal reasoning in section 1.2:

1. We defined symbolic reasoning as any kind of computation (or thinking) that relies on the rule-based manipulation of symbols. The rules might be implicitly coded in the environment. In our toy world robot example, planning a route is considered an act of symbolic reasoning as a route is understood as a relation between symbols that encode objects. The rules for valid routes and hence for valid relations are implicitly coded in the robot’s environment.

2. We defined formal reasoning as an act of symbolic reasoning that ignores the meaning of the symbols (symbol grounding/ reference to concepts). Computation carried out by Turing machines or finite automata is of this type. Formal reasoning relies on an
explicit (algorithmic) formulation of the rules after which the symbols are manipulated. In the robot example, this corresponds to the step of learning the generative grammar of the language of possible routes (section 1.2.2).

3. We developed the notion that symbolic reasoning is based on mappings between prototypes that are associated with symbols. A natural idea is that these mappings between prototypes translate into mappings between the corresponding tokens at the physical level. We mentioned several possibilities for symbol-representing tokens in neural circuits, most prominently we have the following: First, if a neuron only exhibits a number of clearly distinguishable firing states, we may assign a symbol to each firing state. Second, following the attractor paradigm, a symbol may be assigned to each attractor of a network.

This translates into the following two models for symbolic reasoning: In the first case, the neurons themselves are the carriers of the symbols. Consequently they are also the sites where the symbol manipulation takes place. Early neural models, such as the McCulloch-Pitts model [51] or the discrete Hopfield model [55], belong to this case. Here, the symbol manipulation is in the form of local Boolean operations. As in these models the time is also discretised or clocked, neural circuitries are treated as ‘logical machines’. As a consequence, such models are equivalent to finite automata [60] that provide a standard notion of formal reasoning. This is the picture of a completely algorithmic brain. For this picture, the open question is of course: how do the neural algorithms reprogram themselves? In particular: how is the right symbol coding and the appropriate network structure implemented? In the second case, the neurons are supposed to be analogue computational devices. The symbols and the symbol manipulations are identified at the level of whole populations of neurons. We can only speculate about how mechanisms of symbolic reasoning could be realised at this level. Here come two crude suggestions: 1) The activity corresponding to one attractor in a neural circuit may trigger the activity of a second attractor in a distinct neural circuit. This would implement the simple implication ‘If $a$ then $b$’. A simple generative grammar (of the type ‘concatenating grammar’, see section 1.2.2) could simply be realised by chains ‘If $a$ then $b$ then $c$ then $\ldots$’. Alternatively, the simultaneous activity of two (or more) attractors in distinct neural circuits may trigger the activity of an attractor in a third circuit. This would be a simple mechanism for realising the implication ‘If $a$ and $b$ then $c$’. Such a mechanism could also be used for signaling a binding of two ‘objects’ $a$ and $b$. 2) In contrast to the situation of attractors in several distinct circuits, all attractors could be realised in one neural circuit. In this case, only one attractor can be activated at once. Hence, an implication such as ‘If $a$ then $b$’ must be realised in a temporal order, i.e., the activity of an attractor $a$ triggers the subsequent activity of an attractor $b$. A generative grammar (of the concatenating type) can be realised as a Markov chain model, where the probability of noise-induced jumps between attractors reflects the transition probabilities between the Markov states.

(III) Non-symbolic processes: According to our considerations in section 1.2, the main characteristics of a non-symbolic process are better captured by differential equations (DE) than by computer algorithms. Therefore, we argue that, insofar as DEs are the appropriate concept to describe nature, any natural process is intrinsically non-symbolic. The distinction between non-symbolic processes and symbolic processes lies in the level of signification. A
non-symbolic process implements a symbolic process if the process can be interpreted as a mapping between macrostates. In contrast, the significance of a non-symbolic is at the level of microstates. That is, a non-symbolic process must be interpreted as a mapping between i) microstates, ii) a macrostate and a microstate, or iii) a microstate and a macrostate. Fundamental clustering processes can be understood as instances of case iii). In section 1.2.1 we speculated that certain mechanisms of human reasoning, such as decision making, may also be non-symbolic. The first part of chapter 2 is dedicated to this hypothesis.

1.4 Some Philosophical Reflections

In this chapter, we discussed the idea that non-Turing processes, akin to analogue-to-digital conversions, could underlie the cognitive act of concept acquisition. The role of non-Turing processes for understanding human cognition has been debated controversially. Partially, the discussion was motivated by the opposition to a strict machine-state functionalism (H. Putnam, ‘Minds and Machines’, 1960). It has been argued that the purely syntactic treatment of symbols can hardly account for semantics or phenomena such as creativity or novelty of thoughts (J. Searle, ‘Chinese room argument’, 1980). This observation has also been found reflected by the argument that a Turing machine cannot prove Gödel’s theorem (see: Can a Turing machine know that Gödel’s sentence is true? [49]). In summary, the core argument goes as follows: humans can see the existence of true (arithmetic) sentences that cannot be seen by computer algorithms. This is because for a pure syntactic system, truth is equivalent to formal proofability, and Gödel’s theorem for the system cannot be proved within the system. R. Penrose used this argument to conclude that some cognitively relevant non-Turing (non-algorithmic) processes must take place in human brains and offered some rather controversial explanations ([41, 71]). While our considerations support the opinion that non-Turing processes could play a relevant part in cognition, our considerations in this chapter suggest an alternative view on the issue: First, the embedding of a system in an ever-changing environment and the ‘embodiment of intelligence’ demand for non-Turing-like fundamental clustering processes. Second, a fundamental clustering can be qualified as a concept learning process that provides new symbols. This is ultimately the limitation of a Turing machine: it is unable to create completely new symbols (or concepts respectively) which may allow the system to change to a meta-level perspective.

Our argument in favour of non-Turing processes is based on the assumption that real-world input is, as a matter of principle, analogue and unrepresentable. From this, we concluded that continuous state spaces provide the adequate mathematical structures to capture the essence of the computational processes that underlie concept acquisition. This assumption must be accepted as an unproved hypothesis. To what extent fundamental clustering processes can really account for putative non-Turing or even super-Turing capabilities remains a matter of dispute. We could only point out some aspects, driven by thoughts about the cognitive role of clustering and classification.
Chapter 2

Belief Propagation BeP

2.1 Probabilistic Reasoning: A Paradigm for Non-symbolic Computation

All men are mortal. All Greeks are men. Hence all Greeks are mortal.

This is Aristotle's famous example for syllogistic reasoning or deductive logic. Nobody is in doubt about the correctness of the conclusion. Being so impressively convincing and seemingly trivial, syllogism has served as the paradigm for (correct) human reasoning for many centuries. However, human reasoning cannot always be cast into such a clear-cut scheme. Take the following simple example:

'My dog barks about three times a week during the night, either due to a cat or due to a burglar. During one year, one burglar showed up. How likely is it that a burglar is around when my dog barks?'

Our feeling certainly tells us that it is very unlikely, although it seems hard to quantify this feeling. So there is no clear-cut conclusion to the type of question posed, and the problem is somehow very different from the syllogism example above. Quantitatively, we can approach the problem by means of Bayes' theorem [9, 46]. Let $p(A) = 1/365$ be the prior probability for a burglar and $p(B) = 3/7$ be the marginal probability for barking. We can further assume that the dog always barks if a burglar is around, so $p(B|A) = 1$. Then Bayes' theorem gives us the probability that a burglar is at work when the dog barks:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} = \frac{1/365}{3/7} = 0.0064.$$  \hspace{1cm} (2.1)

In line with our feeling, the probability is very small.

In practice, the relevant question is whether the police should be called when the dog barks. The decision must be related to the posterior probability $p(A|B)$. In order to make the decision, humans must possess an internal model that expresses the knowledge about the single concepts, such as 'barking' and 'burglar', and their (causal) interrelations. Such an internal model can be expressed in a convenient way by a Bayesian belief network (BBN) [9]. BBNs are directed graphs in which each node represents a probabilistic variable and...
the edges point out the conditional dependencies. In Fig. 2.1, the BBN corresponding to the dog example is depicted. Node $A$ represents the variable $x_a$ (where $x_a = 1$ means that a burglary happens and $x_a = -1$ means that no burglary happens), node $C$ represents $x_c$ (where $x_c = 1$ means that a cat is present and $x_c = -1$ means that no cat is present) and node $B$ represents $x_b$ (where $x_b = 1$ means the dog barks and $x_b = -1$ means that the dog does not bark). The arrows indicate the conditional dependencies $p(B|A)$ and $p(B|C)$.

For making the decision (call the police?), one is interested in the probability that the dog barks and a burglar is present $p(x_a = 1, x_b = 1)$. This is, in our example, equivalent to the probability $p(x_a = 1)$ that a burglar is present, as the dog always barks when a burglar is present ($p(x_a = 1) = p(x_a = 1, x_b = 1) + p(x_a = 1, x_b = -1) = p(x_a = 1, x_b = 1)$). In order to estimate or to infer probabilities, such as $p(x_a = 1)$, humans must employ some kind of neural inference algorithm. Mathematically, $p(x_a)$ is a marginal probability that can be obtained by summing over all other variables involved in the model, i.e.,

$$p(x_a) = \sum_{x_b, x_c} p(x_a, x_b, x_c).$$

Here, $p(x_a, x_b, x_c)$ is the joint probability for a configuration of $x_a, x_b$ and $x_c$. For large models, this summation is very time-consuming; it seems unlikely that noisy neural circuits precisely execute such laborious tasks. In recent years, belief propagation (BeP) [109] has become a popular means for doing efficient inference in complex models. This algorithm is based on the local exchange of messages and local computations in networks of distributed elements, such as BBNs. BeP for a BBN can be translated into BeP for a Markov random field (MRF), i.e., BBN and MRF are two semantically equivalent graphical models [9] for which two equivalent BeP variants exist. MRF, in turn, can be brought into the form of a spin model (or equivalently: a Boltzmann machine [102]). Spin models are often considered as abstract models of neural networks with symmetric synaptic weights (Hopfield models [26], see also section 1.3.1). Hence, there is a striking resemblance of BeP to the signal transduction in models of neural networks. This observation naturally raises the following
question: Are there any models of neural networks that can be interpreted strictly in terms of BeP? If we find such a model, we will argue that BeP provides a novel interpretation of neural activity at the level of abstraction and validity of this model. In other words, the activity in neural networks should be interpreted as a process of probabilistic reasoning. It turns out that this process is non-symbolic (see section 1.2.1). According to these ideas, we work out the following points:

- On the basis of the dog example, we sketch how a BBN can be translated into a MRF.
- We then introduce BeP for MRF.
- As a central result, we show that BeP on binary pairwise MRF can be reformulated in terms of a continuous Hopfield network.
- We then reason about the applicability of the model with respect to real neural networks.

### 2.1.1 Markov Random Fields

Generally, a BBN can be converted into an equivalent MRF. We demonstrate this conversion by means of the simple BBN for the dog example used above (Fig. 2.2 a)). Starting from the BBN model, we can express the joint probability \( p(\{x\}) \) for a configuration of the binary variables \( x_a, x_b \) and \( x_c \) in the factorised form

\[
p(\{x\}) = p(x_a)p(x_c)p(x_b|x_a, x_c).
\]

This translates readily into a factor graph model (Fig. 2.2 b)), where a circle indicates a variable and the squares indicate factors. That is,

\[
p(\{x\}) = f_1(x_a)f_2(x_c)f_3(x_a, x_b, x_c),
\]

where \( f_1(x_a) = p(x_a) \), \( f_2(x_c) = p(x_c) \) and \( f_3(x_a, x_b, x_c) = p(x_b|x_a, x_c) \). A factor graph is a convenient intermediate step for a further conversion into a MRF. Generally, a MRF is an undirected graphical model that has a set of nodes, each of which corresponds to a variable (in our case \( x_a, x_b \), and \( x_c \)), as well as a set of links that connect pairs of nodes. Assuming pairwise dependencies, we can associate a function \( \psi_{ij}(x_i, x_j) \) with each link between two linked variables \( x_i \) and \( x_j \). Additionally, to each node some function \( \phi_i(x_i) \) is attached, expressing the evidence for \( x_i \). In our case, we need the functions \( \phi_a(x_a) = p(x_a) \) and \( \phi_c(x_c) = p(x_c) \) so that the evidences are equivalent to the prior probabilities. Usually, \( \phi_i(x_i) \) is supposed to be related to an additional link to a visible node (node with a fixed value \( y_i \)), for reasons that will become obvious in the example further below. The joint probability \( p(\{x\}) \) for a configuration of the binary variables \( x_a, x_b \) and \( x_c \) is the product of the functions \( \phi \) and \( \psi \), normalised by \( Z \), i.e.,

\[
p(\{x\}) = \frac{1}{Z}\psi_{ab}(x_a, x_b)\psi_{ac}(x_a, x_c)\psi_{bc}(x_b, x_c)\phi_a(x_a)\phi_c(x_c).
\]

An important alternative question is how BBNs can be implemented at the neural level. Please note the difference of the perspective compared to our question.
CHAPTER 2. BELIEF PROPAGATION BEP

Figure 2.2: The conversion of the BBN for the 'dog problem' a) into a factor graph b) and a MRF c). In the MRF, the filled circles indicate visible nodes and the empty circles indicate variable nodes. With each connection, a function \( \phi \) or \( \psi \), respectively, is associated.

This MRF is visualised in Fig. 2.2 c). A general setting would not be restricted to pairwise dependencies, but would allow for general clique potentials \( \psi(x_a, x_b, x_c) \) – which would in fact be more appropriate in the case of the dog example – but the conversion from a general BBN to a MRF is along the same lines [9]. We thus conclude that models for a problem of probabilistic reasoning can be represented as MRF, and vice versa - any MRF encodes some model of probabilistic reasoning. We now introduce BeP for MRF.

2.1.2 Belief Propagation for Binary Pairwise Markov Random Fields

In order to present BeP for more general MRF settings than the one from Fig. 2.2 c), we consider a problem from the field of computer vision [23]. Here, from a given set of observations \( \{y_i\} \), we typically want to infer some hidden quantities \( \{x_i\} \). We only consider the binary case, so that \( x_i \) can take on one of the two values \( \{-1, 1\} \). For instance, the pixel values of a grey-scaled image may be represented by \( \{y_i\} \), whereas a particular variable \( x_i \) describes whether pixel \( i \) belongs to an object \( (x_i = 1) \) or to the background \( (x_i = -1) \).

The natural question that emerges in this context is: **Given the observations \( \{y_i\} \), what is the probability for \( x_i = 1 \)?** The relation between \( \{y_i\} \) and \( \{x_i\} \) is usually given by a joint probability, written in the factorised form [109]

\[
p(\{x_i\}, \{y_i\}) = p(\{x_i\}) = \frac{1}{Z} \prod_{(i,j)} \psi_{ij}(x_i, x_j) \prod_i \phi_i(x_i, y_i),
\]

(2.5)

where the functions \( \{\psi_{ij}\} \) describe the pairwise dependencies of the hidden variables \( \{x_i\} \) and the functions \( \{\phi_i\} \) give the evidences from \( \{y_i\} \). \( Z \) is the normalisation constant. Obviously, (2.5) is the generalised form of (2.4). It can be reformulated as an Ising system...
2.1. PROBABILISTIC REASONING AND NEURODYNAMICS

with the energy
\[ E(s) = - \sum_{i,j} J_{ij}(s_i, s_j) - \sum_i h_i(s_i), \tag{2.6} \]
where the Boltzmann distribution provides the probability \( p(s) \) of a spin configuration \( s \),
\[ p(s) = \frac{1}{Z} e^{-E(s)/T}. \tag{2.7} \]
A comparison with (2.5) yields
\[ s_i = x_i, \quad J_{ij}(s_i, s_j)/T = \ln \psi_{ij}(x_i, x_j) \quad \text{and} \quad h_i(s_i)/T = \ln \phi(x_i, y_i). \]
In many cases, it is reasonable to assume that \( J_{ij}(s_i, s_j) = J_{ji} s_i s_j = J_{ji} s_i s_j \) and that \( h_i(s_i) = h_i s_i \), where \( J_{ij} \) and \( h_i \) are real-valued constants, so that (2.6) transforms into the familiar two-dimensional Ising Hamiltonian [32]. For convenience, we set \( T = 1 \).
The inference task inherent to MRF amounts to extracting marginal probabilities
\[ p_i(x_i) = \sum_{x_k, k \neq i} p(\{x_k\}). \tag{2.8} \]
An exact evaluation of \( p_i \) according to equation (2.8) is generally very time-consuming. BeP provides us with approximated marginals within reasonable time. This approach is based on the idea that connected elements (where a connection is given by \( J_{ij} \neq 0 \)) interchange messages that contain a recommendation about what state the other elements should be in [109]. Given the set of messages \( \{m_{i \rightarrow j} (x_j)\} \) at time \( t \), the messages at time \( t + 1 \) are determined by
\[ m_{i \rightarrow j}^{t+1}(x_j) = \sum_{x_i} \phi_i(x_i, y_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}^t(x_i). \tag{2.9} \]
Here, \( m_{i \rightarrow j} \) denotes the message sent from the hidden variable (or node) \( i \) to node \( j \). \( N(i) \setminus j \) denotes the set of all neighbouring nodes of \( i \) without \( j \). Usually, the messages are normalised at every time step, i.e., \( m_{i \rightarrow j}^t(1) + m_{i \rightarrow j}^t(-1) = 1 \). After (2.9) has converged, the marginals \( p_i \) are approximated by the so called beliefs \( b_i \) that are calculated according to
\[ b_i(x_i) = k \phi_i(x_i, y_i) \prod_{j \in N(i)} m_{j \rightarrow i}^t(x_j), \tag{2.10} \]
where \( k \) is a normalisation constant. In particular in connection with Ising systems, one is primarily interested in the quantity \( m_i = b_i(1) - b_i(-1) \), the so-called local magnetisation.
We will come back to a more detailed description of BeP in section 2.2.1.

2.1.3 Belief Propagation and the Neurodynamics of Hopfield Networks
We now demonstrate that BeP on binary MRF can be reformulated as a continuous Hopfield network along the lines of the following correspondence: neurons correspond to the hidden nodes of a MRF, the messages are related to the neural signal transmission and the message updates correspond to the local computation performed by single neurons. More precisely, the equations of a continuous Hopfield network follow from the equations of BeP on a binary
MRF, if there are many, but weak, connections per neuron. As a central result in this case, attractive fixed points of the Hopfield network provide very good approximations of BeP fixed points of the corresponding MRF. Due to the highly recurrent nature of biological neural networks, MRF obtained in correspondence to a neural network are naturally very ‘loopy’. Convergence of BeP on loopy structures is a delicate matter [53, 109]. Luckily, in the Hopfield case a Lyapunov function guarantees the convergence towards these fixed points. As a consequence, Hopfield networks implement BeP with guaranteed convergence. Furthermore, the result of the inference is directly represented by the activity of the neurons in the steady state.

