Chapter 1

Introduction

Data explosion of recent years has promoted artificial intelligence to an important stature in a variety of applications. Studies estimate that in three recent years, more data are generated than during their preceding human history (Lyman and Varian 2003). The processing of the increasingly available data require advanced techniques in, among others, data mining (Han and Kamber 2005), pattern recognition (Duda et al. 2000) and machine learning (Mitchell 1997). Even with great progresses in these fields, this remains a highly challenging task. A well-funded framework to optimize existing methods and develop novel schemes is necessary to produce faster, more efficient data analysis.

The field of machine learning concerns the design of algorithms to enable an adaptive system to learn from the arrival of information and recognize complex patterns from a subset of examples. A concise history of machine learning can be found in, e.g. (Duda et al. 2000, Mitchell 1997, Ripley 1996). Machine learning can be broadly categorized into supervised learning, unsupervised learning and reinforcement learning. Supervised learning involves an external supervision providing correct responses to given inputs and aim to maximize generalization of novel data. Unsupervised learning, on the other hand, works without supervision with a common goal of discovering underlying structures within the data. Reinforcement learning is concerned with finding suitable actions to maximize long-term rewards, see e.g. (Sutton and Barto 1998). The analysis of pattern involves a number of tasks, including data representation, classification, clustering, density estimation, regression and feature extraction, to name a few.

We concentrate on supervised learning and unsupervised learning for mainly two objectives: representation, which focuses on identifying the characteristic features of the data, and classification, which deals with discriminant analysis to assign unknown data into discrete categories with high accuracy. One of the simplest pattern analysis for these tasks is to compare the unknown data to all known reference patterns on the basis of a similarity measure, known as Nearest Neighbor (NN) methods (Duda et al. 2000). Obviously, the required computation scales immediately with the number of known patterns, therefore it is useful to employ representations of data subsets from each class. Hence, the subsets are quantized into a
number of prototype vectors. In this thesis, we focus on the theoretical aspects of one particular implementation, termed Learning Vector Quantization (LVQ).

1.1 Learning Vector Quantization

Learning Vector Quantization constitutes a simple yet powerful family of prototype-based machine learning algorithms, originally proposed by Kohonen in (Kohonen 1997). In its general form, a number of designated labeled prototypes is adapted to input examples to simultaneously provide a proper representation of the data and good classification by means of Nearest Prototype Classification (NPC). Originally, these prototype-based methods are devised for unsupervised learning, termed simply Vector Quantization (VQ). Similar techniques are later developed to incorporate supervised learning problems, which are called Learning Vector Quantization (LVQ).

The use of prototypes in the same feature space of the data is a particularly attractive feature which facilitates a straightforward interpretation of the data, especially to researchers outside the field of machine learning. This is in contrast to, for instance, the black-box approach of feed-forward neural networks or Support Vector Machines. The success of LVQ learning schemes are well documented in the fields of spectral analysis, medical data and gene expression (Bojér et al. 2003, Kuncheva 2004, Schleif et al. 2006, Villmann et al. 2003) among others. An extensive, up-to-date bibliography on LVQ systems are managed in (Neural Networks Research Centre, Helsinki 2002).

LVQ methods are fast, easy to use and well equipped for multi-class classification problems. Despite its apparent simplicity, LVQ is particularly flexible and can be adapted to specific problems with increasing complexity. The adaptation prescription vary greatly and can be modified into various algorithms. Numerous variants have been proposed, including LVQ 1, LVQ 2.1, LVQ 3, Optimized LVQ1 (Kohonen 1990), Generalized LVQ (Sato and Yamada 1995) and Soft LVQ (See and Obermayer 2003). Several algorithms focus on discovering distance metrics with highly discriminative orientations, including Distinction Selective LVQ (Pregenzer et al. 1996), Relevance LVQ (Bojer et al. 2001), Generalized Relevance LVQ (Hammer and Villmann 2002) and Matrix Relevance LVQ (Schneider et al. 2009). The framework of unsupervised prototype-based models can also be derived into popular many-prototype methods such as the popular Self Organising Maps (SOM) and Neural Gas methods (NG) which allow topologically correct representations of the high dimension data. Many modifications allow the LVQ systems to achieve performances comparable to, or better than, many other machine learning techniques, see
for example a comparative study of LVQ and Support Vector Machines (SVM) can be found in (Hammer et al. 2004).