In mathematical terms, the goal of this section is to establish a relationship between the update rules (2.9) and the dynamical equations of a continuous Hopfield network,

\[
\frac{dv_i(t)}{dt} = -v_i(t) + f \left( \sum_k w_{ki} v_k(t) \right) + K_i(t). \tag{2.11}
\]

Here, \(v_i\) is some quantity describing the activity of neuron \(i\) (e.g., the membrane potential) and \(f(x)\) is the activation function, typically implemented in a sigmoid form, such as \(f(x) = \tanh(x)\). \(w_{ij} = w_{ji}\) are the connection (synaptic) weights which need to be symmetric in the Hopfield model (see, e.g., [26] for a general introduction to Hopfield networks). The connectivity might be all-to-all or sparse. \(K_i(t)\) is an external signal or bias. According to the sketched picture, each neuron represents a variable node \(x_i\), whereas the messages are encoded in the variables \(v_i\) and \(w_{ij}\). The exact nature of this encoding will be worked out below. The Hopfield architecture implements the point attractor paradigm, i.e., by means of the dynamics the network is driven into a fixed point. At the fixed point, the beliefs \(b_i\) can be read out. In the MRF picture, this corresponds to (2.9) and (2.10). We will now realise the translation from MRF into Hopfield networks as follows:

1. Reduction of the number of messages per connection from \(m_{i \rightarrow j}(1)\) and \(m_{i \rightarrow j}(-1)\) to one reparametrised variable \(n_{ij}\).

2. Translation into a continuous system.

3. Translation of the obtained equations into the equations of a Hopfield network, where we find the encoding of the variables \(n_{ij}\) in terms of \(v_i\) and \(w_{ij}\).

This will establish the exact relationship between Hopfield and BeP.

**Reparametrisation of the messages**

In the case of binary variables \(\{x_i\}\), the messages \(m_{i \rightarrow j}(x_j)\) can be reparameterised [53] according to

\[
\tanh n_{ij} = m_{i \rightarrow j}(x_j = 1) - m_{i \rightarrow j}(x_j = -1). \tag{2.12}
\]

By this, the update rules (2.9) transform into update rules for the new ‘messages’ \(n_{ij}\)

\[
f(n) : n_{ij}^{t+1} = \tanh^{-1} \left[ \tanh(J_{ij}) \tanh \left( \sum_{k \in N(i) \setminus j} n_{ki}^t + h_i \right) \right]. \tag{2.13}
\]
For each connection $i \rightarrow j$ we obtain one single message $n_{ij}$. We can now directly calculate the local magnetisation according to $m_i = \tanh(\sum_{k \in N_i} n_{ki} + h_i)$ [54]. The Jacobian of (2.13) for $n$ is denoted by

$$df(n) = \left( \frac{\partial n_{ij}^{t+1}}{\partial n_{kl}^t} \right)_{ij}.$$  (2.14)

The used reparametrisation translates the update rules into an additive form (‘log domain’) which is a basic assumption of most models of neural networks.

Translation into a time-continuous system

(2.13) can be translated into the equivalent time-continuous system

$$\frac{dn_{ij}(t)}{dt} = g_{ij}(n(t)) = -n_{ij}(t) + \tanh^{-1} \left( \tanh(J_{ij}) \tanh \left( \sum_{k \in N(i) \setminus j} n_{ki}(t) + h_i(t) \right) \right),$$  (2.15)

where $h_i(t) = h_i$ is time-independent. The corresponding Jacobian in a point $n$ is denoted by $dg(n) = -Id + df(n)$, where $Id$ is the $|n|$-dimensional identity matrix ($|n|$ is the number of messages $n_{ij}$). Obviously, (2.13) and (2.15) have the same fixed points $n_{fp}$ which are given by

$$n_{ij} = \tanh^{-1} \left( \tanh(J_{ij}) \tanh \left( \sum_{k \in N(i) \setminus j} n_{ki} + h_i \right) \right),$$  (2.16)

with identical stability properties in both frameworks: for stability of (2.13) it is required that the real part of the largest eigenvalue of the Jacobian $df(n_{fp})$ be smaller than 1, whereas for the stability of (2.15) the condition is that the real part of the largest eigenvalue of $dg(n_{fp}) = -Id + df(n_{fp})$ must be smaller than 0. It is obvious that both conditions are identically satisfied.

Translation into a Hopfield network

The comparison between equation (2.11) and equation (2.15) does not lead to a direct identification of $v_i$ with $n_{ij}$. Rather, under certain conditions, we can identify $n_{ij}$ with $w_{ij}v_i$. That is, a message corresponds to the presynaptic neural activity weighted by the synaptic strength. Formally, we may define a variable $v_j^l$ by $n_{ij} = w_{ij}v_j^l$ and rewrite (2.15) as

$$\frac{d}{dt}w_{ij}v_j^l = -w_{ij}v_j^l + \tanh^{-1} \left[ w_{ij} \tanh \left( \sum_{k \in N(i)} w_{ki}v_k^l - w_{ji}v_j^l + h_i(t) \right) \right],$$  (2.17)

where we set $w_{ij} = \tanh(J_{ij})$.\(^2\) In the following, we assume that the synaptic weights $w_{ij}$ are relatively small, i.e., $w_{ij} \ll 1$. Hence $\tanh^{-1}(x)$ can be approximated by $\tanh^{-1}(x) \approx x$.

\(^2\)Hence the synaptic weights $w_{ij}$ are automatically restricted to the interval $[-1, 1]$.
Moreover, if a neuron receives many inputs (number of connections $q_i \gg 1$), then the single contribution $w_{ji}v_i^j$ can be neglected. Thus (2.17) simplifies to

$$\frac{d}{dt}w_{ij}v_i^j = -w_{ij}v_i^j + w_{ij}v_i^j \tanh \left( \sum_{k \in \mathcal{N}(i)} w_{ki}v_k^i + h_i(t) \right). \quad (2.18)$$

Upon a division by $w_{ij}$, we arrive at the equation

$$\frac{d}{dt}v_i^j = -v_i^j + \tanh \left( \sum_{k \in \mathcal{N}(i)} w_{ki}v_k^i + h_i(t) \right) \quad (2.19)$$

which for a uniform initialisation $v_i^{k_1}(0) = v_i^{k_2}(0) = \ldots = v_i^{k_q}(0)$ for all $i$ preserves this uniformity through time, i.e., $v_i^{k_1}(t) = v_i^{k_2}(t) = \ldots = v_i^{k_q}(t)$. $q_i$ is the number of connected neighbours of the variable $v_i$ and the indices $i$ and $k$ indicate a connection between $v_i$ and $v_k$. In other words, the subset defined by $v_i^{k_1} = v_i^{k_2} = \ldots = v_i^{k_q}$ is invariant under the dynamics of (2.19) (the subset is an invariant subspace). For such an initialisation we can therefore replace for a $i$ all $v_i^j$ by a single variable $v_i$, which leads to the equation

$$\frac{dv_i}{dt} = -v_i + \tanh \left( \sum_{k \in \mathcal{N}(i)} w_{ki}v_k + h_i(t) \right). \quad (2.20)$$

Using tanh$(x + y) \approx \tanh(x) + \tanh(y)$ if $y \ll 1$, and with $y = h_i$, we end up with the postulated equation (2.11). After the convergence to an attractor fixed point, the local magnetisation is simply the activity $v_i$. This is because the fixed point and the read out equations collapse under the approximation tanh$(\sum_k w_{ki}v_k + h_i) \approx \tanh(\sum_k w_{ki}v_k) + K_i$, i.e., $v_i(t = \infty) = m_i$.

In summary, we can emulate the original BeP procedure by a continuous Hopfield network provided that (I) the single weights $w_{ij}$ and the external fields $h_i(t)$ are relatively weak, (II) that each neuron receives many inputs and (III) that the original messages have been initialised according to $v_i^{k_1}(0) = n_{ik_1}/w_{ik_1} = v_i^{k_2}(0) = n_{ik_2}/w_{ik_2} = \ldots = v_i^{k_q}(0) = n_{ik_q}/w_{ik_q}$. From a biological point of view, the first two points seem reasonable. The effect of a single synapse is typically small compared to the totality of the numerous synaptic inputs of a cell [19, 40]. In this sense, single weights are considered weak. In order to establish a firm biological correspondence, particular consideration will be required for the last point. In the next section, we show that Hopfield networks are guaranteed to converge and thus, the required initialisation can be considered a natural choice for BeP on MRF with the properties (I) and (II).

**Guaranteed convergence for Hopfield networks**

A basic Hopfield model of the form

$$\frac{dx_i(t)}{dt} = -x_i(t) + \sum_j w_{ji}f(x_j(t)) + I_i, \quad (2.21)$$
2.1. PROBABILISTIC REASONING AND NEURODYNAMICS

Figure 2.3: The magnetisation $m$ as a function of $T$ and $w$ for the symmetric ferromagnetic model. The results for the original BeP (grey stars) and for the Hopfield network (black circles) are compared.

with $f(x) = \tanh(x)$, has the same attractor structure as the model (2.11) described above (see [26] and references therein). For the former model, an explicit Lyapunov function has been constructed [30] which assures that these networks and with them the networks considered by us are globally asymptotically stable [6].

Moreover, the time-continuous model (2.11) can be translated back into a time-discrete model, yielding

$$v_i(t+1) = \tanh \left( \sum_j w_{ij} v_j(t) \right) + K_i(t).$$

(2.22)

This equation is the proper analogue of (2.13).

Results for the ferromagnetic model

In this section, we evaluate the Hopfield-based inference solution $m_i = v_i(t = \infty)$ for networks with a simple connectivity structure: we assume constant positive synaptic weights $w = w_{ij}$ (ferromagnetic couplings) and a constant number of connections per neuron $q$. We furthermore abstain from an external field and set $K_i = 0$. To realise this symmetric model, we may either think of an infinitely extended network or of a network with some spatial periodicity, e.g., a network on a torus. According to the last section, $w$ is related to $J$ in a spin model via $w = \tanh(J) = \tanh(1/T)$, where, for convenience, we reintroduced a quasi-temperature $T$ as a scaling parameter.

From (2.11), it is clear that $v_{fp} = (v_{fp}, v_{fp}, ..., v_{fp})$ is a fixed point of the system if $v_{fp} = \tanh(qwv_{fp})$. This equation has always a solution $v_{fp} = v_0 = 0$. However, the stability of $v_0$ is restricted to $T > T_{cr}^{hn}$, where the bifurcation point is given by

$$T_{cr}^{hn} = \frac{1}{\tanh^{-1}(\frac{1}{q})}.$$  

(2.23)

This follows from the critical condition $\frac{\partial \tanh(qwv)}{\partial v}_{v=v_0} = 1$. For $T < T_{cr}^{hn}$, two additional and stable fixed points $v_{\pm}$ emerge which are symmetric with respect to the origin.
CHAPTER 2. BELIEF PROPAGATION BEP

After the convergence to a stable fixed point, \( v_+ \) for \( T < T_{\text{hn}}^{\text{crit}} \) and \( v_0 \) for \( T > T_{\text{hn}}^{\text{crit}} \), the obtained magnetisation \( m = \tanh(qw_0v_0) \) equal to \( v_0 \) is shown in dependence of \( T \) in Fig. 2.3 a) (black circles), for \( q = 20 \). The critical point is found at a temperature \( T_{\text{hn}}^{\text{crit}} = 1/\tanh^{-1}(1/20) = 19.98 \).

The result is compared to the result obtained on the basis of the original BeP equations (2.9) (grey stars in Fig. 2.3 a). We see that the critical point is slightly lower in the original BeP case. This can be understood from (2.13), for which the point given by the messages \( n_0 = (0, 0, ..., 0) \) loses stability at the critical temperature

\[ T_{\text{bep}}^{\text{crit}} = 1/\tanh^{-1} \left( \frac{1}{q-1} \right) \]  

(2.24)

For the value \( q = 20 \), this yields \( T_{\text{bep}}^{\text{crit}} = 18.98 \). \( T_{\text{bep}}^{\text{crit}} \) is in fact the critical temperature for Ising grids obtained in the Bethe-Peierls approximation (for \( q = 4 \), we get \( T_{\text{bep}}^{\text{crit}} = 2.88539 \)) [32]. In this way, we casually come across the deep relationship of BeP and Bethe-Peierls, which has been established by the theorem stating that stable BeP fixed points are local minima of the Bethe free energy functional. We will come back to this point in section 2.2.1.

In the limit of small weights, i.e., large \( T \), the results for Hopfield nets and BeP must be identical. This, in fact, is certainly true for \( T > T_{\text{hn}}^{\text{crit}} \), where \( m = 0 \) in both cases. For very large weights, i.e., small \( T \), the results are also identical in the case of the ferromagnetic couplings studied here, as \( m \to 1 \). It is only around the critical values, where the two results seem to differ. A comparison of the results against the synaptic weight \( w \), however, shows an almost perfect agreement for all \( w \) (Fig. 2.3 b) ) The differences can be made arbitrarily small for larger \( q \).

Recapitulation

We outlined the general structural affinity between belief propagation on binary Markov random fields and continuous Hopfield networks. According to this analogy, synaptic weights correspond to the pairwise dependencies in the MRF and the neuronal signal transduction corresponds to the message exchange. In the limit of many synaptic connections per neuron, but comparatively small individual synaptic weights, the dynamics of the Hopfield network is an exact mirror of the BeP dynamics in its time-continuous form. To achieve the agreement, the choice of initial messages needs to be confined. From this we can conclude that Hopfield network attractors are also BeP attractors (whereas the opposite does not necessarily hold). Unlike BeP, Hopfield networks are guaranteed to converge to a fixed point. We may thus argue that Hopfield networks naturally implement useful message initialisations that prevent trapping into a cycle. As a further benefit, the local magnetisations, as the result of the inference process, are reflected in the asymptotic neural activity. The binary basis of the implementation is not necessarily a drawback, but could simply reflect the fact that many decisions have a yes-or-no character.

2.1.4 Biological Plausibility and Non-symbolic Computation

Our work so far clearly has preliminary character. The Hopfield network model is a relatively crude simplification of biological neural networks and the relevance of our results for
such real-world structures remains questionable. However, the search for a possible neural implementation of BeP or for an interpretation of neural activities in terms of BeP is appealing. A different concept has already been outlined in [81], sharing our guiding idea that the neural activity should directly be interpreted as a message passing process. Whereas our approach is a mathematically rigorous intermediate step towards more realistic models, the approach chosen in [81] tries to directly implement BeP with spiking neurons. Our focus, however, is different. The goal of our study is not to propose possible neural implementations of BeP, but to find possible interpretations of neural activity in terms of message passing ‘algorithms’ and probabilistic reasoning. The question is now: To what extent are our results applicable to biological networks?

Answers:

- The Hopfield model equations (2.11) can be derived from spiking models in the limit of many contributions to the synaptic input [26]. This assumption is consistent with the requirement of high connectivity.

- The activity variables $v_i$ of the Hopfield model express the short-term behaviour of the firing rate of the neurons. Hence, our model presupposes a rate code, rather than a spike-timing code. To what extent (and in which cerebral area) a rate code is the basis of computation is a largely open experimental question.

- The Hopfield model assumes symmetric synaptic weights. This is in accordance with the undirected graph of a MRF. For more realistic models, this assumption must be softened. In consequence, we might have to modify the original BeP concept, while we still want to stick to the message passing idea. After all, there is no obvious reason why the brain should implement exactly the BeP algorithm. It rather seems plausible that the brain employs inference algorithms that might be conceptually close to BeP.

- Learning issues are not addressed above. There are important questions that must be addressed in the future.

The Hopfield model (2.11) is given in the form of a system of ordinary differential equations. In section 1.2.1, we argued that such systems perform non-symbolic computation. In a more realistic scenario, noise effects are also included in the model. In the most simple case, noise is considered in an additive manner, yielding the equation

$$\frac{dv_i(t)}{dt} = -v_i(t) + f \left( \sum_k w_{ki}v_k(t) \right) + K_i(t) + \eta_i(t), \quad (2.25)$$

where $\eta_i(t)$ is the additional noise term. The reliability of the inference process dependent on noise is an open issue to be analysed.

At this point, we leave the discussion about the connection between probabilistic reasoning and neural computation and turn to some more technical issues of BeP. It turns out that BeP is a powerful tool in connection with the superparamagnetic clustering algorithm that will be presented in chapter 3. The following section provides the necessary background.
2.2 Convergence of Belief Propagation on Ising Systems

In the last section, we learnt that BeP is a message passing algorithm for calculating marginal probabilities, either in Bayesian belief networks [69], in factor graphs [44] or in MRFs. In connection with these concepts, BeP has prominent applications in the fields of error correcting codes (e.g.,[36]) and computer vision (e.g., [79]). BeP has the advantage that it is often much faster than traditional, e.g., Monte Carlo-based methods. However, the speed benefit comes at a price: first, convergence of BeP depends on the initial conditions and cannot always be guaranteed. Second, even if convergent, BeP not necessarily provides exact results, but approximations whose quality is to be evaluated. In this section, we address these issues on the basis of a particular subclass of MRFs, namely ferromagnetic Ising spin systems [32] with local pairwise couplings. Here, the typical inference tasks comprise the calculation of statistical averages such as the local magnetisation and the pairwise spin correlations. The restriction on spin systems has two reasons:

- Predicting BeP convergence for arbitrary initial conditions can be non-trivial. Whereas on tree structures BeP terminates within a few steps (proportional to the number of connections [109]), this is not generally true for very loopy structures such as typical grids. Moreover, even if BeP convergence is assured, the speed of convergence can vary drastically in dependence on the system temperature $T$. More precisely, we will see that BeP shows a critical slowing down at bifurcation points of the BeP equations, where $T$ plays the role of the control parameter. These points coincide with the paramagnetic phase transition(s) as predicted by the Bethe-Peierls approximation.

- Our particular interest in BeP is rooted in its application to data clustering (see chapter 3). In our sequential superparamagnetic clustering algorithm [62, 65], the basic calculation task is to infer the spin correlations in (inhomogeneous) ferromagnetic Potts or Ising spin systems. This inference was originally based on Monte-Carlo methods. In order to speed up the algorithm, we replaced Monte-Carlo by BeP [66, 67]. This led to a successful application of BeP in the context of clustering, for which the initial conditions were chosen on the basis of trial and error.

To substantiate this BeP application, we now focus on analytical studies on convergence properties. We will work out that for ferromagnetic Ising systems

1. there is a set of reliable initial conditions leading to convergence to a fixed point,

2. there is a set of ‘dangerous’ initial conditions for which BeP is trapped in a cycle (period 2 orbit),

3. the speed of convergence (to a fixed point (1) or a cycle (2)) is characterised by critical slowing down.

For readability and shortness, some proofs are compressed to a general argument or a reference. Physical units are omitted.
2.2. CONVERGENCE OF BELIEF PROPAGATION ON ISING SYSTEMS

2.2.1 Believe Propagation on Ising Grids: Fundamentals

First, we recall some fundamental results about BeP on Ising systems with \( N \) spins \( s_i \in \{-1, 1\} \), described by the Hamiltonian\(^3\)

\[
H(s = (s_1, ..., s_N)) = - \sum_{(i,j)} J_{ij} s_i s_j - \sum_i h_i s_i. \tag{2.26}
\]

The first sum is over all pairs \((i, j)\) of spins, \( h_i \) is an external field and \( J_{ij} \) is the strength of the coupling between spin \( i \) and spin \( j \). The corresponding Boltzmann (Gibbs) distribution

\[
p(s) = \frac{1}{Z} \exp(-H(s)/T) \tag{2.27}
\]

for a ‘temperature’ \( T \) can be written in the factorised form (2.5), with \( \psi_{ij}(s_i, s_j) = \exp(J_{ij}s_i s_j/T) \) (we use \( s_i \) instead of \( x_i \)). (2.5) describes a probability measure on \( \{-1, 1\}^N \) of the type of a pairwise MRF with the \( N \) discrete random variables \( s_i \). We are interested in marginal probabilities \( p_i(s_i) \) (see equation (2.8)) or the pairwise marginals

\[
p_{ij}(s_i, s_j) = \sum_{s_1} ... \sum_{s_{i-1}} \sum_{s_{i+1}} ... \sum_{s_{N-1}} \sum_{s_N} p(s_1, ..., s_N). \tag{2.28}
\]

From this we can calculate the probability that two spins are aligned by determining the correlation \( G_{ij} = \sum_{s_i, s_j} p_{ij}(s_i, s_j) \delta_{s_i s_j} \), which is an important quantity for the superparamagnetic clustering algorithm.

For large Ising grids \( (N \gg 1) \), \( p_i(s_i) \) or \( p_{ij}(s_i, s_j) \) are usually not feasible due to the exponential costs for carrying out the sums. Here, BeP comes into play having the potential of providing an approximation within a reasonable time. According to the BeP message passing idea, neighbouring spins interchange messages that contain a recommendation about what state they should be in. The messages are modified simultaneously in a discrete, iterative time process, for which the update rule is given by (2.9). After convergence\(^4\) to \( \{m_{i, j}^{\infty}\} \), the marginals \( p_i(s_i) \) are approximated by the so-called beliefs \( b_i(s_i) \), calculated by

\[
b_i(s_i) = c_1 \phi_i(s_i) \prod_{k \in N(i)} m_{k \rightarrow i}^{\infty}. \tag{2.29}
\]

Similarly, we obtain the beliefs for the pairwise marginals \( p_{ij} \) by

\[
b_{ij}(s_i, s_j) = c_2 \phi_i(s_i) \phi_j(s_j) \psi_{ij}(s_i, s_j) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}^{\infty} \prod_{l \in N(j) \setminus i} m_{l \rightarrow j}^{\infty}, \tag{2.30}
\]

where \( c_1 \) and \( c_2 \) are normalisation constants. A justification for approximating marginals using BeP is found in the remarkable theorem that BeP fixed points correspond to stationary

---

\(^3\)Variants such as \(- \sum_{(i,j)} J_{ij} \delta_{s_i s_j} \) or \( \sum_{(i,j)} J_{ij}(1 - \delta_{s_i s_j}) \) are used as well. These forms are particularly convenient for the Potts spin extension.

\(^4\)A suitable termination criterion is \( \Delta m = \frac{1}{m} \sum_{i,j,s} |m_{i,j}^{t+1}(s) - m_{i,j}^{t}(s)| < \epsilon \), normalised with the number of messages \( m \). We then write \( m_{i,j}^{\infty} = m_{i,j}^{t} \) if \( \Delta m < \epsilon \).
points of the Bethe free energy functional $G_{\text{Bethe}}$. The Bethe free energy ansatz assumes that the free energy functional can be expressed as a functional of marginals $p_i$ and $p_{ij}$. The correct marginals are then found by minimising $G_{\text{Bethe}}$. Although only special structures such as treelike grids comply with this assumption, a minimum of $G_{\text{Bethe}}$ often provides good approximations to the exact marginals in other cases [56]. $G_{\text{Bethe}}$ is defined as

$$G_{\text{Bethe}}(\{b_i(s_i)\}, \{b_{ij}(s_i, s_j)\}) = U(\{b_i(s_i)\}, \{b_{ij}(s_i, s_j)\}) - TS_{\text{Bethe}}(\{b_i(s_i)\}, \{b_{ij}(s_i, s_j)\}),$$

with the exact energy term

$$U = \sum_{(i,j) s_i, s_j} b_{ij}(s_i, s_j)(-J_{ij}s_is_j) - \sum_i b_i(s_i)h_is_i,$$  

and the approximate entropy term

$$S_{\text{Bethe}} = -\sum_{(i,j) s_i, s_j} b_{ij}(s_i, s_j) \ln b_{ij}(s_i, s_j) + \sum_i (q_i - 1) b_i(s_i) \ln b_i(s_i).$$

Here, $q_i$ is the number of coupled neighbours of spin $i$. Two fundamental theorems provide the link between BeP and statistical physics:

**Theorem 2.1** A set of beliefs $\{b_i(s_i), b_{ij}(s_i, s_j)\}$ corresponds to a BeP fixed point if and only if the beliefs are stationary points of the Bethe free energy functional.

**Theorem 2.2** BeP on loop-free, i.e., treelike, grids always provides the exact marginals within a finite number of iterations. In this case, the Bethe free energy functional coincides with the exact free energy functional.

**Proof:**

i) Theorem 2.1: The proof can be found in Ref. [109].

ii) Theorem 2.2: In the case of trees, it is easy to proof (via induction in the number of edges) that BeP terminates within a time proportional to the number of edges and that the obtained marginals are exact [109]. Furthermore, we can directly see that the Bethe entropy term is the exact entropy term in the tree case since $b(s) = \prod_{(i,j) s_i, s_j} b_{ij}(s_i, s_j) / \prod_i b_i(s_i)$. Hence, the Bethe free energy functional becomes identical to the exact free energy functional $F = \sum_s H(s)b(s) - T \sum_s b(s) \ln(b(s))$. □

**Remarks:**

i) In [28], theorem 2.1 was proved to have a stronger form: stable fixed points of BeP are local minima of the Bethe free energy functional (but the converse is not necessarily true).

ii) The approximation obtained for finite systems by minimising the Bethe free energy functional with respect to the constraints $\sum_i b_i(s_i) = 1$ and $\sum b_{ij}(s_i, s_j) = b_j(s_j)$ is explicitly referred to as the finite-type Bethe-Peierls approximation. The shorter term Bethe-Peierls approximation shall explicitly denote the approximation of Ising grids in the thermodynamic limit (infinite grid size).
2.2. CONVERGENCE OF BELIEF PROPAGATION ON ISING SYSTEMS

2.2.2 Belief Propagation on Regular Periodic Grids

Theorem 2.2 does not guarantee BeP convergence in the case of loopy structures such as regular Ising grids. In fact, in section 2.2.4 we will see that for such grids BeP can get trapped in a cycle. First, however, we look for message initialisations that lead to convergence (section 2.2.3).

To simplify the analysis, we introduce three additional assumptions upon our class of systems under consideration. (I) We only take regular ferromagnetic Ising grids with constant couplings $J > 0$ into account. This restriction is justified from the perspective of clustering as – in accordance with the notion developed in [65] – clusters are understood as relatively homogeneous subsets that translate into regions of almost constant positive couplings. (II) We assume periodic boundary conditions that particularly simplify the analysis due to the ensuing translational invariance. The number of couplings per spin is then constant for all spins and is denoted by $q$. (III) We abstain from an external field, i.e., $h_i = 0$ or equivalently $\phi_i(s_i) = 1$.

2.2.3 Convergence to a Fixed Point

In this part, we show that BeP always converges for a uniform (i.e., site-independent) message initialisation according to $m_{i \to j}(1) = x$ and $m_{i \to j}(-1) = 1 - x$ (with identical $0 < x < 1$ for all messages $m_{i \to j}$) (proposition 2.1 and proposition 2.3). Moreover, we show that the convergence is robust even if the initialisation is not strictly uniform (corollary 2.1). Based on this finding, it is plausible that BeP also reliably converges for finite (and possibly slightly inhomogeneous) grids with open boundary conditions (remark on corollary 2.1). We further demonstrate that for the periodic Ising grid, BeP undergoes a bifurcation (from one stable to more stable fixed points) at a critical temperature $T_{\text{crit}}$ (proposition 2.2). We show that $T_{\text{crit}}$ coincides with the temperature $T_{\text{Bethe}}$ of the paramagnetic phase transition in the Bethe-Peierls approximation (as it is obtained in the thermodynamic limit).

In this subsection, we assume a grid size of $N = k \times k, k > 2$. The number of coupled neighbours is typically $q = 4$; however, $q$ might take on other values as well. For further considerations, it is convenient to use the message parametrisation (2.12) that gives us the BeP equations in the particularly simple form of the $\mathbb{R}^{Nq}$ map $f$ (equation (2.13)).

Remarks:

i) Convergence of $f$ is a necessary condition for the convergence of the original message update rule (2.9). We thus identify BeP with $f$ in the following.

ii) A uniform initialisation of (2.9) as introduced above corresponds to a uniform initialisation of $f$ according to $n_{ij} = u \forall i, j$ (that is, all messages $n_{ij}$ are set to the same initial value).

iii) The Jacobian $df$ is a sparse matrix given by

$$
\frac{\partial m_{ij}}{\partial n_{im}}(n) = \frac{\tanh(J/T) \left(1 - \tanh^2(\sum_{k \in N(i) \setminus j} n_{ki})\right)}{1 - \tanh^2(J/T) \tanh^2(\sum_{k \in N(i) \setminus j} n_{ki})} \delta_{im} \chi_{N(i) \setminus j}(l),
$$

(2.34)

where $\chi_{N(i) \setminus j}(l) = 1$ if $l \in N(i) \setminus j$ and 0 else.
Chapter 2. Belief Propagation (BEP)

Proposition 2.1 Given a $N = k \times k, k > 2$ Ising grid with periodic boundary conditions. Then:

i) For $T > T_{\text{crit}} = J / \text{tanh}^{-1}(q - 1)$, BeP always converges for any initialisation to the unique fixed point $n_0 = (0, ..., 0)$. More precisely, $n_0$ is a unique and stable fixed point for $T > T_{\text{crit}}$ and looses stability for $T < T_{\text{crit}}$.

ii) $T_{\text{crit}}$ coincides with the critical temperature $T_{\text{Bethe}}$ of the Bethe-Peierls approximation in the thermodynamic limit ($T_{\text{Bethe}} = 2.8854J$ for $q = 4$).

Proof:

i) In [54] it is shown that $f$ is a global contraction and thus converges to a unique fixed point if the spectral radius of $B_{ij,kl} = \text{tanh}(J/T)\delta_{ij}\chi_{N(i)}\chi_{N(j)}(k)$ is strictly smaller than 1. For periodic $k \times k, k > 2$ Ising grids, each row of the square matrix $B$ contains exactly $q - 1$ nonzero elements of value $\text{tanh}(J/T)$. Hence $v = (1, 1, ..., 1)^T \in \mathbb{R}^Nq$ is an eigenvector to the largest possible eigenvalue $(q - 1)\text{tanh}(J/T)$ which corresponds to the spectral radius $\rho(B)$ (this follows from the generalised Frobenius-Perron theorem, see Appendix). The critical condition $(q - 1)\text{tanh}(J/T) = 1$ yields $T_{\text{crit}} = J / \text{tanh}^{-1}(1/(q - 1))$ ($T_{\text{crit}} = 2.8854J$ for $q = 4$). For $T > T_{\text{crit}}$, we have $\rho(B) < 1$.

ii) The critical condition coincides with the critical condition of the Bethe-Peierls approximation [8, 32, 70] and thus $T_{\text{crit}} = T_{\text{Bethe}}$ (see also Appendix). □

Remark:
We call $T_{\text{crit}}$ a bifurcation point of $f$ (see also proposition 2.3). Part ii) of proposition 2.1 indicates a relation between this bifurcation point and the Bethe-Peierls approximation in the thermodynamic limit. In contrast, theorem 2.1 indicates a connection between BeP and the Bethe free energy functional for finite systems. In the next proposition, these connections are elaborated.

Proposition 2.2 For $N = k \times k, k > 2$ Ising grids with periodic boundary conditions the following statements hold:

i) The bifurcation point of $f$ at $T_{\text{crit}}$ coincides with the stability change of the paramagnetic minimum of the Bethe free energy functional. More precisely, for $T > T_{\text{crit}}$ the beliefs corresponding to $n_0 = (0, ..., 0)$ give a (unique) minimum of the Bethe free energy functional (paramagnetic minimum). For $T < T_{\text{crit}}$, this minimum disappears. The paramagnetic minimum corresponds to the finite-type Bethe-Peierls approximation. Hence this approximation shows a paramagnetic-ferromagnetic phase transition at $T_{\text{crit}}$.

ii) The phase transition of the finite-type Bethe-Peierls approximation coincides with the Bethe-Peierls phase transition in the thermodynamic limit.

Proof:

i) In [54] (Appendix), a general proof is given showing that the bifurcation point of $f$ coincides with the stability change of the paramagnetic minimum obtained from minimising the Bethe functional (2.31).

ii) The bifurcation point also coincides with the critical temperature in the Bethe-Peierls approximation (proposition 2.1). □

Remark:
The Bethe-Peierls approximation is an improved mean-field approach. For 2D ($J = 1$), the
predicted critical temperature for Ising grids in the thermodynamic limit is \( T_{\text{Bethe}} = 2.8854 \), which is a significant improvement to the mean field solution \( T_{\text{MF}} = 4 \) towards the exact value \( T_{\text{exact}} = 2.269 \) [32]. It is interesting to compare the ‘traditional’ way of deriving the critical point in the Bethe-Peierls approximation via the partition sum with the principle of minimising the Bethe free energy functional (see Appendix).

**Definition:** In the \( N_q \)-dimensional space on which the map \( f \) is operating, we define the set \( d := \{ \mathbf{n} = (n_1, ..., n_{N_q}) : n_k = n_l \forall k, l \} \) and call it ‘diagonal’.

**Remark:** The uniform message initialisation \( (m_{i\rightarrow j}(1) = x \text{ and } m_{i\rightarrow j}(-1) = 1 - x) \), with identical \( 0 < x < 1 \) corresponds to an initialisation of \( f \) in \( d \).

**Proposition 2.3** For the \( k \times k, k > 2 \) Ising grid with periodic boundary conditions the following statements hold:

i) The set \( d \) is invariant under the dynamics of \( f \).

ii) The 1D restriction \( g_1 := f|_d \) always converges for any \( n^0 \) (corresponding to an initialisation \( n^0 \in d \)).

iii) For \( T < T_{\text{crit}} \), \( g_1 \) has two stable fixed points \( n_{\pm} \) and one unstable fixed point \( n_0 = 0 \).

For \( q = 4 \), the stable fixed points are \( n_{\pm} = \pm \cosh^{-1} \left\{ \left( \frac{-3 + \tanh(J/T) + \tanh((q - 1)n^0)}{-4 + 4\tanh(J/T)} \right)^{1/2} \right\} \).

iv) For \( T > T_{\text{crit}} \), \( n_0 = 0 \) is the only fixed point and is stable.

**Proof:**

i) For any point \( \mathbf{n} = (n^0, ..., n^0) \in d \), the image \( f(\mathbf{n}) \) is in \( d \) as \( n_{ij}^1 = \tanh^{-1} \{ \tanh(J_{ij}/T) \tanh ((q - 1)n^0) \} \forall i, j \).

ii) \( g_1 \) is given by

\[
g_1 : n^{i+1} = \tanh^{-1} \{ \tanh(J/T) \tanh ((q - 1)n^i) \}. \tag{2.35}
\]

This is a one-dimensional monotonous and bounded function. Obviously, \( g_1 \) converges for any initialisation and for \( T \) to one of the fixed points (Fig. 2.4).

iii) For \( T < T_{\text{crit}} \), we solve the equation \( n = \tanh^{-1} \{ \tanh(J/T) \tanh ((q - 1)n) \} \). Clearly, \( n_0 = 0 \) is always a solution. For \( q = 4 \), we obtain the given additional solutions \( n_{\pm} \).

iv) This statement can be derived as a special case of proposition 2.1. It also follows directly by noticing that \( 0 < g'(x) < 1 \) for all \( |x| \) and hence \( g \) is a contraction. □

**Proposition 2.4** Consider \( f \) on a \( k \times k, k > 2 \) Ising grid with periodic boundary conditions. The eigenvector \( v_1(\mathbf{n}_d), \mathbf{n}_d \in d \) associated with the largest eigenvalue \( \lambda_1(\mathbf{n}_d) \) of the Jacobian \( df(\mathbf{n}_d) \) is unique and aligns with \( \mathbf{n}_d \).

**Corollary 2.1** For \( T < T_{\text{crit}} \), the diagonal \( d \) of the map \( f \) on a \( k \times k, k > 2 \) Ising grid with periodic boundary conditions is stable against perpendicular perturbations for \( |n| > n_{\text{crit}} \), where

\[
n_{\text{crit}}^n = \frac{1}{q-1} \tanh^{-1} \left( \frac{\sqrt{\cosh(J/T)} - \sqrt{1 - (q - 1) \tanh(J/T)}}{\sqrt{1 - q + \tanh(J/T)}} \right). \]
\[ g_1(n) \]

Figure 2.4: \( g_1(n) \) for a) \( J/T = 2 \) (ferromagnetic phase) and b) \( J/T = 1/4 \) (paramagnetic phase).

**Proof:** i) Proposition:

\[ df(n_d) = \frac{\tanh(J/T) (1 - \tanh^2(q'n_d))}{1 - \tanh^2(J/T) \tanh^2(q'n_d)} \]

with \( q' = q - 1 \). This is a sparse matrix with \( q' \) non-negative identical entries per row. Thus \((1,1,...,1)\) (or any vector in \( d \)) is an eigenvector. The according eigenvalue \( \lambda_1(n_d) \) is the largest possible one:

\[ \lambda_1(n_d) = g_1'(n_d) = q \frac{\tanh(J/T) (1 - \tanh^2(q'n_d))}{1 - \tanh^2(J/T) \tanh^2(q'n_d)}. \]

The eigenvector corresponding to \( \lambda_1(n_d) \) is positive and \( \lambda_1(n_d) \) is unique by virtue of the Frobenius-Perron theorem (see Appendix).

ii) Corollary:

We assume \( T < T_{crit} \). The stability of \( d \) against perpendicular perturbations \( \Delta n \) is determined by

\[ |df(n_d)\Delta n| < |\lambda_1\Delta n| = |g_1'(n_d)\Delta n|. \]

Consequently, \( d \) is certainly stable against perpendicular perturbations if \( |g_1'(n_d)| < 1 \). The critical condition \( |g_1'(n_d)| = 1 \) yields \( n_{crit}^T \).

**Remark:** The stability of the diagonal supports the numerical observations [67] that for a finite system with open boundary conditions a uniform message initialisation always leads to convergence. Here, we assume that a sufficiently large system essentially behaves like a periodic one and that the boundary effects can be modelled as (sustained) perturbations. In particular for clustering applications, the restriction on uniform initialisations guarantees reliable results.