Despite its wide range of applications, the theoretical understanding of LVQ in general remains very limited. In its original form, LVQ 1 was developed by heuristic methods, unlike the statistical flavor of other machine learning methods such as Support Vector Machines and related theories (Cortes and Vapnik 1995, Vapnik 1995). Several approaches are developed in order to provide a mathematical understanding of LVQ, such as constructing algorithms from well-chosen cost functions (Sato and Yamada 1995, Seo and Obermayer 2003) or viewing LVQ systems as Large Margin theorems (Crammer et al. 2002), among others. Nevertheless, a variety of LVQ algorithms remain heuristic and their properties such as convergence speed, stability and overall performance are unknown. A theoretical study would provide great insights which would be highly beneficial to the development of novel, efficient learning schemes.

1.2 Statistical mechanics of learning

A statistical mechanics approach to learning from examples was first proposed to study the generalization ability of Boolean learning networks in (Carnevali and Patarnello 1987) and further elaborated in (Levin et al. 1990, Tishby et al. 1989). Since then, this framework has been applied with success to investigate various machine learning techniques, including perceptrons e.g. (del Giudice et al. 1989, Hansel and Sompolinsky 1990), feed forward neural networks e.g. (Engel and van den Broeck 2001, Seung et al. 1992). Furthermore, the treatment has been extended to prototype based systems, e.g. (Biehl et al. 2007, Engel and van den Broeck 2001, Saad 1999).

Along the lines of statistical mechanics developed by Maxwell, Gibbs and Boltzmann, see e.g. (Huang 1987), macroscopic thermodynamical laws are deduced from microscopic dynamical properties of the system. In parallel with physical system which study properties such as work, heat, energy and pressure, we analyse learning systems in terms of energy, quantization error, generalization error and entropy, among others. The system can be fully described in terms of a set of quantities which characterizes the system, which is referred to as order parameters in the statistical mechanics framework. It is shown later that from the order parameters, it is possible to derive any relevant quantities of the system.

A key ingredient is the high degree of freedom in the system, so-called the thermodynamic limit, where the number of $N$ degrees of freedom tends to infinity. A remarkable property in this limit is that the probability distribution of a relevant quantity becomes sharply peaked around its maximal values. The approach of sta-
stistical mechanics allows exact description of the typical learning behavior.

Note that this approach is different from the probably almost correct (PAC) notion (Valiant 1984) which are highly applied in many mathematical learning problems. In the general PAC concept, strict bounds on the achievable learning are studied by choosing the worst possible combinations of the target rule and student complexity (Baum and Haussler 1989, Valiant 1984). However PAC is also overly pessimistic because the typical learning behaviors are often substantially better than the worst case scenario. Additionally, most PAC studies are limited to realizable tasks which can be perfectly obtained by the learning system, whereas many real world problems are unrealizable tasks which can only be approximated by the system.

Furthermore learning can be done in two general fashions: on-line learning and off-line learning. In on-line learning, training examples are presented sequentially and the prototypes are adapted to each example. Under a guiding cost function, this is equivalent to a stochastic gradient descent procedures (Bottou 1991). Meanwhile, off-line learning minimizes a properly constructed cost function, for instance by a gradient descent. Obviously, the selection of the cost function is highly relevant to the success of training. The statistical mechanics analyses treat on-line and off-line learning differently in many respects.

In the on-line learning analysis, we study the dynamics of the learning process with respect to learning time. Learning steps after each presentation of examples can be written as recursive relations of the order parameters. By capitalizing on the thermodynamic limit, we can remove the stochasticity from the recursion and describe the learning process as a set of ordinary differential equations of order parameters. This framework was applied with success to, e.g., perceptrons, feedforward neural networks, principal component analysis (Biehl and Caticha 2003, Engel and van den Broeck 2001, Saad 1999, Watkins et al. 1993).