### 2.2.4 Convergence to a Cycle

In this section, we demonstrate that for the set of alternating uniform message initialisations, BeP gets trapped in a period-2 orbit if \( T < T_{crit} \). This set is defined by messages \( m_{i\rightarrow j}(1) = y, m_{j\rightarrow k}(1) = 1 - y \), and \( m_{i\rightarrow j}(-1) = 1 - y, m_{j\rightarrow k}(-1) = y \) \((y \in [0,0.5] \) identical
2.2. CONVERGENCE OF BELIEF PROPAGATION ON ISING SYSTEMS

Figure 2.5: An alternating initialisation leads to a cycle.

for all messages) and translates into initialisations of the map $f$ according to Fig. 2.5. Small deviations from these initial conditions do not change the outcome, indicating the existence of a non-negligible basin of attraction. In practical BeP applications, the algorithm can usually be terminated if the difference between successive messages $\{m_{i\rightarrow j}^t\}$ and $\{m_{i\rightarrow j}^{t+1}\}$ becomes small. Hence alternating initialisations must be absolutely avoided as they can make a termination impossible. Throughout the section, we assume $q = 4$.

Definition: For a $N = (2^k) \times (2^k), k > 1$ periodic Ising grid we define the following sets for the map $f$:

\[
d_+ := \{n | n_{ij} = -n_{ji}, n_{12} \geq 0\} \quad \text{and} \quad d_- := \{n | n_{ij} = -n_{ji}, n_{12} \leq 0\}.
\]

We call the union $d_c := d_+ \cup d_-$ ‘cross diagonal’.

Remarks:

i) The condition $N = (2^k) \times (2^k), k > 1$ is necessary to make sure that the definition is consistent with the periodic boundary conditions. In particular, we have the requirement $n_{ij} = (-1)^{(l+1)}n_{ji}$, where $l = 2k$ is the ‘length’ of the grid. The grid extension is $N = (2k) \times (2k), k > 1$ throughout this section.

ii) The proposed alternating message initialisation corresponds to an initialisation of $f$ in $d_c$.

Proposition 2.5 For a $N = (2^k) \times (2^k), k > 1$ periodic Ising grid, the set $d_c = d_+ \cup d_-$ is invariant under $f$.

Proof: We first note that $n_{ij} = -n_{ji}$ (apply the condition $n_{ij} = -n_{ji}$ over a loop) and that $|n_{ij}| = \text{const} \ \forall \ i,j$. Assume now (w.l.o.g.) $n^0 \in d_+$, i.e., $n^0 = (n_+ = n^0, n_- = -n^0, ...)$.

Due to the definition, the image of $n^0$ must be in $d_-$.
Proof: The proofs are essentially analogous to the proofs for proposition 2.4 and corollary 2.1. In particular, we use that

\[ df_{ij} \]

\[ n_{ij} \]

\[ f \]

\[ g \]

\[ n_{ij}^{t+1} = \tanh^{-1}(\tanh(J/T) \tanh((q-1)n_{-})) \]

\[ n_{ij}^{t+1} = -\tanh^{-1}(\tanh(J/T) \tanh((q-1)n_{+})) \]

\[ n_{ij}^{t+1} = -\tanh^{-1}(\tanh(J/T) \tanh((q-1)n_{-})) \]

Thus, the map flips the sign of all entries, whereas the absolute values are changed uniformly. Consequently, the condition \( n_{ij} = -n_{ji} \) is met again, but \( n_{12} < 0 \). Hence, \( n^1 \in d_- \). The analogous argument holds for \( n^0 \in d_- \). □

Definition: We define the one-dimensional map \( g_2(n) \) as the restriction of \( f \) to \( d_c \), i.e.,

\[ g_2 := f_{d_c} \]

Proposition 2.6 For a \( N = (2k) \times (2k), k > 1 \) periodic Ising grid, the following statements hold:

i) \( g_2 \) has one unique fixed point \( n_0 = 0 \) corresponding to \( n_0 = 0 \) which is stable for \( T > T_{crit} \).

ii) The one-dimensional map \( g_2 \) is equal to \( -g_1 \) and hence \( g_2 \circ g_2 = g_1 \circ g_1 \), where \( f \circ g \) denotes \( f(g(x)) \).

iii) For \( T < T_{crit} \), \( g_2 \) has a stable orbit of period 2.

Proof:

i) \( n_0 = 0 \) is clearly a fixed point of \( g_2 \) which can be checked by inserting in (2.39) and (2.40). Its uniqueness and stability for \( T > T_{crit} \) follows from proposition 2.1 (see also proposition 2.3).

ii) The statement \( g_2 = -g_1 \) immediately follows from (2.39), (2.40) and (2.35). Furthermore, from \( g_1(-n) = -g_1(n) \) it follows that \( g_2 \circ g_2 = g_2(g_2(n)) = g_1(g_1(n)) = g_1 \circ g_1 \).

iii) As \( g_2 \circ g_2 = g_1 \circ g_1 \) and \( n_{\pm} \) are stable fixed points of \( g_1 \) (and thus of \( g_1 \circ g_1 \)), there is a stable period-2 orbit \( (n_{\pm}^t, n_{\pm}^{t+1}) \). □

Proposition 2.7 Consider \( f \) on a \( (2k) \times (2k), k > 1 \) periodic Ising grid. For points \( n_{d_c} \in d_c \) of the cross diagonal, the eigenvector \( v_1(n_{d_c}) \) associated with the largest eigenvalue \( \lambda_1(n_{d_c}) \) of the Jacobian \( df(n_{d_c}) \) is unique and aligns with vectors \( n_{d_c} \) of the diagonal \( d \).

Corollary 2.2 For \( T < T_{crit} \), the cross diagonal \( d_c \) of the map \( f \) on a \( (2k) \times (2k), k > 1 \) Ising grid with periodic boundary conditions is stable against perpendicular perturbations for \( n > n_{T_{crit}} \), where again

\[ n_{T_{crit}} = \frac{1}{q-1} \tanh^{-1} \left( \frac{\sqrt{\coth(J/T)\sqrt{1-(q-1)\tanh(J/T)}}}{\sqrt{1-q\tanh(J/T)}} \right) \]

Proof: The proofs are essentially analogous to the proofs for proposition 2.4 and corollary 2.1. In particular, we use that \( df(n_{d_c}) = df(n_d) \), where \( |n_d| = |n_{d_c}| \). □
2.2. CONVERGENCE OF BELIEF PROPAGATION ON ISING SYSTEMS

2.2.5 Critical Slowing Down

In this section, we discuss the phenomenon of critical slowing down, which is observed when $T$ approaches $T_{crit}$. We again consider periodic Ising grids and assume that the messages are either initialised in $d$ or in $d_c$. We can thus restrict the map $f$ to the one-dimensional map $g_1$ (the results extend to $g_2 = -g_1$). These restrictions are only introduced for the purpose of a straightforward analysis. The phenomenon of critical slowing down is, however, generic and is also found for finite inhomogeneous systems with open boundary conditions (see Fig. 2.6).

To understand the phenomenon, we recall that $g_1$ shows a bifurcation point at $T_{crit}$ that coincides with the phase transition $T_{Bethe}$ predicted by the Bethe-Peierls approximation. Close to a stable fixed point $n^\infty (n_0$ for $T \geq T_{crit}$ and $n_{\pm}$ for $T \leq T_{crit})$ a perturbation $\epsilon_0 = |n^0 - n^\infty|$ (where $n^0$ is the initial value) vanishes as $\epsilon_l = |g'_1(n^0) - n^\infty|$ or

$$\epsilon_l \approx (g'_1(n^\infty))^l \epsilon_0. \quad (2.41)$$

From this equation, we estimate the number of iterations $l$ to fall below an arbitrary small threshold $\epsilon_c$ as $l > \ln(\epsilon_c/\epsilon_0)/\ln(g'_1(n^\infty))$, from which it follows that

$$l \sim \frac{1}{\ln(g'_1(n^\infty))}. \quad (2.42)$$

Since $g'_1(n^\infty) \to 1$ for $T \to T_{crit}$, we see that $l$ peaks at the critical point. The shape of the numerically calculated peak in fact is very well predicted by (2.42) (Fig. 2.6). However, the singularity of (2.42) is an artifact of this analytical approach and cannot be encountered in numerical simulations as they always terminate for a finite $\epsilon_c$. Nonetheless, we may define a BeP critical exponent $\gamma$ by expanding $r(T) := \ln(g(n^\infty(T)))$ around $T_{crit}$ and noticing that
the leading order is 1, as \( r'(T_{\text{crit}}) \neq 0 \). We may thus write

\[
l \sim \frac{1}{|T - T_{\text{crit}}|^{\gamma}} \quad \text{with} \quad \gamma = 1.
\]

(2.43)

**Remarks:** Critical slowing down may seem to be a serious problem for clustering applications. Fortunately, the sequential superparamagnetic clustering algorithm is very robust in two respects, so that critical slowing down does not affect the performance. First, even at the critical temperature the fixed point can quickly be approached to within a certain \( \epsilon \)-neighbourhood. Further convergence is difficult due to the critical slowing down. However, the \( \epsilon \)-approximation after about 10 to 20 iterations is already good enough and suffices for robust clustering results. Second, the sequential procedure inherent to our algorithm does not rely on very exact results, but can deal with relatively crude approximations [67].

![Figure 2.7: Stability of the diagonal \( d \) and the cross-diagonal \( d_c \): Within the inner circle with radius \( n_{T_{\text{crit}}}^d \), small perturbations perpendicular to one of the diagonals diverge, whereas outside they vanish, i.e., the diagonals are stable. The outer circle with radius \( |n_+| = |n_-| \) gives the fixed point of first (on \( d \)) and second (on \( d_c \)) order.](image)

2.2.6 Recapitulation and Conclusions

In this section, we have examined the BeP convergence problem for periodic ferromagnetic Ising grids. We dealt with the following questions: 1) For what message initialisation does BeP certainly converge to a fixed point? 2) for what message initialisation does BeP certainly not converge to a fixed point? As a result, we have elaborated two simple and natural sets of initial conditions that certainly lead to convergence either to a fixed point or to a cycle of period 2 (see Fig. 2.7). Whereas fixed point convergence is necessarily desired
2.2. CONVERGENCE OF BELIEF PROPAGATION ON ISING SYSTEMS

Figure 2.8: In some regions of the parameter space of an periodic Ising model (where $T$ is the temperature and $\lambda$ describes some path in the state space) the boundary between the regions of convergence (bright) and non-convergence (dark) is unclear. When zooming in (small black square), a self-similar structure is revealed, resembling fractal basin boundaries.

for BeP in connection with inference problems, convergence to cycles, or non-convergence in general, must be avoided. For the first set of initialisations, i.e., the set of uniform (site-independent) message initialisations, we showed that fixed point convergence is always guaranteed. But even in this case, the speed of convergence is affected by critical slowing down: at a critical temperature that coincides with the phase transition in the Bethe-Peierls approximation, BeP undergoes a bifurcation that is accompanied by a characteristic slowdown of convergence.

From the viewpoint of application, we are mainly interested in the performance of BeP in connection with the sequential superparamagnetic clustering algorithm. Hence, we question to what extend the obtained results can help to assure and understand a fast BeP convergence for this particular application. We summarise the main points: (I) The typical graph structure underlying superparamagnetic clustering can be roughly sketched by local Ising grids that interact weakly. It is thus plausible that site-independent message initialisations lead to a reliable convergence. This observation has been substantiated by numerous successful simulations. (II) Critical slowing down of BeP is also observed in the
case of superparamagnetic clustering. The performance of the algorithm, however, is not affected if the number of iterations is limited to about $10 - 20$. Thus critical slowing down can be neglected. (III) The quality of the approximations obtained by BeP is thoroughly good enough to yield identical or almost identical results as Monte-Carlo methods within the context of sequential superparamagnetic clustering [67].

Despite their elementariness, periodic ferromagnetic Ising grids exhibit a surprising complexity if we try to judge whether BeP converges for a given arbitrary initialisation in the ferromagnetic phase. There is even numerical evidence that regions of convergence and non-convergence are separated by fractal basin boundaries (see Fig. 2.8, more details in [91]). The mathematical details behind this observation remain to be worked out in the future. We now address ourselves to the field of clustering algorithms, where we can make use of the gained knowledge about BeP.
Chapter 3

Clustering Algorithms

Qualitatively, the task of a clustering algorithm can be characterised as *grouping together items that – in a situation-dependent sense – belong together*. Clustering algorithms have gained increased importance in all domains of data analysis and data processing, most prominently: bioinformatics [110], chemoinformatics [62], neuroscience [74], visual scene analysis [22, 79], or the design of autonomous systems [100].

The problem of clustering is actually ill-posed, as the notion of a cluster is inherently diffuse. A set of geometric figures can be clustered according to the shape or to the colour, of the figures, to the combined features, or to any other feature. Whether a clustering result will be useful depends on whether the clustering is suitable for the particular focus taken. With multiple competing attributes, a data item is therefore most naturally described by a likelihood of belonging to different clusters (fuzzy clustering). Furthermore, clusters inherently emerge nested in hierarchies, of which only a substructure may be of interest. In practice, however, many decisions relying on clustering require a clear assignment of each item to one cluster or category.

In section 1.1.2, we noticed that common clustering algorithms are typically off-line procedures that, in particular, do not automatically incorporate mechanisms for improvements due to performance feedback. Nevertheless, clustering results can be improved by an additional external cluster validation. This typically assumes that the whole clustering process consists of two processing stages. In the first stage, the relevant features or attributes that characterise the items to be clustered are identified and the similarity measure that quantifies the pairwise affinities of items, given the chosen features, is specified. This stage introduces a priori information into the clustering so that at this point the possible clustering solutions are already predetermined. Consequently, acquired or learnt prior information is implemented in the form of modified attributes or an optimised similarity measure. Unfortunately, the choice of appropriate validation criteria is highly nontrivial [6]. In the example given above, the validation may force the weight to shift from shape to colour, or to a new attribute, such as size. The clustering algorithm itself, which is the second stage of processing, is supposed to work in a completely unbiased way, as in most situations, no a priori information about the number of clusters, their shapes, or their internal structures (as the most relevant aspects), is available. If the clustering fails to detect satisfactory solutions, the first stage of processing needs to be improved, while the properties of the actual clustering process...
remain unchanged.

A clustering algorithm’s challenge is to bring out ‘the best in possibly ambiguous data’. The quality of a clustering result is ill-defined. There is no reasonable chance for a validation with a unique and absolute measure. Most clustering algorithms will yield optimal results under specific conditions or when assessed by means of an appropriate algorithm-specific validation measure. Nonetheless, some basic requirements for a reasonable clustering method can be listed as follows:

a) The method should provide a unique clustering hierarchy and identify the most ‘natural’ clusters. Preferentially, some measure should specify the naturalness of a cluster. According to our unbiased approach, the most practical notion of a natural cluster is a group without any significant substructure (as quantified by the measure).
b) The method should not assume any a priori information about number, shape or internal distribution of clusters.
c) Preferentially, the method should be based on the set of pairwise affinities. This allows for optimising the results by optimising the similarity measure.
d) The method should be able to deal with clusters of different shapes, densities and largely unequal distances between clusters.

In the next section, we will review some of the most popular and interesting algorithms.

3.1 A very Short Survey of Clustering Algorithms

There is a plethora of different clustering algorithms (for overviews see e.g., [11, 20, 35, 108, 110]). A systematic classification of clustering algorithms is not straightforward as – depending on the criteria chosen – some algorithms can be assigned to different classes (this is the clustering problem of the clustering algorithms). A common denominator of all algorithms is their goal of partitioning a given set of data items into subgroups such that the similarity between items of a subgroup is larger than the similarity between items belonging to different subgroups [108]. Most algorithms can be classified along the rough criteria hierarchical/nonhierarchical and fuzzy/hard (nonfuzzy). Some methods, however, do not follow these simple discrimination criteria. For example, for central clustering, the fuzzy vs. nonfuzzy character can be emphasised by fixing the number of clusters, but also the hierarchical aspects can be emphasised by changing the Lagrange parameter $T$ (temperature) and introducing hard assignments. The hierarchical and the fuzzy clustering approaches emphasise the ambiguous character of clustering problems. In practise, however, many applications require clear-cut solutions which are emphasised by nonhierarchical and nonfuzzy clustering approaches. In table 3.1, a number of important methods are listed. This list is by far not exhausting, and many other methods are discussed in the afore-cited review articles.

Many of these methods are in their very core similar, although they may follow from different views on the clustering problem. Consequently, different names show up for largely equivalent methods. For instance, fuzzy c-means and soft k-means start from the idea of achieving fuzzy cluster assignments by minimising only slightly different objective function(al)s that even completely coincide in the limit of hard assignments. Yet, the strategy
3.1. A VERY SHORT SURVEY OF CLUSTERING ALGORITHMS

<table>
<thead>
<tr>
<th>Method</th>
<th>hierarchical?</th>
<th>fuzzy?</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-means [108]</td>
<td>nonhierarchical</td>
<td>nonfuzzy</td>
</tr>
<tr>
<td>Soft k-means [46]</td>
<td>nonhierarchical</td>
<td>fuzzy</td>
</tr>
<tr>
<td>Fuzzy c-means [18]</td>
<td>nonhierarchical</td>
<td>fuzzy</td>
</tr>
<tr>
<td>Self-organised maps [42]</td>
<td>nonhierarchical</td>
<td>nonfuzzy</td>
</tr>
<tr>
<td>Linkage methods (single/complete/Ward)</td>
<td>hierarchical</td>
<td>nonfuzzy</td>
</tr>
<tr>
<td>Information bottleneck method clustering</td>
<td>hierarchical aspects</td>
<td>fuzzy</td>
</tr>
<tr>
<td>Central clustering by deterministic annealing [75]</td>
<td>hierarchical aspects</td>
<td>fuzzy</td>
</tr>
<tr>
<td>Normalised cut [80]</td>
<td>nonhierarchical</td>
<td>nonfuzzy</td>
</tr>
<tr>
<td>Pairwise clustering [29]</td>
<td>hierarchical aspects</td>
<td>fuzzy</td>
</tr>
<tr>
<td>Superparamagnetic clustering [10]</td>
<td>hierarchical aspects</td>
<td>nonfuzzy</td>
</tr>
</tbody>
</table>

Table 3.1: Table of different clustering algorithms.

For minimisation is different. Furthermore, k-means is a special case of soft k-means, which again can be derived from the principles of the information bottleneck method ([90], see also section 1.1.3). Central clustering is ultimately a generalised scheme for soft k-means (see below). The problem of central clustering can be solved by deterministic annealing, which again is a minimisation heuristic that can also be used in the context of pairwise clustering [29]. Normalised cut (NC), in turn, is a graph theoretic approach for which the data setting is first translated into an undirected weighted graph. NC then tries to find an optimal partition of the graph. Having this picture in mind, superparamagnetic clustering has been reinterpreted as an average cut algorithm [79].

From an overall perspective, we can distinguish between two important clustering schools. One school centres the idea of objective functions, i.e., the best clustering is the one that minimises a predefined cost function (or functional). The other school emphasises the role of local rules which are often in the guise of unsupervised learning or self-organisation processes within a network. Many methods can be treated from both perspectives. For instance, the goal of the k-means algorithm is to minimise the cost function

\[ H_{km} = \sum_{j=1}^{k} \sum_{i=1}^{n} r_{ij} d(x_i, c_j), \]  

where \( x_i \) denotes the \( i \)th of \( n \) data vectors, \( c_j \) denotes the \( j \)th of \( k \) central vectors and \( d(x_i, c_j) := (x_i - c_j)^2 \) is the square of the distance or dissimilarity between \( x_i \) and \( c_j \). \( r_{ij} \) denotes the cluster assignment, where \( r_{ij} = 1 \) if \( j = \text{argmin}_l d(x_i, c_l) \) and \( r_{ij} = 0 \) otherwise.

Usually, k-means clustering is introduced in the form of the following algorithmic scheme:

1. Set the \( k \) centre vectors \( c_j \) to random values.
2. Determine for each data vector \( x_i \) the closest centre \( c_j \) and assign the label \( j \) to \( x_i \) \((r_{ij} = 1)\). All vectors with the same label form a cluster.
3. Calculate the mean of each cluster and identify it with \( c_j \).
4. Go back to 2 as long as the assignments do change.
This scheme can be interpreted as an algorithm for rewiring processes within a network [63]: We imagine a connection between each data point and its corresponding central unit with a connection strength that is given by the distance \(d(x_i, c_j)\). In this picture, the k-means algorithm results in a self-organisation of the network structure, ending in a stable configuration (see Fig. 3.1).

![Figure 3.1: The k-means algorithm can be interpreted as a rewiring process: a) initial state, b) final state (black circles denote the centers).](image)

### 3.1.1 Clustering with Cost Functions

Cost functions or objective functions are probably the most common means to attack a clustering problem. In section 1.1.3, we learnt in connection with the information bottleneck method that the usage of cost functions can be justified from the more abstract principle of optimal predictions. Following this idea, we can conclude that the general goal of clustering is to achieve a global (or at least local) minimum of a (distortion) function

\[
D(p(x, c)) = \sum_{j=1}^{k} \sum_{i=1}^{n} p(x_i, c_j) d(x_i, c_j).
\]  

(3.2)

In this formulation, \(p(x_i, c_j)\) denotes the probability that \(x_i\) is associated with the cluster center \(c_j\) (degree of cluster membership). We therefore generally deal with a fuzzy clustering approach, where \(D\) is a function(al) on the space of valid assignment probabilities. If we demand that \(p(x_i, c_j) = r_{ij}\), we get back the cost function for k-means clustering (3.1).

Instead of the vector-based function \(D\) for central clustering, also objective functions for pairwise clustering have been suggested that only require the (dis)similarity matrix of the data [29]. In general, clustering methods differ in the way they define the cost function, and in the way they try to find an optimum (we usually have some kind of gradient descent).

Obviously, the cost function \(D(p(x, c))\) is a straightforward generalisation of the k-means cost function \(H_{km}\) for fuzzy clustering. However, if a real generalisation shall be obtained, we have to impose conditions on the probability distributions, otherwise the hard (nonfuzzy) clustering assignment will always meet optimality and fuzziness is impossible. The conditions are realised by prespecifying the level of fuzziness we want to achieve. For instance, this could be done by assuming that the cluster membership probability for a vector \(x_i\) is described by
a Gaussian around the center \( c_j \) and with a fixed standard deviation \( \sigma \). \( \sigma \) specifies the level of fuzziness. The k-means clustering algorithm can then be adapted in the following form (soft k-means):

1. Set the \( k \) centre vectors \( c_j \) to random values.

2. Determine for each data vector \( x_i \) the degree of membership to a cluster with center \( c_j \) according to

\[
p(x_i, c_j) = \frac{e^{-d(x_i,c_j)/(2\sigma^2)}}{Z_i},
\]

where \( Z_i := \sum_{j=1}^{k} e^{-d(x_i,c_j)/(2\sigma^2)} \) ensures normalisation.

3. The centre vectors are then adjusted by

\[
c_{j}^{\text{new}} = \frac{\sum_{i=1}^{n} p(x_i, c_j) x_i}{\sum_{i=1}^{n} p(x_i, c_j)}
\]

4. Go back to 2 as long as the assignments do change.

By comparing the Gaussian (3.3) to the solution (1.15) found in section 1.1.3, we see that the expressions for the assignment probabilities coincide when using \( d(x_i,c_j) = |x_i - c_j|^2 \) and when establishing the correspondence \( \sigma \sim T^{1/2} \). In other words, the generalised (soft) k-means algorithm above is nothing else, but a heuristic to minimise the Lagrangian \( F = D - TH \) for a fixed fuzziness level \( T \).

**Expectation maximisation and deterministic annealing**

The soft k-means algorithm allows for gaining even deeper insight into the principles behind clustering by cost functions: interestingly, the soft k-means algorithm is actually a variant of the expectation maximisation algorithm (EM) [9]. The EM algorithm provides an iterative procedure for computing maximum likelihood estimates [26]. Consider the example of a Gaussian mixture model which, after all, is the basis of the soft or fuzzy k-means algorithm above. We may generalise the Gaussian mixture by allowing for general covariance matrices \( \Sigma_j \) and general mixing coefficients \( \pi_j \geq 0 \) with \( \sum_{j=1}^{k} \pi_j = 1 \) [9]. To be consistent with the considerations above, we assume \( \Sigma_j = \sigma_j^2 Id \) and hold \( \sigma_j \) fixed. \( Id \) denotes the identity matrix. The degree of membership to a cluster with center \( c_j \) then takes on the form

\[
p(x_i, c_j) = \frac{\pi_j N(x_i|c_j, \Sigma_j)}{\sum_{l=1}^{k} \pi_l N(x_i|c_l, \Sigma_l)},
\]

where \( N(x|c, \Sigma) \) denotes the Gaussian distribution. To find the best solution for the parameters \( c_j \) (for given \( \pi_j \) and \( \Sigma_j \)), we have to maximise the log-likelihood of the mixture density [9]

\[
L(c) = \ln f(x|\pi, c, \Sigma) = \ln \left( \prod_{i=1}^{n} \sum_{l=1}^{k} \pi_l N(x_i|c_l, \Sigma_l) \right) = \sum_{i=1}^{n} \ln \left( \sum_{l=1}^{k} \pi_l N(x_i|c_l, \Sigma_l) \right).
\]
Setting the derivative with respect to the means $c_j$ to zero, we obtain

$$0 = \sum_{i=1}^{n} p(x_i, c_j) \Sigma_j (x_i - c_j).$$

(3.7)

By multiplying with the inverse $\Sigma_j^{-1}$, equation (3.4) is obtained. Similarly, we obtain $\pi_j = \sum_{i=1}^{n} p(x_i, c_j) / n = n_j / n$. Due to the interdependence of the formulas (3.4) and (3.5), we do not have a closed form for the solution of the max-likelihood problem. At this point, the EM algorithm suggests a simple two-step iterative scheme. In the first step (the expectation step), the posterior probabilities, or degrees of cluster membership, are calculated according to (3.5). Initially, the values for $c_j$ and $\pi_j$ are chosen at random. In the second step (the maximisation step), the means are re-estimated with (3.4) and after that, the new $\pi_j$ is $\pi_j = n_j / n$. This procedure is run until convergence. It can be shown that the log-likelihood is increased with each step [9]. Furthermore, if we assume fixed $\pi_j = 1/k$ from the beginning, the EM algorithm exactly describes the soft k-means algorithm from above. We thus interpret this algorithm as a likelihood maximisation for Gaussian mixtures.

Instead of working with Gaussians, we may directly minimise the Lagrangian $F = D - TH$. The formal solution for the degrees of cluster membership at a fixed $T$ has been determined in section 1.1.3 by the Boltzmann distribution

$$p(x_i, c_j) = \frac{e^{-d(x_i, c_j)/T}}{Z_i} \quad \text{with} \quad Z_i := \sum_{j=1}^{k} e^{-d(x_i, c_j)/T}. \quad (3.8)$$

Unfortunately, the parameters $c_j$ are still unknown. As we want to minimise $F$, we can plug (3.8) back into $F$, yielding $F^*$. We then minimise $F^*$ with respect to $c_j$ which gives the conditions [75]

$$\sum_{i=1}^{n} p(x_i, c_j) \frac{d}{dc_j} d(x_i, c_j) = 0, \quad j = 1, \ldots, k. \quad (3.9)$$

The obtained solutions for $c_j$ can then be used to estimate a better $p(x_i, c_j)$. If we assume that $d(x, c) = |x - c|^2$, $p(x_i, c_j)$ becomes a Gaussian and the conditions (3.9) possess the solution (3.4). In other words, the two steps described by (3.8) and (3.9) coincide with the two iterative EM steps found in the soft k-means algorithm. However, the scheme via minimising $F$ is more general than EM, as the choice of the distortion measure is not confined to the Euclidian distance. The iterative application of (3.8) and (3.9) while slowly lowering $T$ is known as deterministic annealing. It aims at finding a global minimum of the hard clustering problem, where the annealing process shall prevent from being stuck in a local minimum.

The cost function approaches follow a top-down perspective: the rules for clustering are derived from an abstract principle, formulated as an optimisation problem. In contrast, if we give the rules for clustering, then this is identified with a bottom-up perspective: possibly, a cost function corresponding to the process can be identified, but this cost function appears as a kind of epiphenomenon. We now briefly recapitulate this alternative perspective.
3.1.2 Clustering through Self-organisation

The key idea of clustering by self-organisation is that clusters emerge due to some local interaction rules in a network. This view seems appropriate to reflect natural clustering processes (for which illustrative examples are ant-based clustering [25] or neural gases [48]). In section 1.3.1, we discussed original clustering approaches based on Hebbian learning. These approaches are not repeated here. Instead, we briefly mention the SOM algorithm (self-organising maps [42]), which has been paradigmatic for self-organised clustering. The goal of SOM is to transform an incoming high-dimensional signal pattern into a two-dimensional discrete map with a topological order. It is a feedforward network for which each input provokes the activity of one winner-neuron in a two-dimensional output layer, and similar input patterns provoke the activity of nearby neurons. In accordance with the principles of self-organisation formulated in section 1.3.1, the algorithm can be divided into three essential processes [26]: 1) a competition process, where for an input the winning neuron is determined, 2) a cooperative process, where the topological ordering is built out, and 3) an adaptive process, where the synapses are modified according to some Hebbian-like learning rule. The exact algorithm formulated along the guideline of these processes can be found in [26]. In [89], the batch-mode version of SOM is explained as a gradient descent search for a certain criterion function which allows to view SOM also from the top-down perspective. This demonstrates that top-down and bottom-up views must be understood as complementary descriptions for clustering.

3.2 Sequential Superparamagnetic Clustering SSC

In the introduction to this chapter, we listed four requirements to be fulfilled by a reasonable nonparametric clustering algorithm. While most algorithms are optimal or optimised for special situations, the goal to take all requirements into account is rarely met. In particular, little attention has been paid to requirement d). It thus comes as no surprise that many, mainly standard, algorithms have to struggle to find natural clustering solutions for inhomogeneous data distributions (see Fig. 3.4 further below) [64]. However, inhomogeneous distributions leading to highly varying intra- and inter-cluster distances are characteristic for many real-world applications. Such inhomogeneities are often introduced when insufficient care is applied during the choice or the scaling of the attributes. Even the most sophisticated similarity measures can sometimes not compensate for this neglect.

In this section, we introduce the sequential superparamagnetic clustering algorithm [62] as a solution to these problems: in addition to taking into account points a), b), c), it also offers a solution to d). The core of the algorithm is the superparamagnetic clustering approach (SC) [10], which provides ‘natural’ clustering solutions on different resolution levels. The levels are controlled by a ‘temperature’ parameter \( T \) that controls the number of clusters. The approach, however, leaves open which resolution level provides the best clustering, in the sense of a unique choice of the ‘most natural’ clusters. As was already pointed out in [10], the best clustering sometimes chooses clusters across different levels. The central question thus remains: according to which criterion should levels and clusters be chosen? The sequential procedure answers the question. Strictly speaking, it renders
the question meaningless, as the most natural clusters turn out not to be characterised by global parameters $T$.

### 3.2.1 Superparamagnetic Clustering

Superparamagnetic clustering has been described from different viewpoints inherent to different fields, such as graph theory [79] or stochastic neural networks [63]. Here, we shall use the description via Potts spin models, as put forward in the original work by E. Domany’s group (see e.g. [10, 107]). We only briefly outline the algorithm, for more details we refer to the aforementioned references and those that will be given below. For $N$ items to be clustered with pairwise affinities $d_{ij}$, an inhomogeneous grid of Potts spins is constructed in the following way: each item $i$ is represented by one site of the grid with Potts spin variable $s_i$, where $s_i \in \{1, \ldots, q\}$. $q$ is typically set to 10 or 20 [10, 62]. It is important to note that the choice of $q$ is largely arbitrary and is not related to the number of emerging clusters. Each spin is symmetrically coupled to its $k$ (not necessarily mutual) nearest neighbours. The choice of $k$ will be discussed in section 3.2.3. The coupling strength $J_{ij}$ is a decreasing function of $d_{ij}$, e.g.,

$$J_{ij} = J_{ji} = \frac{1}{\hat{K}} \exp\left(\frac{-d_{ij}^2}{2a^2}\right). \quad (3.10)$$

$\hat{K}$ is the average number of coupled neighbours per site (not necessarily equal to $k$). Here, $a$ is a local length scale which is set per default to the average distance between coupled spins. However, sometimes it may be advantageous to use differing length scales. Each spin configuration is characterised by an energy expressed by the Potts spin Hamiltonian

$$H(s) = \sum_{(i,j)} J_{ij}(1 - \delta_{s_i s_j}), \quad (3.11)$$

where the sum runs over all connections $(i, j)$ and $s$ denotes a spin configuration. The system is considered in the canonical ensemble. The probability for a certain spin configuration is thus given by the Boltzmann/Gibbs distribution

$$p(s) = \frac{1}{Z} e^{-H(s)/T}, \quad (3.12)$$

where the partition function $Z = Z(T)$ serves as a normalisation factor. As the temperature $T$ is increased, inhomogeneous Potts systems typically undergo a number of phase transitions. This is the inspiration source for the SC clustering algorithm. (I) For small $T$, the system is in the ferromagnetic phase, where spins like to be aligned. (II) At an intermediary $T$-range, a superparamagnetic phase occurs: strongly coupled spins tend to be aligned, whereas weakly coupled spins behave independently. Thus, clusters of aligned spins occur, reflecting groups of similar data items. A further increase of $T$ generally leads to a continued breaking-up of these clusters into smaller clusters, so that a hierarchy of classes and subclasses is obtained. (III) For high $T$, the system enters the paramagnetic phase where any order disappears and only singleton clusters remain.
3.2. SEQUENTIAL SUPERPARAMAGNETIC CLUSTERING SSC

At a given temperature $T$, clusters are identified with the help of the pair correlation: two points $i$ and $j$ belong to the same cluster, if the pair correlation

$$G_{ij} = \sum_s p(s) \delta_{s_i s_j}$$  \hspace{1cm} (3.13)

exceeds a given threshold $\Theta$, i.e.,

$$G_{ij} > \Theta.$$  \hspace{1cm} (3.14)

Whole clusters are defined by areas of the grid whose sites are connected through (3.14). Interestingly, the choice of $\Theta$ is uncritical as long as $1/q < \Theta < (1 - 2/q)$ for larger $q$ [107]. This is due to the fact that either $G_{ij} \approx 1$ or $G_{ij} \approx 1/q \approx 0$ for most pairs $(i,j)$, which reflects the robust self-organisation at work in these systems.

While up to here, we rather stressed the view that SC should be understood as a network self-organisation process, where order appears as a consequence of local interaction, it is interesting to also look at it from the top-down view of cost functions. As in section 3.1.1, the Boltzmann distribution (3.12) is obtained as the minimum of the free energy functional $F = H - TS$, where $H$ is the Hamiltonian (3.11) and $S$ is the entropy (note that above in section 3.1.1, $H$ denoted the entropy and $D$ denoted the energy). The difference to central clustering (soft k-means) discussed in section 3.1.1 lies in the cost function $H$ or $D$. This becomes particularly apparent in the limit of zero temperature. For $T = 0$, the energy landscape of central clustering looks like a spin glass [32] and it is difficult to find the global minimum, whereas in the SC case, the global minima are just the states of global alignment due to the ferromagnetic structure of the energy.

For large data sets, the computation of (3.13) is not feasible. Therefore, the pair correlation is usually approximated by an appropriate Monte Carlo simulation, such as the Swendsen-Wang [106] or the Wolf algorithm [74]. For data sets with $N \lesssim 10^5$, Swendsen-Wang turned out to be quite efficient, where we generally obtained reliable results for less than 230 Monte Carlo steps [62]. As an alternative to Monte Carlo methods, we proposed belief propagation (BeP) in chapter 2. Here, we briefly summarise the Swendsen-Wang algorithm and BeP in connection with clustering.

3.2.2 Swendsen-Wang Algorithm and BeP for SSC

Swendsen-Wang SW

The SW algorithm provides a Markov chain Monte-Carlo simulation whose steady state distribution corresponds to the Boltzmann distribution (3.12) [97]. The advantage of SW lies in the fact that it does not suffer from a severe critical slowing down effect as it is observed for local Monte-Carlo methods such as the Metropolis algorithm. Starting with a random configuration $s(0)$, the following steps are performed iteratively:

1. Cancel all the connections between spins that do not coincide in $s(n)$.

2. Two connected spins remain connected with probability $p = 1 - \exp(-J_{ij}/T)$.

3. Assign randomly a spin state to each connectivity component, yielding a new configuration $s(n + 1)$. 

The algorithm is terminated after $M$ steps and the pair correlation is approximated by

$$G_{ij} \approx \frac{1}{M} \sum_{n=1}^{M} \delta_{s_i(n)s_j(n)}.$$  

(3.15)

**BeP for superparamagnetic clustering**

In order to use BeP for SC, we need to implement the BeP equations described in chapter 2 for the Hamiltonian (3.11):

$$m_{i \rightarrow j}^{t+1}(s_j) = k_1 \sum_{s_i} e^{J_{ij} s_i s_j} \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}^t(s_i),$$  

(3.16)

The pair correlation is calculated according to $G_{ij} = \sum_{s_i, s_j} p_{ij}(s_i, s_j) \delta_{s_i s_j}$ based on the pairwise beliefs given by (2.30). In practice, to save computational time, we confined the number of possible Potts spin states to two. Hence, we actually deal with the Ising spin case. In chapter 2, we worked out that in this case the following choice for the initial messages leads to reliable convergence:

$$m_{i \rightarrow j}^0(1) = y \text{ and } m_{i \rightarrow j}^0(-1) = 1 - y,$$

(3.17)

with the same $0.5 < y < 1$ for all connections $i \rightarrow j$.