This approach allows rigorous study of various LVQ algorithms with respect to training parameters and initialization of prototypes. We can compare learning curves and convergence speed to identify efficient training schemes. Given long learning times and fixed point analysis, we can also inspect the asymptotic properties of the algorithm.

In the off-line learning analysis, we study at the equilibrium state of the system with a fixed training set. We interpret learning as a stochastic minimization of the cost function under noise presented as the training temperature. A seminal work on this approach was applied to perceptrons, neural networks in (Seung et al. 1992). This approach generated a great interest leading to applications in (Biehl and Caticha 2003, Engel and van den Broeck 2001, Seung et al. 1992, Watkins et al. 1993), among others.

The training temperature acts as a smoothing function over the jagged landscape
of the actual cost function, allowing for an easier examination of present local and
global minima. However, even with the smoothing function, such analysis tend
to be rather complicated due to the valleys of the cost function and requires more
advanced methods used in statistical mechanics, such as replica method or cavity
method (Mezard et al. 1987), in order to understand the process at finite learning
temperatures. An approximation which has been found to be extremely useful as a
good predictor of these analyses is the so-called annealed approximation (Biehl and
Caticha 2003, Seung et al. 1992, Solla and Levin 1992). Both annealed approxima-
tion and the replica method yield the same exact result in the limit of high training
temperatures.

1.3 Thesis organization

Here we briefly address the topics of the chapters of this thesis. The material is
mainly composed of two parts: the analysis of on-line and off-line learning systems.
The former part spans from Chapters 3 to 4, while the latter part is explained in
Chapters 5 and 6.

Chapter 2 briefly presents the learning model and generic LVQ algorithms to be
studied throughout the thesis. The extents of the input density models and their
relevance in practical situations are discussed.

Chapter 3 explains in detail the analysis of on-line VQ and LVQ systems. We
describe the learning dynamics as ordinary differential equations in terms of the
order parameters and obtain results for both transient and asymptotic properties.
In particular, this chapter focuses on unsupervised learning using VQ and Neural
Gas (NG). We compare the convergence speed and asymptotic properties of Winner-
Takes-All methods to rank-based Neural Gas and demonstrate robustness of NG
under various initial conditions.

Chapter 4 deals with on-line supervised learning using multiple prototype LVQ
algorithms, where we investigate the influence of additional competing prototypes
to fine tune the decision boundary. We analyse and compare various LVQ prescrip-
tions, and the effectiveness of a window-based approach which selects examples
during learning.

Chapter 5 presents the detailed analysis of off-line learning for VQ and LVQ
cost functions along the lines of equilibrium physics. In this chapter, we analyse
unsupervised VQ and NG scenarios based on the high training temperature limit
and the annealed approximation. We demonstrate the phase transition behavior in
the learning curves which depend on the training set size and temperature.

Chapter 6 extends the analysis of high temperature off-line learning to super-
vised methods. We compare the performances of cost-function based LVQ algorithms depending on the size of the training set under the influence of an additional weight decay.