Two questions are of interest: (i) How fast is BeP in comparison to Monte-Carlo methods, and (ii) how good is the approximation provided by BeP? It turns out that clustering can be accelerated by a factor of at least 10 by using BeP instead of SW. Monte-Carlo requires about 200 steps for stable results at a fixed $T$, BeP can manage with about 10 to 20 iterations. We worked out the details of this statement in [67]. There, we also argued in favour of the quality of the results achieved by BeP. Although, the results achieved by BeP are not one-to-one identical to the results achieved by SW, clusters that persist over larger temperature ranges are equally well recognisable for both approaches. This fact is illustrated in Fig. 3.3, where the results for both approaches are compared for a simple toy system (Fig. 3.2). The stable clusters are the ones we are interested in for applications. The notion that stable clusters correspond to clusters with a long temperature life time is explicitly explored by our enhancement of SC, the sequential superparamagnetic clustering SSC. By extracting the stable or reliable clusters and reclustering the remainder, SSC iteratively re-generates reliable coarse-grained structures on the finer levels of a stability hierarchy. This procedure also guarantees that the results from Monte-Carlo and BeP are largely equivalent. SSC is discussed next.

### 3.2.3 The Sequential Procedure

Although SC has been successfully applied in different fields, it does not provide a direct evidence which clusters, among those emerging on different resolution levels $T$, should be selected as the natural ones. Clear clusters express themselves as regions of order that are stable over an entire phase, i.e., over a substantial range of $T$. The idea is thus to choose
3.2. **SEQUENTIAL SUPERPARAMAGNETIC CLUSTERING SSC**

Figure 3.2: A simple two-dimensional toy system.

Figure 3.3: A comparison of the results for Fig. 3.2 for BeP and SW shows that the main cluster structures are equally well recognisable for both approaches. a) Results for BeP, b) Results for SW.
the clusters that have the largest $T$-range extensions (denoted by $T_{cl}$). Consequently, we define the stability $s_T$ of a cluster as

$$s_T := \frac{T_{cl}}{T_{\text{max}}},$$

(3.18)

where $T_{\text{max}}$ is the temperature of the paramagnetic transition. In this way, $s_T$ expresses the stability of the cluster in relation to the stability of the whole set. If the clusters are selected according to the stability $s_T$, the procedure will still be suboptimal, as the most natural clusters need not always be the most stable ones. Differences in shape, density and size of clusters lead to different temperature ranges of occurrence. Sparse clusters only exist at small temperatures, whereas dense clusters are globally more stable and decay later. As a consequence, clusters of interest often emerge only for short $T$-ranges, after the breakup of dense super-clusters at higher temperatures. Inhomogeneities in shape, density and size of clusters can thus render the recognition of the best clusters a difficult task. To overcome these difficulties, we introduce a sequential procedure \[62\]. In this approach, the most stable cluster in terms of $s_T$ is extracted and it is, as well as the residual set, reclustered with readjusted weights (3.10). The procedure continues, resulting in a binary tree structure. The branches of the tree form sequences of sets of increasing homogeneities. Consequently, the detection of clusters becomes easier. The procedure stops in a branch, if no more stable substructures can be found, i.e., if the most stable cluster detected is less stable than a threshold value $s_\Theta$. Typically, natural clusters themselves do not have any substructures. They show a direct transition from the ferromagnetic phase to the paramagnetic phase. Therefore, the temperature that marks the end of the ferromagnetic phase $T_{\text{ferro}}$ is a good indicator for how natural a cluster intrinsically is. In contrast, $s_T$ gives the information how natural the cluster is in relation to the clustered mother set. $T_{\text{ferro}}$ is also useful for judging whether the last residual set should be considered an intrinsic (i.e., natural) cluster or a set of unclustered (i.e., background) points. In the first case, $T_{\text{ferro}}$ is of the same magnitude as for the other detected clusters, whereas in the second case, $T_{\text{ferro}}$ is vanishingly small.

In summary, the following program is executed:

1. For the input set $S$, determine the weights according to (3.10) and apply SC.
2. Determine the most stable cluster $C$ in terms of $s_T$.
   If $s_T > s_\Theta$: Go back to (1) with $S_1 = C$ and $S_2 = S \setminus C$.
   Else: Determine $T_{\text{ferro}}$.
3a) If $S = S_1$ in the last loop: Identify $S$ as a natural cluster.
3b) If $S = S_2$ in the last loop or $S = \text{initial set}$:
   Use $T_{\text{ferro}}$ to judge whether $S$ is a natural cluster
   or a set of unclustered points.

We emphasise that in this way, $s_\Theta$ is the only control parameter that is set from outside. The procedure does not require to maximise the superparamagnetic phase \[1\] and it also works on Potts spin graphs composed of several components. For sets with clear cluster structures, the choice of $s_\Theta$ is not critical, as we can expect that $s_T > 0$ (and $T_{\text{ferro}} > 0$).
3.3. SSC APPLICATIONS

only holds for the actual clusters. For sets with less clear structures, the user needs to choose an appropriate minimal stability \( s_0 \). How this is properly done is illustrated in more detail in the examples discussed below. As a further benefit of the sequential procedure, the choice of the number of nearest neighbours \( k \) becomes an uncritical issue. Changing \( k \) in the range between 5 and 15 is usually without any effect on the final result. Thus, for computational reasons, the \( k = 5 \) has been chosen as the default value.

3.3 SSC Applications

3.3.1 A Toy System

In this section, a simple toy system is used to illustrate the basic advantages of SSC. In Fig. 3.4, a 2-dimensional distribution consisting of three clusters is displayed. Whereas the clusters have the same size and shape, they differ considerably in the density.

Figure 3.4: Toy system with 3 clusters of different densities.

Fig. 3.5 shows the results obtained by SC. Remarkably, no interval \([T_{min}, T_{max}]\) can be
found for the sparse cluster 3. After going through a short phase in a joint cluster with cluster 1, cluster 3 immediately decays into smaller units. Thus a detection of cluster 3 is intrinsically impossible. However, our automated SSC detects the three clusters without problems, see Fig. 3.6. In this diagram, extracted clusters form left branches and residual sets form right branches. In each box, the size of the set $N$, the length of the ferromagnetic phase $T_{\text{ferro}}$, the length of the $T$-range of the most stable cluster $T_{\text{cl}}$ (if any), and the paramagnetic transition temperature $T_{\text{max}}$ are reported. For the extracted clusters, the stability values $s_T = s$ are given as well. Since $T_{\text{ferro}}$ is of the same magnitude for all three clusters, i.e., between 0.052 and 0.082, the residual set (the rightmost one) should be considered a natural cluster. When clustering the mother set, the last two clusters emerge as subclusters (with $s_T > 0$), which is consistent with the above-made cluster identification. The findings are very stable against variations of $s_\Theta$: any value in $s_\Theta \in [0.05, 0.58]$ reflects in the same stable and clear-cut cluster structure. As a typical default value, $s_\Theta = 0.2$ can be taken.

### 3.3.2 Classifying Chemical Compounds

The rationale of clustering in chemistry is given by the assumption that structurally similar compounds are likely to exhibit similar properties. The search for new drugs among thousands of different compounds in a chemical library can thus be greatly facilitated if the library can be sectioned into natural classes of structurally similar chemicals. For optimally
chosen classes, efficient testing can be achieved. In [62], we investigated the potential of SSC for chemoinformatics in detail; here we report on the results obtained from our automated implementation of SSC. The test set comprised 153 compounds from 7 different chemical classes (Fig. 3.7). While the appropriate description of the compound’s structure by chemical fingerprints or keys, i.e., high-dimensional vectors, and the choice of a good similarity measure are complex issues themselves (for details see [62]), we here concentrate on the very aspect of clustering. The results achieved by SSC were compared to results by other methods, notably Ward clustering, which in the field of chemoinformatics is regarded to yield the most reliable results [104]. Yet, among the tested methods, SSC was the only one able to reliably reconstruct the seven involved chemical classes (up to about three misclassified compounds depending on the chemical fingerprint descriptors and the similarity measure used [62]). The direct comparison of Ward clustering and SSC is documented in table 3.2. To measure the difference between the clustering results and the ideal result, we computed the Jaccard coefficient [103] for the clusterings obtained with the different descriptors, relative to the ideal clustering (1 is perfect and 0 is complete failure). When using Ward’s method, the optimal number of clusters was determined by applying the Kelley measure [103]. The Kelley measure is an accepted cluster level selection criterion that balances between the number of clusters and their inherent densities.
### Table 3.2: Comparison between Ward clustering and SSC for the chemical test set. For explanations see text.

<table>
<thead>
<tr>
<th>method</th>
<th>keys</th>
<th>similarity measure</th>
<th>no. of clusters</th>
<th>Jaccard coeff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ward</td>
<td>ISIS, binary</td>
<td>Euclidian</td>
<td>19</td>
<td>0.495</td>
</tr>
<tr>
<td>SSC</td>
<td>ISIS, binary</td>
<td>Euclidian</td>
<td>9</td>
<td>0.956</td>
</tr>
<tr>
<td>Ward</td>
<td>ISIS, binary</td>
<td>Tanimoto</td>
<td>14</td>
<td>0.654</td>
</tr>
<tr>
<td>SSC</td>
<td>ISIS, binary</td>
<td>Tanimoto</td>
<td>9</td>
<td>0.956</td>
</tr>
<tr>
<td>Ward</td>
<td>ISIS, count</td>
<td>Euclidian</td>
<td>15</td>
<td>0.674</td>
</tr>
<tr>
<td>SSC</td>
<td>ISIS, count</td>
<td>Euclidian</td>
<td>9</td>
<td>0.941</td>
</tr>
<tr>
<td>Ward</td>
<td>ISIS, count</td>
<td>Tanimoto</td>
<td>16</td>
<td>0.613</td>
</tr>
<tr>
<td>SSC</td>
<td>ISIS, count</td>
<td>Tanimoto</td>
<td>8</td>
<td>0.986</td>
</tr>
<tr>
<td>Ward</td>
<td>Similog</td>
<td>Euclidian</td>
<td>53</td>
<td>0.407</td>
</tr>
<tr>
<td>SSC</td>
<td>Similog</td>
<td>Euclidian</td>
<td>24</td>
<td>0.420</td>
</tr>
<tr>
<td>SSC</td>
<td>Similog</td>
<td>Tanimoto</td>
<td>15</td>
<td>0.534</td>
</tr>
</tbody>
</table>

**Figure 3.8:** Clustering result for the chemical data set achieved by SC.
3.3. SSC APPLICATIONS

From applying SC without the sequential procedure, the origins for the failure of other algorithms become apparent. In Fig. 3.8, the clusters detected by SC are displayed. While some of the classes are easily detectable by manifesting themselves as stable clusters from the very beginning (e.g., the class of proton pump inhibitors), other classes hardly ever appear as stable clusters. Most prominently, the classes of statins and corticosteroids form a joint stable cluster (with size 45) that for large $T$'s quickly decays into smaller units, not revealing the actual chemical class structures. The reason for this behavior can be found in the intra- and inter-cluster distances. For all tested similarity measures, the classes statins and corticosteroids are much compacter than all other clusters. Moreover, the distance between these two classes is found to be much smaller than the distances between other pairs of classes. This distance is even smaller than the average compound distance within some of the other classes. Thus, in a loose sense, from far the two classes look like one single compact cluster. Only after zooming in (which is what we do with SSC), the cluster reveals its actual substructure. In fact, Fig. 3.9 shows that SSC successfully identifies the seven original chemical classes. In this example, SSC has been optimised to provide the right number of classes. In terms of performance, SSC would actually further subdivide the two marginally stable clusters to achieve higher purity (see table 3.2).

3.3.3 SSC: Why Does It Work?

So far, SSC has been introduced as a heuristic concept whose justification is to a large extent in its success in applications. In the following, we will provide some analytical considerations applied to a simple mean-field like model in order to reveal the origin of this success.

In Fig. 3.10, three large clusters $C_1$, $C_2$ and $C_3$ have $N \gg 0$ spins each. The internal connectivity of each cluster $C_l$, $l = 1, 2, 3$ is as follows: each spin is connected to $k_l$ neighbours. The corresponding distances (or similarity values) are fixed values $d_l$. There is exactly one connection between the clusters. This means, no more than one of the spins from cluster 1, $s_{1m}$, is connected to one of the spins of cluster $m$, $s_{ml}$, where the corresponding distances are denoted by $b_* := b_{12}$ and $b := b_{13} = b_{23}$. We generally assume that $b \geq b_* \geq d_1, d_2$ and $b \geq d_3 \geq d_1, d_2$. This situation sketches the situation encountered in the preceding examples. For simplicity, the system is treated in the familiar Ising spin formalism, i.e., the Hamiltonian is given by

$$H(s) = - \left( J^b s_{12} s_{21} + J^b (s_{13} s_{31} + s_{23} s_{32}) + \sum_{l=1,2,3} \sum_{(i,j) \in C_l^2} J^l s_i s_j \right), \quad (3.19)$$

where $s_i$ and $s_j$ can take on the values -1 and 1. $J^l$ is the coupling strength within the cluster $C_l$ corresponding to the distance $d_l$. $J^b$ and $J^{b*}$ correspond to $b$ and $b_*$. The connections are determined according to (3.10). The sum $(i, j)$ is taken over all connected pairs in each cluster. Each $s_{im}$ reappears in this sum as one of the $s_i \in C_l$.

If we neglect the influences from the other clusters by means of two single connections, each cluster is independent. For each cluster, we can thus discriminate a ferromagnetic and a paramagnetic phase whose transition can be estimated for not too small $k_l$ from the mean-field self-consistency equation for Ising spin models

$$\langle s_i \rangle = \tanh(T_c^l \langle s_i \rangle / T) \text{ with } T_c^l = J^l k_l. \quad (3.20)$$
Figure 3.9: Our automated standard SSC successfully reveals the 7 chemical classes from which the test set is composed. The only error is the assigning of three compounds to the class of statins instead of corticosteroids, yielding class sizes 24 and 21 instead of 27 and 18. A relatively small value of $s_T = s = 0.07$ indicates that this subdivision is less evident than other divisions in the set. For the set, ISIS binary keys with Euclidian distance was used. The results can be improved by using ISIS count keys with our extended Tanimoto similarity measure [62].
3.3. SSC APPLICATIONS

For a first estimate, we can use the critical temperature \( T_{cl} = Jl_kl \) as the indicator for the decay of \( C_l \) (We assume a short decay phase of \( G_{ij} \) from 1 to 1/2. The effective \( T_{cl} \) is slightly lower, depending on \( \Theta \). \( \Theta = 0.7 \) is a typical threshold value for \( q = 2 \) [66]). Furthermore, we can estimate the length of the global ferromagnetic phase by the temperature

\[
T^b_c = \frac{-2J^b}{\ln(1/\Theta - 1)} = k_\Theta J^b.
\]

At this temperature the configuration \( \hat{s}_{13} = \hat{s}_{31} \) (and \( \hat{s}_{23} = \hat{s}_{32} \) respectively) loses stability, i.e.,

\[
p(\hat{s}_{13} = \hat{s}_{31}) = \frac{\exp(J^b/T)}{(\exp(J^b/T) + \exp(-J^b/T))} \leq \Theta \text{ for } T \geq T^b_c.
\]

This estimation is justified since for these low temperatures the clusters are still fully ordered. Similarly, the conglomerate of the clusters \( C_1 \) and \( C_2 \) decays into its components at \( T^b_c = k_\Theta J^b \). In Fig. 3.11, the typical phase diagram is displayed.

We now have a simple model to hand that allows us to understand some of the advantages of sequential clustering.

\textbf{a)} It is clear that if \( b_* \ll d_3 \) then \( J^3 \ll J^b_* \), and thus \( T^3_c < T^b_c \). In this case, \( C_3 \) decays way before \( C_1 \) and \( C_2 \) emerge as single independent clusters. In other words, a single level \( T \) that would contain all three clusters does not exist. Sequential clustering, however, does not rely on the specification of one (or several) resolution level(s) \( T \).

\textbf{b)} If \( b \approx d_3 > d_2 > d_1 \) then \( 0 \approx T^b_c \lessapprox T^3_c \lessapprox T^{b*}\lessapprox T^2_c \approx T^1_c \). In this situation, \( C_1 \) dominates the picture and the stability of \( C_3 \), \( s^3_f = (T^3_c - T^b_c)/T^1_c \), is marginal compared to the the stability of \( C_1 \), \( s^1_f = (T^1_c - T^{b*}_c)/T^1_c \). However, the stability \( s^3_f \) is greatly enhanced to \( s^3_f = (T^3_c - T^{b*}_c)/T^2_c \approx T^1_c \) if \( C_1 \) is removed, allowing a clear detection of \( C_3 \). This explains why SSC could successfully detect the sparse cluster in the toy system of the first example above.

\textbf{c)} The last argument in b) holds as long as a change of the weights \( J \), which is mainly a consequence of a changed local parameter \( a \) in (3.10), can be neglected. This change,

![Figure 3.10: Model system with 3 clusters that are pairwise connected through one bond.](image-url)
however, can be crucial in other cases. Consider a situation with \( d_1 = d_2 \approx b_* \ll d_3 \). Then \( a \approx d_1 \) is to a large extent dominated by \( d_3 \). If, however, \( C_3 \) is removed and only the set containing \( C_1 \) and \( C_2 \) is clustered, we observe \( a \approx d_1 \). Due to (3.10), \( J^1 \) and thus \( T^1_c \) is larger in the first case. However, in the second case the difference \( \Delta J = J^1 - J^{b_*} \) is larger as \( \Delta J(a) \) is a decreasing function in \( a > d_1 \). The collected effects cause the stability

\[
\Delta^1 J = \frac{T_c^1 - T_c^{b_*}}{T_c^1} = \frac{k_1 J^1 - k_Theta J^{b_*}}{k_1 J^1} = 1 - \frac{J^{b_*}}{J^1} = 1 - k \exp \left( \frac{d_2^2 - b_2^2}{2a^2} \right)
\]

(3.23)

to decrease, as a function of \( a \). In other words, removing \( C_3 \) enhances the stability of the clusters \( C_1 \) and \( C_2 \). These structures therefore can be found by clustering the subset containing \( C_1 \) and \( C_2 \) which has a large stability and is thus easy to detect. In contrast, we might not be able to extract \( C_1 \) and \( C_2 \) when clustering a superset including also some less dense clusters. In this case, due to a small degree of stability, the \( C_1 \) and \( C_2 \)-structures may remain unrevealed. This explains why SSC could successfully uncover all the chemical classes in the example given above. In conclusion, inhomogeneities decrease the stability of single clusters. SSC counteracts this problem by successively producing sets of increased homogeneity.

### 3.3.4 Visual Scene Analysis Application

Here we touch another field of application of clustering techniques of current interest: visual scene analysis. The overall goal of visual scene analysis is to identify and classify the (relevant) objects present in the visual field (of a visual sensor, such as a biological or artificial retina or a camera). A reliable scene analysis system also plays an important role in the realisation of autonomous robots. Obviously, autonomously interacting robots require some degree of visual scene understanding. Usually, the visual information is provided in the form of digital images, i.e., a two-dimensional matrix \( f(x,y) \), where \((x,y)\) denotes the
Figure 3.12: Different objects on a noisy background.

spatial coordinates and $f(x, y)$ is the according feature value. In our simple studies, $f(x, y)$ codes for the pixel intensities of a grey-scaled picture.

Clustering has been proposed as a unifying principle for all levels of image understanding [22]. Low-level analysis is associated with scene segmentation by means of pixel clustering, whereas clustering of edge elements for perceptual grouping is referred to as mid-level analysis. High-level analysis is associated with clustering of whole objects of the visual scene, in order to find object categories.

In the following, we concentrate on a simple low-level task, the separation of objects (i.e., homogeneous regions) from a noisy background. Our working example is shown in Fig. 3.12, where two black objects of size 65 and 25 pixels, respectively, and three small white objects of size 12, 8, and 8 pixels, are embedded. For clustering, to each pixel $i$ a Potts spin $s_i$ is assigned. The connectivity graph is obtained by connecting each pixel to its 4 adjoined neighbours. The weight of a connection is again given by (3.10). The distance between two connected pixels $i$ and $j$ is determined by the pixel values difference, i.e.,

$$d_{ij} = |f(i) - f(j)|,$$

where $f(i) = f(x_i, y_i)$ and the pixels are numbered in some way. The task posed by Fig. 3.12 might appear highly factitious, but it has a pedagogic intention. We now deal with a crucial difference to the examples discussed above: the connectivity of the Potts spin system does not just reflect an actual neighbourhood relation in terms of similarity of the quantities that are expressed by the spin (i.e., the pixel values). Instead, it is forced from outside by the pixel arrangement. As a consequence, the original choice of the local parameter $a$ (equation (3.10)) turns out to be problematic. However, convincing results (Fig. 3.13) are achieved over a large range of constant, but relatively small values of $a$. For the displayed results, $a = 0.01$ was chosen. The residual set (782 pixels) has a clearly shortened ferromagnetic phase $T_{ferro}$ compared to the ferromagnetic phase of the extracted clusters. It is thus clearly classified as background.

### 3.3.5 Spike Sorting with SSC

Spike sorting is a fundamental step in the analysis of neural recordings. The spike sorting problem can be seen as a simpler analogue of the cocktail party problem in the field of
Figure 3.13: Clusters recognised in Fig. 3.12. The residual set (782 pixels) must be interpreted as background since $T_{\text{ferro}} \approx 0$. A minimal cluster size of 5 pixels was used.
3.3. SSC APPLICATIONS

Figure 3.14: A set of 241 wave traces from extracellular recordings of a single electrode. The time resolution is 30 kilo-samples/s, the amplitude is measured in $\mu V$.

auditory scene analysis. In both cases, we want to extract one or more signals from a mixture of different signals and noise. The goal of spike sorting is to detect spikes in extracellular voltage recordings and assign them to their different sources, i.e., the neurons. Common spike sorting schemes are based on two ingredients [43]: (I) The description of the spike-underlying recorded waveforms by means of – usually high-dimensional – vectors, and (II) a clustering algorithm that identifies groups of similar waveforms. Spikes in each group are assumed to originate from the same neuron. This is a reasonable assumption if the underlying processes are stationary and the number of spike overlaps is small. For the clustering algorithm, two often conflicting properties are desirable. Firstly, the algorithm should be as unbiased as possible, i.e., it should not rely on prior information about the number or the shape of the clusters and it should also work for high dimensions. Secondly, it should be reasonably fast.

In the preceding sections, we introduced SSC as a powerful nonparametric algorithm that, in combination with BeP, is reasonably fast, as well. We thus anticipate that it is an ideal algorithm for spike sorting.

The Basic Stages of Spike Sorting

Extracellular recordings by a single electrode provide a waveform train made up of the activity of an unknown number of neurons and of background noise. In a first step, potential spike shapes are identified by a thresholding procedure: if the recorded voltage crosses a threshold, a sample vector (wave trace) containing the voltage values over a certain time
window length is selected. In our example\(^1\) (Fig. 3.14), the sample vector is of dimension 48. The choice of the threshold is delicate. For too small thresholds, noise artifacts may be detected and noise fluctuations can lead to a bad alignment of the peaks. Although for the latter can be corrected by a subsequent alignment, a fixed window size may still cause problems. If the threshold is too large, some spikes might be missed. In practice, one tries to use relatively large thresholds that still catch all spikes. In a next stage, the spike traces need to be characterised in the form of feature vectors which then allow for a comparison and cluster identification. For this purpose, different strategies have been introduced. A simple characterisation of the spikes can be achieved by a small number of characteristic features such as spike height or spike width. Sometimes, the whole sample vector is used for a comparison by means of the Euclidian distance. More frequently, the dimension is reduced by a principal component analysis (PCA) that can be considered an automatic feature selection. This sometimes allows for a straightforward detection of clusters by eye on the basis of the first two principal components. Alternatively, the wave trace may be characterised by Fourier- or Wavelet coefficients \([74]\), or by coefficients of any other transform that works out suitable characteristics. In short, the clustering preprocessing phase provides us with feature vectors \(v_i\) whose distances are given by an appropriate measure.

In the final stage, we can thus perform the actual clustering. Although we should keep in mind that the preprocessing is of equal importance, we entirely focus on this last stage in the following. The prospects of a successful spike sorting vanish if a cluster spreads over the whole feature space or if different clusters largely overlap. In such cases, the preprocessing must be revised.

**Performance**

The advantages of SSC if compared to the often used k-means algorithm (or similar algorithms) are obvious: we do not need to know the number of clusters and there is no shape bias. Only if the number of clusters is known and their shape is spherical, then it might be advantageous to use k-means. Unfortunately, a direct comparison of the performance of different algorithms based on real-world data recordings is often difficult. Either the correct sorting is unknown and a standard solution to compare with is missing, or the sorting is rather simple and all algorithms perform well. The advantages and superiority of SSC, however, have been demonstrated above (e.g., table 3.2). Here, we do not provide a systematic discussion, but only give an impression by means of two examples.

In Fig. 3.15, SSC has been applied to an artificially created data set with 3 clearly distinguishable basic waveforms\(^2\). The 3 classes are easily recognised by SSC without specifying any parameters (b)-d)).

In Fig. 3.16, we show the clustering result obtained by SSC for the set shown in Fig. 3.14. The clustering was performed on the first two principal components PCA1 and PCA2 only. A fast coarse-grained clustering with a large \(s_\Theta = 0.5\) yields 2 clusters (Fig. 3.16(a)). Cluster 1 (Fig. 3.16(b)/3.16(c)) is the purer cluster as it is extracted first. Cluster 2 (Fig.

---

\(^1\)Data from a randomly chosen set of recordings from macaque parietal cortex provided by H.Scherberger and collaborators.

\(^2\)This data set *testspikes.mat* is provided by R.Q.Quiroga and can be downloaded under www.vis.caltech.edu/~rodri
Figure 3.15: a) An unsorted set of artificial data and b)-d) the sorting solution provided by SSC. Axes in arbitrary units (the recorded wave traces correspond to 65 dimensional sample vectors).
CHAPTER 3. CLUSTERING ALGORITHMS

Figure 3.16: Spike sorting results for Fig. 3.14. A fast coarse-grained clustering by SSC provides 2 clusters. a) SSC-dendrogram. b)-e) Clusters and corresponding wave traces.
3.3. **SSC APPLICATIONS**

3.16(d)/3.16(e)) is the residual cluster. It still contains some clear outliers (Fig. 3.16(d)). Fortunately, SSC provides information about the reduced purity of cluster 2 in the form of $T_{\text{ferro}}$. $T_{\text{ferro}}$ for cluster 2 is much smaller than $T_{\text{ferro}}$ for cluster 1 (Fig. 3.16(a)). Consequently, by using a smaller $s_\Theta$, cluster 2 will be further decomposed into a core cluster and a residual cluster (cluster halo) with again marginal intrinsic coherence (small $T_{\text{ferro}}$; results not shown).

### 3.3.6 Spike train classification

M. Christen applied SSC for spike train classification in connection with different similarity measures, addressing several scientific questions \[14, 15\]. Here, we report on the study on the classification properties of the LZ-distance (Lempel-Ziv distance; for details, we refer to [15]).

The investigated question was whether the LZ-distance could correctly classify spike trains of physiological and simulated data when clustering spike trains with similar, but not synchronous patterns. For the first step, a set of spike trains was generated, consisting of in vivo and model data of comparable firing rate (80-90 spikes/s), as classes of spike trains that differ substantially in firing rate are easily recognised using the LZ-distance. The used multi-train data set contained the following five classes, each represented by nine spike trains of length 2400 ms. Class (A): spike trains of a complex cell (macaque monkey visual cortex data, for further explanations, see \[15\]) driven by gratings drifting at 6.25 Hz. Class (B): spike trains of a simple cell driven by gratings drifting at 12.5 Hz. Class (C): spike trains of a homogeneous Poisson process with refractory period that models the firing of the recorded complex cells. Class (D): spike trains of an inhomogeneous, step function driven Poisson process with refractory period that models the firing of the recorded simple cells. Class (E): Poisson spike trains containing weakly synchronised burst patterns. The overall firing rate was similar to that of the other four classes.

The order of the spike trains was randomised and the LZ-distance between all trains was calculated (see Fig. 3.17 (a)-(c)). Clustering led to the following result: The classes B, D and E formed one cluster each, whereas the classes A and C fell into a single cluster (Fig. 3.17(c)). When the algorithm was applied to the latter cluster for a decreased minimal cluster stability $s_\Theta$, only an incomplete separation between spike trains of the classes A and C occurred, as two smaller clusters of five (spike trains of class C) and 13 elements each (spike trains of classes A and C) emerge (not shown).

Two main conclusions can be drawn from this result. First, the metric is able to classify correctly spike trains with comparable firing rate, but differing temporal structures. Second, the incomplete separation between the classes A and C can be interpreted as follows. The spike trains C derive from a model of the firing of a complex cell and the incomplete separation from the spike trains of the complex cell indicates that the firing behaviour of the cell appears to be (in a first approximation) properly modelled by a Poisson process with refractory period. The sinusoidal firing of the simple cell, however, could be distinguished from the inhomogeneous Poisson process driven by a step function, as the two classes B and D are separated by the clustering algorithm.
Figure 3.17: Clustering of multi-train data. a) Raster plot of initial spike set. b) Distance matrix (white: $d(X, Y) = 0$, black: $d(X, Y) = 1$) and histogram of distances obtained after calculating the pairwise LZ-distance. c) Dendrogram outlining the result of clustering.
3.3.7 Auditory Scene Analysis

The goal of auditory scene analysis is to decompose an acoustic signal into its constituent sources. The general idea behind our approach can be described by three major steps: 1) The incoming sound signal is transformed into a spectrogram. 2) From the spectrogram, feature vectors are extracted that describe the atomic parts of the scene. 3) Atoms that belong to the same acoustic source are grouped together by a clustering process. The auditory scene analysis problem is generally considered a very difficult problem. The main difficulty is the extraction of appropriate features that allow for a successful grouping. A. Kern found in his work that local cosine packets are able to represent speech signals in an appropriate way such that their components can be used as input for SSC. The approach was successfully applied to noise cleaning and vowel separation [39]. The results are a promising step towards a complete artificial hearing system. In such a system, the auditory feedback loop can ideally be closed by modulating the early stage signal in a top-down manner (ongoing research by Stoop group). This would be a realisation of our paradigm of embedded clustering (section 1.1.2).

3.4 SSC Recapitulation

Sequential superparamagnetic clustering provides a unique natural clustering solution, where the algorithm can be implemented in an automated way. Unlike other algorithms, the approach is introduced as a heuristic concept, and is not directly based on the optimisation of a chosen objective function. The approach takes advantage of the robustness of self-organisation processes in Potts spin systems in two respects. On the one hand, superparamagnetic clustering uses the statistical robustness properties of the Boltzmann distribution to evaluate an averaged, balanced and thus natural clustering for different resolution levels $T$. On the other hand, the sequential procedure reaches across the clusterings obtained for different $T$’s, in order to compare the stability ($s_T$) of all found clusters within the clustered set.

However, the sequential clustering concept does more than just search for the most stable clusters in the original set. By extracting the most stable sets and reclustering, the algorithm overcomes the problem that natural entities or classes that are inhomogeneously represented in the feature space, have to be selected against their respective background. Some classes therefore are not found in the original set, but only in a subset of adequate homogeneity. This also expresses itself in the fact that, for some clusters, no longer a global resolution level $T$ can be specified. Instead the adequate resolutions for a best unique choice of clusters become dependent on a ‘local’ clustering context. As a consequence of the sequential procedure, the clusters identified by the algorithm tend to be homogeneous themselves. This is consistent with the strictly non-parametric approach of our two-stage clustering paradigm that prohibits the assumption of any particular consistency or internal distribution of clusters.

The rationale behind this paradigm is the notion that clusters are not actual entities by themselves, but a fragmentary representation of some kind of real class entities that express a structural dependence in the world. Clustering therefore should not just be considered a simple data or information compression. Rather, clustering is a means for accessing hidden
information about object relations. Eventually, this allows for unexpected predictions based on the similar-property principle, i.e., the experience that (structurally) similar objects tend to behave similarly in many situations. In our future work, we aim to use SSC for on-line applications with real-world sensory input. The real-world embedding is ultimately an important step towards real machine intelligence.
Appendix A

A.1 Proof of Theorem 1.1

A generative grammar $G$ consists of the following components [12]:

- A finite set $N$ of nonterminal symbols.
- A finite set $\Sigma$ of terminal symbols.
- A distinguished symbol $S \in N$ that serves as start symbol.
- A finite set $P$ of production rules of the form $(\Sigma \cup N)^* N (\Sigma \cup N)^* \rightarrow (\Sigma \cup N)^*$.

That is, each production rule maps from one string of symbols to another, where the first string contains at least one nonterminal symbol. A (right) regular grammar is a generative grammar whose production rules are of the following form:

- $A \rightarrow a$, where $A \in N$ and $a \in \Sigma$.
- $A \rightarrow aB$, where $A, B \in N$ and $a \in \Sigma$.
- $A \rightarrow \epsilon$, where $A \in N$ and $\epsilon$ is the empty string.

In order to prove that $L_{A,f_s}$ is regular, we define $N = K \cup \{S\}$ (S is an additional start symbol) and $\Sigma = C \cup \{\epsilon\}$. It is straightforward to see that the following production rules generate any word of $L_{A,f_s}$ (as any path of the graph is covered by the rules):

- $S \rightarrow \epsilon, \epsilon n_i$ ($n_i \in K$).
- $n_i \rightarrow f_s(n_i)n_j$ if $A_{n_i,n_j} = 1$ ($n_i, n_j \in K$).
- $n_i \rightarrow f_s(n_i)$.

This is essentially a right regular grammar, therefore $L_{A,f_s}$ is a regular language. $\square$
A.2 Attractor Clustering in Recurrent Networks

In our concept (see page 31), the network learns online, which means, the process of learning never really stops. Therefore, the weight matrix is time-dependent. Learning is driven by an internal synaptic dynamics which again is influenced by the last presented input pattern. The internal states of the synapses may be stored in another symmetric $m \times m$ matrix $S$, where $s_{ij}(t)$ is the internal state of the synapse connecting neuron $i$ and $j$ at time $t$.

The time scales of the global dynamics of the network and the local dynamics of the synapses are not necessary the same. We presume, on the one hand, that the synaptic dynamics is so slow compared to the global dynamics that it is not influenced by the global dynamics. On the other hand, we assume that the process of receiving (watching) a new input pattern creates a stable activity in the network, long enough to affect the synaptic dynamics. After the relaxation of the global dynamics to a fixed point, the network can quickly pick up a new input pattern.

Changes of the weight in turn are only possible in a stable phase of the synaptic dynamics, i.e., when the dynamics reaches a fixed point $s_{ij}(t) = f(s_{ij}(t), w_{ij}(t), x_i(t), x_j(t))$, where $f$ is the function that rules the transformation of $s_{ij}$. Its exact form has to be determined.

The updating of the whole system is divided into two parts:

In a first part, the internal states and the weights are updated after:

$$s_{ij}(t+1) = f(s_{ij}(t), w_{ij}(t), x_i(t), x_j(t)),$$  \hfill (A.1)

for all $i \neq j \in \{1, 2, \ldots, m\}$, and

$$w_{ij}(t+1) = \begin{cases} s_{ij}(t) & \text{if } s_{ij}(t) = f(s_{ij}(t), w_{ij}(t), x_i(t), x_j(t)) \\ w_{ij}(t) & \text{else.} \end{cases} \hfill (A.2)$$

Here, $x(t) = \xi^n$ is an input pattern. The order of the updating of the variables is not relevant.

In a second part, we use the update rule (1.20) for updating the global dynamics and hold the synaptic variables constant until a fixed point is reached. In other words, the network is updated as long as the pattern is unstable. This update schedule is expressed in the following program:

**Update algorithm:**

```
t=0;
w_{ij}(0) = s_{ij}(0) = 0 \text{ for all } i,j;
1 \text{ Read external pattern } y;
x_k(t)= y_k \text{ for all } k;
for i=1,...,m
    for j=i+1,...,m
        w_{ij}(t) -> w_{ij}(t+1);
        w_{ji}(t+1)= w_{ij}(t+1);
        s_{ij}(t) -> s_{ij}(t+1);
        s_{ji}(t+1)= s_{ij}(t+1);
next j
```
next i
xk(t+1)=xk(t) for all k;
t=t+1;

while (pattern unstable)
( for k=1,...,m,
  xk(t) -> xk(t+1);
  xl(t+1)=xl(t) for all j!=k;
  wij(t+1)=wij(t),sij(t+1)=sij(t)
    for all i,j;
  t=t+1;
  next k
)
go to 1

How should we choose $f$? A necessary requirement is that after a couple of presentations of the same picture, the synaptic weight $w_{ij}$ must be updated according to the learning rule chosen. The following choice of $f$ is appropriate:

$$f(s_{ij}, w_{ij}, x_i, x_j) =
\begin{cases}
  s_{ij} + \epsilon_{ij}^+ & \text{if } x_i x_j = 1, s_{ij} \leq w_{ij}^+ - \epsilon_{ij}^+ \\
  w_{ij}^- & \text{if } x_i x_j = 1, s_{ij} > w_{ij}^+ - \epsilon_{ij}^+ \\
  s_{ij} - \epsilon_{ij}^- & \text{if } x_i x_j = -1, s_{ij} > w_{ij}^- + \epsilon_{ij}^- \\
  w_{ij}^+ & \text{if } x_i x_j = -1, s_{ij} \leq w_{ij}^- + \epsilon_{ij}^-, 
\end{cases}
$$

(A.3)

where $w_{ij}^\pm$ and $\epsilon_{ij}^\pm$ are given by:

$$w_{ij}^\pm = \frac{1}{m} \Phi(m w_{ij} \pm \beta)$$

$$\epsilon_{ij}^\pm = \epsilon \left( \frac{1}{m} \Phi(m w_{ij} \pm \beta) - w_{ij} \right)$$

for the rule (1.21). An extension for other rules is straightforward in a similar way.