A summary of all chapters is written in chapter 7. The chapter also discusses the relation of on-line and off-line learning and the outlook of further statistical physics analysis of prototype-based learning systems.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>rescaled number of examples $\alpha = P/N$</td>
</tr>
<tr>
<td>$\tilde{\alpha}$</td>
<td>number of examples rescaled with temperature $\tilde{\alpha} = \beta\alpha$</td>
</tr>
<tr>
<td>$\hat{\alpha}$</td>
<td>number of examples rescaled with learning rate $\hat{\alpha} = \eta\alpha$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>inverse training temperature</td>
</tr>
<tr>
<td>$\mathbf{B}$</td>
<td>data cluster centers</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>weight decay parameter</td>
</tr>
<tr>
<td>$\delta$</td>
<td>window parameter</td>
</tr>
<tr>
<td>$\ell_{\sigma}$</td>
<td>separation between cluster centers</td>
</tr>
<tr>
<td>$\epsilon_g$</td>
<td>generalization error</td>
</tr>
<tr>
<td>$\epsilon_t$</td>
<td>training error</td>
</tr>
<tr>
<td>$\eta$</td>
<td>learning rate</td>
</tr>
<tr>
<td>$\mathcal{D}$</td>
<td>training data set</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>rank parameter for neural gas</td>
</tr>
<tr>
<td>$\mu$</td>
<td>example index</td>
</tr>
<tr>
<td>$\sigma^\mu$</td>
<td>cluster generating the $\mu$-th example</td>
</tr>
<tr>
<td>$\Theta$</td>
<td>Heaviside function</td>
</tr>
<tr>
<td>$v_G$</td>
<td>softness parameter for GLVQ</td>
</tr>
</tbody>
</table>
$V_{soft}$  softness parameter for RSLVQ
$
W$
set of prototypes
$
w_s$
$s$-th prototypes
$
\xi^\mu$
$\mu$-th example
$b_s$
projection of examples to cluster centers $h_S = \xi \cdot B_s$
$C$
covariance matrix
$c_s$
class of $s$-th prototypes
$d$
squared Euclidean distance
$E$
quantization error
$F$
quenched free energy
$f$
free energy function
$H$
cost function
$h_S$
projection of examples to prototypes $h_S = \xi \cdot w_S$
$K$
number of prototypes
$L$
likelihood ratio of RSLVQ mixture model
$M$
number of clusters
$N$
dimensionality of data
$P$
number of examples
$P(\xi)$probability density of data
$Q_{ST}$order parameter $Q_{S\sigma} = w_S \cdot w_T$
$r_S$
rank of $w_S$
$R_{S\sigma}$order parameter $R_{S\sigma} = w_S \cdot B_\sigma$
$T$
training temperature
$Y$
number of classes
$y$
class of example
$Z$
partition function
Chapter 2

Model and Algorithms

Abstract

In this chapter we shortly present the learning model and general notations to be used throughout the thesis. A key ingredient of the model involves very high dimensional input data to allow analysis in the so-called thermodynamic limit. The input densities to be considered can be extended to a mixture of an arbitrary number of isotropic homogenous clusters. The relevance of this model with regard to practical situations is discussed.

We investigate two types of problems in this thesis: unsupervised learning and supervised learning. For a supervised problem, we are given a data set \( \mathcal{D} \) consisting of \( P \) pairs of data points

\[
\mathcal{D} = \{ (\xi^\mu, y^\mu) \}_{\mu=1}^{P} \quad \text{with} \quad \xi^\mu \in \mathbb{R}^N, \quad y^\mu = \{1, 2, \ldots, Y\},
\]

where \( \xi^\mu \) is the input data vector in \( N \)-dimensional space and \( y^\mu \) is the single class label of a particular example with index \( \mu \). \( Y \) represents the number of unique classes of the data set. For an unsupervised problem, \( y^\mu \) is not present or ignored.

A nearest prototype classifier consists of a set of \( K \) labeled prototype vectors

\[
\mathbf{W} = \{ (\mathbf{w}_S, c_S) \}_{S=1}^{K} \quad \text{with} \quad \mathbf{w}_S \in \mathbb{R}^N, c_S = \{1, 2, \ldots, Y\},
\]

where \( \mathbf{w}_S \) is the prototype vector in the same \( N \)-dimensional space and carries the label \( c_S \). Each class in the data set is represented by one or more prototypes. In unsupervised VQ, the class labels \( c_S \) are ignored.

2.1 Algorithms

We shortly review LVQ algorithms while referring to the actual algorithms in detail in the following chapters. In on-line learning schemes, examples are presented
Figure 2.1: Visualization of a nearest prototype classification scheme using Euclidean distances, producing piecewise linear decision boundaries. Here three classes are present ($\triangle$, $\Box$, $\circ$), each class represented with two prototypes.

sequentially to the system and the prototypes are adapted by the following update step

$$w_S^\mu = w_S^{\mu-1} + \frac{\eta}{N} [d_1^\mu, \ldots, d_K^\mu, y^\mu, \ldots] \left( \xi^\mu - w_S^{\mu-1} \right),$$  \hspace{1cm} (2.3)

where $w_S^\mu$ denotes the prototype after presentation of $\mu$ examples and the learning rate $\eta$ is rescaled with $N$. We use the shorthand $f_S$ for the modulation function which controls, along with the learning rate $\eta$, the magnitude of the update of $w_S$ towards or away from the current example. For example, in LVQ 1 we define $f_S = +1(f_S = -1)$ if $w_S$ is the closest prototype and has the same (different) label as the example, $y^\mu$. We will discuss the actual training algorithms in detail in the following chapters.

Classification is implemented through a nearest prototype scheme: novel data will be assigned to the class of the closest prototype according to a dissimilarity measure. An example is displayed in Fig. 2.1. Typically, the number of prototypes is larger than the number of classes, such that every classification boundary is determined by more than one prototype. A popular choice for the dissimilarity measure is the squared Euclidean distance

$$d_S^\mu = (\xi^\mu - w_S)^2.$$  \hspace{1cm} (2.4)

Obviously, the selection of an appropriate dissimilarity measure is critical to successful learning. Otherwise, the presence of noise from irrelevant dimensions can
interfere with the relevant dimensions for classification. The adaptation of the dis-
similarity measure using automatic weighting factors to overcome this problem has
been the subject of various studies and several algorithms have been proposed,
such as Distinction Selective LVQ (Pregenzer et al. 1996), Relevance LVQ (Bojer
et al. 2001) and Generalized Relevance LVQ (Hammer and Villmann 2002, Ham-
mer et al. 2003, Strickert et al. 2006). Furthermore, combinations of features can be
examined together by employing matrix based relevance learning, similar to Ma-
halanobis distances (Schneider et al. 2009). However, in this thesis, we restrict the
dissimilarity measure to the squared Euclidean distance which allows for important
simplifications to the following analyses.

2.2 Model

Throughout the thesis, we study LVQ algorithms in a model situation: high dimen-
sional data are generated from a mixture of Gaussian clusters and presented to a
system of two or three prototypes. We restrict to isotropic and homogeneous clus-
ters, i.e. each cluster labeled \( \sigma = \{1, \ldots, M\} \) generates only data with one class label
\( y_\sigma = \{1, 2, \ldots, Y\} \). For various analyses in this thesis with only two clusters, we
use \( \sigma = y_\sigma = \pm 1 \) for simplification. Examples \( \xi \in \mathbb{R}^N \) are drawn independently
according to the probability density function

\[
P(\xi) = \sum_{\sigma=1}^{M} p_\sigma P(\xi|\sigma) \quad \text{with} \quad P(\xi|\sigma) = \frac{1}{(2\pi v_\sigma)^{N/2}} \exp \left[ -\frac{1}{2v_\sigma} (\xi - \ell_\sigma B_\sigma)^2 \right],
\]

where \( p_\sigma \) are the prior probabilities and \( \sum_{\sigma=1}^{M} p_\sigma = 1 \). The components of vectors \( \xi^\mu \)
from cluster \( \sigma \) are random numbers according to a Gaussian distribution with mean
vectors \( \ell_\sigma B_\sigma \) and variance \( v_\sigma \). The parameter \( \ell_\sigma \) controls the separation between the
mean vectors. In most cases encountered in the thesis, \( B_\sigma \) are orthonormal, i.e. \( B_\sigma \cdot B_\rho = \delta_{\sigma\rho} \) are fulfilled. This model can be extended for mixtures of many Gaussian
clusters with \( M < N \) by choosing a coordinate system where the orthonormality
conditions \( B_\sigma \cdot B_\rho = \delta_{\sigma\rho} \) are fulfilled. Similar densities have been extensively studi-
in, e.g. (Barkai et al. 1993, Biehl 1994, Biehl and Mietzner 1994, Biehl et al. 2007, Engel