$\epsilon$ is small number that controls the number of presentations needed in order to store a pattern.

### A.3 Bethe-Peierls Approximation

We note that the ‘traditional’ way of deriving the critical point in the Bethe-Peierls approximation via the partition sum works likewise for the thermodynamic limit and periodic grids. In both cases, the same self-consistency equation is obtained by taking advantage of the translational invariance of the grid. We treat the interaction of a central spin $s_0$ and its $q$ coupled nearest neighbour spins $s_j$ exactly and the effect of all other spins on the nearest neighbours is subsumed in an effective field $h_{eff}$. The corresponding Hamiltonian then takes
on the form

\[ H_B = -J s_0 \sum_j s_j - h s_0 - h_{\text{eff}} \sum_j s_j. \]  

(A.4)

The corresponding partition function is given by

\[ Z_B = \sum_{s_n=\pm 1, s_j=\pm 1} e^{-H_B/T} \]  

(A.5)

= \[ 2^q(e^{h/T} \cosh^q((J + h_{\text{eff}})/T) + e^{-h/T} \cosh^q((J - h_{\text{eff}})/T)) \].

Due to the translational symmetry of the (infinite or periodic) grid, a self-consistency equation is obtained by the requirement \(<s_0> = <s_j>\), i.e.,

\[ <s_j> = 1/q \frac{\partial}{\partial (h_{\text{eff}}/T)} \ln Z_B = T \frac{\partial}{\partial h} \ln Z_B = <s_0>. \]  

(A.6)

To find the critical temperature, we require \(h = 0\) and obtain

\[ \frac{\cosh^{q-1}((J + h_{\text{eff}})/T)}{\cosh^{q-1}((J - h_{\text{eff}})/T)} = e^{2h_{\text{eff}}/T}. \]  

(A.7)

The critical condition for a solution \(h_{\text{eff}} \neq 0\) is \(\tanh^{-1}(J/T) = 1/(q - 1)\) which coincides with the condition found for BeP criticality.

□

Remark:

i) At first glance, it may seem surprising that a finite system can show a phase transition that coincides with the phase transition obtained in the thermodynamic limit. This is in fact a speciality of the Bethe free energy framework. In contrast, for the exact solution the free energy is analytic for all finite systems, but non-analytic in the thermodynamic limit. Hence the phase transitions of finite systems must be distinguished from the phase transition in the thermodynamic limit. This observation does no longer hold for the Bethe-Peierls approach. Here, also for finite periodic systems the free energy \(F_B = -T \ln Z_B\) is non-analytic, as the effective field \(h_{\text{eff}}\) has to be determined self-consistently. The approximation for a finite periodic grid can thus be identical with the approximation in the thermodynamic limit.

ii) We naturally expect that \(F_B = -T \ln(Z_B/2^q) = \min \ G_{\text{Bethe}}\) (after normalisation necessary due to multiple counting). In the paramagnetic phase \((T > T_{\text{Bethe}})\), this relation is straightforward to verify (next section).

A.3.1 Bethe Free Energy

For the periodic Ising grid, we can directly minimise the Bethe free energy functional \(G_{\text{Bethe}}\) (equation (2.31)) and compare it to the value \(F_B\) that is obtained in the traditional Bethe-Peierls approximation (see above). In the paramagnetic phase, the calculations are particularly simple. We utilise the fact that the minimum of \(G_{\text{Bethe}}\) is given by stationary points of BeP. Solving the fixed point equations for BeP (equation (2.9), yielding \(m_{\text{c}} = 0.5\) and respecting the normalisation and consistency constraints, the beliefs (2.29) take the simple
form $b_i(\pm 1) = 0.5$, whereas the pairwise beliefs (2.30) are given by

$$b_{ij}(1, 1) = b_{ij}(-1, -1) = \frac{e^{J/T}}{2 \cosh(J/T)} = b_i,$$  
$$b_{ij}(1, -1) = b_{ij}(-1, 1) = \frac{e^{-J/T}}{2 \cosh(J/T)} = b^*.$$  

(A.8)

(A.9)

Taking advantage of the full symmetry of periodic Ising grids without external field, $G_{Bethe}$ takes the simple form of a functional of general (but constrained) $a, a^*$ and $a_1$, i.e.,

$$G_{Bethe}(a, a^*, a_1) = U - TS \quad \text{with}$$  
$$U(a, a^*) = Nq(-Ja + Ja^*) \quad \text{and}$$  
$$S(a, a^*, a_1) = -Nq(a \ln a + a^* \ln a^*) + N(q - 1)a_1 \ln a_1.$$  

(A.10)

After inserting the values of $b, b^*$, $b_1$ for $a, a^*, a_1$ we find

$$U(a, a^*) = Nq(-Ja + Ja^*) = NqJ \frac{e^{-J/T} - e^{J/T}}{2 \cosh(J/T)}$$

$$= -NqJ \tanh(J/T)$$  

(A.11)

and

$$S(a, a^*, a_1) = -Nq(a \ln a + a^* \ln a^*) + N(q - 1)a_1 \ln a_1$$

$$= -Nq \left( \frac{e^{J/T}}{2 \cosh(J/T)} \ln(\frac{e^{J/T}}{2 \cosh(J/T)}) + \frac{e^{-J/T}}{2 \cosh(J/T)} \ln(\frac{e^{-J/T}}{2 \cosh(J/T)}) \right)$$

$$= -Nq \left( \frac{e^{J/T}}{2 \cosh(J/T)}(J/T - \ln(2 \cosh(J/T))) \right)$$

$$-Nq \left( \frac{e^{-J/T}}{2 \cosh(J/T)}(-J/T - \ln(2 \cosh(J/T))) - N(q - 1) \ln(2) \right)$$

$$= -Nq \left( \frac{2J \sinh(J/T)}{2T \cosh(J/T)} - \ln(2 \cosh(J/T)) \right) - N(q - 1) \ln(2)$$

$$= -Nq \frac{J}{T} \tanh(J/T) + Nq \ln(2 \cosh(J/T)) - N(q - 1) \ln(2).$$

Consequently,

$$\frac{U(a, a^*) - TS(a, a^*)}{N} = -Tq \ln(\cosh(J/T)) - T \ln(2) = -T \ln(2 \cosh^q(J/T)).$$  

(A.13)

Hence we have

$$\frac{1}{n} \min_{(a, a^*, a_1)} G_{Bethe} = \frac{1}{n} G_{Bethe}(b, b^*, b_1) = -T \ln(2 \cosh^q(J/T)).$$  

(A.14)

The value coincides with $F_B = -T \ln(Z_B/2^q)$ with

$$2^q(e^{h/T} \cosh^q(J + h_{eff}/T) + e^{-h/T} \cosh^q(J - h_{eff}/T)),$$  

(A.15)

where $h = 0$ and $h_{eff} = 0$ (paramagnetic phase). The factor $1/2^q$ has to be introduced to remove multiple counting.
A.4 Frobenius-Perron Theorem

The Frobenius-Perron theorem states that for an irreducible $n \times n$ matrix $(a_{ij})$ with $a_{ij} \geq 0$ and spectral radius $\rho$

(i) $\lambda = \rho > 0$,
(ii) there exists a corresponding eigenvector $(x_j)$ with $x_j > 0$,
(iii) $\lambda$ is a simple eigenvalue of $(a_{ij})$.
(iv) There are no nonnegative eigenvectors of $(a_{ij})$ except $(x_j)$.

The proof can be found in [52].

In the case of periodic Ising grids, the matrix $df$ (equation (2.36)) is positive semi-definite and irreducible. The former directly follows from (2.36). The latter is clear as for any two pairs of connections $(i_1 \rightarrow j_1)$ and $(i_2 \rightarrow j_2)$ we can find a path $(i_1 \rightarrow j_1), (l_1 \rightarrow i_1), (l_2 \rightarrow l_1), ... (j_2 \rightarrow l_k), (i_2 \rightarrow j_2)$, where $\partial n_{i_1,j_1}/\partial n_{l_1,i_1} = 1$ etc.
# List of Figures

1.1 Partition of the input space. ........................................... 7  
1.2 Optimal prediction performance. .................................... 9  
1.3 Feature space representation and macrostates. ....................... 14  
1.4 A robot equipped with two whisker-like sensors. ..................... 18  
1.5 Different description levels involved in concept learning. .......... 21  
1.6 Connectivity graphs. .................................................. 25  
1.7 Ersatz graph. ............................................................ 26  
1.8 Connectivity graph with non-injective symbol map. ................. 27  
1.9 Attractors of a Hopfield network. ................................. 32  
1.10 Winner-take-all network. .............................................. 33  
1.11 Initial state and final state. ........................................... 34  
1.12 A two-dimensional data distribution with three clusters. .......... 36  
1.13 Spike diagram without learning and with learning. .............. 37  
1.14 Structure of strong connections. .................................. 37  
2.1 Bayesian belief network. ............................................... 42  
2.2 Bayesian belief network, factor graph, Markov random field. .. 44  
2.3 The magnetisation $m$ as a function of $T$ and $w$ for the symmetric ferromagnetic model. .................................................. 49  
2.4 $g_1(n)$ for a) $J/T = 2$ (ferromagnetic phase) and b) $J/T = 1/4$ (paramagnetic phase). .................................................. 58  
2.5 An alternating initialisation leads to a cycle. ....................... 59  
2.6 Critical slowing down. .................................................. 61  
2.7 Stability of the diagonal $d$ and the cross-diagonal $d_c$. .......... 62  
2.8 Fractal basin boundaries. .............................................. 63  
3.1 k-means algorithm. ..................................................... 68  
3.2 A simple two-dimensional toy system. ................................ 75  
3.3 A comparison of the results for BeP and SW. ...................... 75  
3.4 Toy system with 3 clusters of different densities. .................. 77  
3.5 Results for the toy system for SC. ................................... 77  
3.6 Binary tree structures resulting from SSC. .......................... 78  
3.7 Seven classes of chemical compounds. ................................ 79  
3.8 Clustering result for the chemical data set achieved by SC. ....... 80
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.9</td>
<td>SSC results for chemical test set.</td>
<td>82</td>
</tr>
<tr>
<td>3.10</td>
<td>Model system with 3 clusters that are pairwise connected through one bond.</td>
<td>83</td>
</tr>
<tr>
<td>3.11</td>
<td>Typical clustering situation.</td>
<td>84</td>
</tr>
<tr>
<td>3.12</td>
<td>Different objects on a noisy background.</td>
<td>85</td>
</tr>
<tr>
<td>3.13</td>
<td>Visual scene analysis results.</td>
<td>86</td>
</tr>
<tr>
<td>3.14</td>
<td>A set of 241 wave traces from extracellular recordings of a single electrode.</td>
<td>87</td>
</tr>
<tr>
<td>3.15</td>
<td>An unsorted set of artificial data and the sorting solution provided by SSC.</td>
<td>89</td>
</tr>
<tr>
<td>3.16</td>
<td>Spike sorting results.</td>
<td>90</td>
</tr>
<tr>
<td>3.17</td>
<td>Clustering of multi-train data.</td>
<td>92</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Table of different clustering algorithms. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
3.2 Comparison between Ward clustering and SSC for the chemical test set. . . . 80
Bibliography


## Index

### A

- attractor ........................................ 31 – 34, 38 ff, 46, 48 ff, 96

### B

- Bayes’ theorem ................................. 41
- Bayesian belief network ...................... 41 – 44, 52
- belief propagation ... 3, 41 f, 44, 50, 53, 55, 73
- Boltzmann distribution ....................... 12, 45, 70, 73, 93

### C

- category ........................................... 2, 4, 8 ff, 19 – 22, 30, 65
- chemoinformatics .............................. 65, 79
- classification ................................. 1 ff, 8, 10, 17, 29 ff, 40, 91
- cluster stability .................................. 76 ff, 83 ff, 91, 93
- clustering algorithm ........................... 2, 4 ff, 10 f, 13, 19, 29 f, 38, 51 ff, 62 f, 65 f, 69, 71 f, 87, 91
- clustering hierarchy ............................. 66
- cognition ......................................... 2, 6, 15 ff, 40
- complexity ....................................... 6, 8, 13, 26
- computation ................................. 2 – 6, 8 f, 13 – 18, 22 f, 25, 28, 31, 38 – 42, 45, 50 f, 73
- concept acquisition ......................... 2, 18, 28 – 31, 40
- Connectionism .................................. 17 f, 28
- cost function ................................. 4 f, 10 ff, 67 – 70, 73
- critical slowing down ........................... 52, 61 – 64, 73

### D

- delay lines ........................................... 31
- deterministic annealing ...................... 67, 69 ff

### E

- entropy ........................................ 6 f, 12, 54, 73
- expectation maximisation .................... 69 f

### F

- feature space .................................. 6 f, 14 f, 18 – 22, 88, 93
- feedback ......................................... 2, 10, 38, 65, 93
- formal language .............................. 2, 3, 25
- formal reasoning ............................. 3, 17, 22, 24, 38
- fractal basin boundaries ...................... 64
- free energy .................................. 50, 54, 56 f, 73, 98
- Frobenius–Perron theorem .................. 56, 58, 100
- fuzzy clustering .............................. 65 f, 68

### G

- Gödel’s theorem .............................. 40
- Gaussian distribution ...................... 69 ff
- grandmother cell .......................... 34 ff
- graphical model ............................... 42 f

### H

- Hamiltonian ................ ...................... 45, 53, 72 ff, 81, 97
- Hebbian learning (Hebb D.O.) ................ 31 f, 36, 71
- Hopfield model (Hopfield J.J.) ........... 4, 31 ff, 39, 42 f, 45 f, 48 – 51

### I

- inference ...................................... 3, 22 f, 28, 42, 45 f, 49 – 52, 63
- information bottleneck method ............ 5, 10 f, 67 f
- integrate-and-fire model .................... 35
- Ising: system, grid, spin .................. 4, 33, 44 f, 50, 52 f, 55 ff, 59 – 64, 74, 81, 98 ff

### K

- k-means ....................................... 30, 66 – 70, 73, 88

### L

- Lagrangian ......................................... 69 f
INDEX

learning 5, 8 ff, 16 f, 19, 21, 29 – 33, 36 – 40, 51, 67, 71, 96 f
Lempel-Ziv distance ......................... 91
liquid state machine ....................... 30

M

Markov random field ... 42 – 46, 48, 50 – 53
McCulloch-Pitts model ................. 39
message passing ...................... 3, 51 ff
Monte Carlo simulation .............. 52, 73

N

natural cluster ............... 10, 66, 71 f, 76, 78, 93
neural code ................. 23
neural network 3, 17, 28, 30, 42 f, 46 f, 50, 72
neurodynamics ................. 45
noise .................... 13, 20, 23, 30, 33 f, 51, 87, 93
non-symbolic processes 3 f, 15 – 18, 22 ff, 28, 38 – 41, 43, 50 f
normalised cut .................. 67

P

perceptron .................. 29
phase locking .................. 30
phase transition .......... 52, 55 f, 61, 63, 72, 98
Potts spin .................. 53, 72, 74, 76, 85, 93
principal component analysis .... 88

R

read-out neuron ............... 30

S

scene analysis ............... 65, 84, 87, 93
self-organisation 3 ff, 10, 28, 30 f, 35, 38, 67 f, 71, 73, 93
self-organising map .......... 31, 71
sequential clustering .......... 74, 77, 83, 93
similar-property principle ...... 10, 94
similarity measure ... 10, 65 f, 71, 79, 81 f, 91
spike sorting .................. 85, 87 f, 90
spike train .................. 91
Stoop R .................. 4 ff, 8, 16, 93

superparamagnetic clustering ... 4, 38, 51 ff, 62 f, 67, 71 f, 74, 93
Swendsen-Wang algorithm ............ 73
symbolic reasoning 15 ff, 21 f, 24, 28, 30, 38 f

token ..................... 19 – 23, 30, 39
Turing A .................. 3, 13 – 16, 23, 38, 40

V

evaluation ...................... 10, 38, 65 f

W

winner-take-all network .......... 32, 34
Acknowledgements

Many people have contributed to my work in different ways. I would like to express my gratitude to all of them. My special thanks go to my PhD supervisor PD Dr. Ruedi Stoop who gave me the opportunity to conduct this work and who was a great source of inspiration for me. I am indebted to Prof. Willi-Hans Steeb for being my host in South Africa. Furthermore, I would like to thank all members and affiliated members of the Stoop group. They all contributed substantially to my work and it was a pleasure to be their companion. Therefore, thousand thanks to (alphabetical order): Dr. Markus Christen (in particular for his collaboration on spike train clustering), Christian Heid, Dr. Albert Kern (in particular for his wonderful programming help), Stefan Martignoli, Martin Schüle, Norbert Stoop (in particular for his collaboration on belief propagation), Dr. Yoko Uwate and Dr. Jan-Jan van der Vyver. I would like to thank all semester and diploma students with whom I had the pleasure to work, which was of great personal benefit. I would like to thank Dr. Justin Dauwels who happens to be the one who propagated the belief in belief propagation. I am very grateful for having had the opportunity to do my PhD at INI headed by Prof. Rodney Douglas and Prof. Kevan Martin. I would also like to thank all the other people of the institute for their companion and for their help and helpful discussions. Furthermore, I would like to thank Prof. Sebastiano Stramaglia for reading my thesis and acting as a referee. Finally, it is my special duty and even more so my most sincere pleasure to thank my family, my girlfriend Sarah and my friends for being here for and with me.
List of Publications

Peer-reviewed Journals:


Peer-reviewed Conference Proceedings:


Curriculum Vitae

Thomas Ott
Born on April 3, 1977,
in Schaffhausen, Switzerland

Education

1984-1992  Primary and secondary school in Schaffhausen
1992-1997  Gymnasium in Schaffhausen
1997-2003  Physics studies at ETH Zürich
2003-2007  Ph.D. student at Institute of Neuroinformatics UZH/ETH Zürich

Professional Experience

1999-2003  Teaching Assistant (Hilfsassistent) at Mathematics Department ETH Zürich
2000-2001  Teacher (Viktariat) at Gymnasium Freudenberg, Zürich
2003       Guest Researcher at University of Johannesburg, South Africa
2003-2006  Teaching Assistant, Physics Department ETH Zürich
Since 2007  Dozent (lecturer) and researcher at ZHAW, Zurich University of Applied Sciences
The Brainform Dissertation series publishes research-based dissertations, habilitations, diploma theses, project reports etc. in neuroscience and related fields. The intention is to provide a source for up-to-date research in a timely way. The series is multidisciplinary and publishes dissertations from all methodological perspectives with a substantial mathematical background and a thematic focus on information processing in biological systems.

Managing editor: Markus Christen
Editorial board: Willi-Hans Steeb, Ruedi Stoop

Published contributions:

1. **M. Christen:** The role of spike patterns in neuronal information processing. A historically embedded conceptual clarification. ETH Zürich, 2006.

2. **T. Ott:** Self-organised clustering as a basis for cognition and machine intelligence. ETH Zürich, 2007.