We denote averages over the probability density \( P(\xi) \) as \( \langle \cdots \rangle \) and the condi-
tional averages over \( P(\xi) \) as \( \langle \cdots \rangle_\sigma \). The average length of data vectors \( \langle \xi_\sigma^2 \rangle \)
are calculated to be

\[
\langle \xi^2 \rangle = \sum_{\sigma=1}^{M} p_\sigma \langle \xi^2 \rangle_\sigma \quad \text{with} \quad \langle \xi^2 \rangle_\sigma = \sum_{j=1}^{N} \langle (\xi_j)^2 \rangle = \sum_{j=1}^{N} (v_\sigma + \langle (\xi_j)^2 \rangle) = v_\sigma N + \ell_\sigma^2.
\]
2. Model and Algorithms

Figure 2.2: Sample data generated by the model, here with two clusters \( M = 2, y_c = \{1, 2\}, N = 100, p_1 = 0.5, \ell_1 = \ell_2 = 3, v_1 = v_2 = 1, \) as projected on two-dimensional subspaces. The clusters separate in the projection to the plane spanned by cluster center vectors \( \mathbf{B}_1, \mathbf{B}_2 \) (left panel), but completely overlap in the projection to a plane composed of two arbitrary unit vectors \( \{\mathbf{v}_1, \mathbf{v}_2\} \) (right panel).

In this framework we formally exploit the thermodynamic limit \( N \to \infty \) corresponding to very high dimensional data. This has simplifying consequences which will be present throughout the paper, for instance to neglect the \( \ell_2^2 \)-terms in Eq. (2.6). Note that data from different classes have a strong overlap and the anisotropy is very weak in high dimensions: while the mean of cluster \( \sigma \), given by \( \ell_\sigma \mathbf{B}_\sigma \), is a vector of length \( O(1) \), the average length of the data vectors \( \langle \xi \rangle^2 \) is in the order \( O(N) \). The clusters become apparent only on a \( M \)-dimensional space spanned by \( \mathbf{B}_\sigma, \sigma = \{1, \ldots, M\} \) while they completely overlap on other subspaces. The non-trivial goal is to identify this subspace from the \( N \)-dimensional data. Our model provides a base for further analysis, e.g., it can easily be extended to \( M \)-isotropic Gaussian clusters and multi-class problems.

2.3 Relevance for practical learning

Obviously, this model is greatly simplified from practical situations. However it represents an ideal scenario to analyse the considered learning algorithms. The use of only few prototypes and few clusters is perfectly appropriate for such scenario, thus the problem of model selection is avoided.

While more complex behaviors are expected in practical applications, the non-trivial effects already observed in this model will clearly influence the outcome under more general circumstances. Our motivation is to understand in highly ideal-
ized scenarios. If it fails in such idealized scenarios, then one can expect even worse results in practical situations. In this analysis, we do not cover all possible learning behavior in practical situations. Instead, we show that some non-trivial effects are already present even in highly idealized situations. These effects are not restricted to the simplified model but will persist or remain unavoidable at even more complex scenarios. Therefore even though the model assumes strong simplifications, the findings should remain highly relevant in practice. We can expect that algorithms which do not perform well on this idealized scenario will also be inappropriate for more complex, real life scenarios.

Several restrictions can be justified to practical problems as follows. While the analysis so far is restricted to Gaussian data, this represents an ideal situation for Gaussian modeling of feature vectors, which is a common technique in many practical scenarios. Also, although formally we use infinite dimensionality, this analysis is already highly accurate at finite dimensions as low as $N = 200$, which is well within the domain of many practical problems.

In many real world applications of LVQ systems, a surprisingly small number of prototypes can already produce good performance, given other considerations, e.g. (Neural Networks Research Centre, Helsinki 2002, Hammer and Villmann 2002, Schneider et al. 2009). Therefore the two and three prototype analysis here already approaches practical learning behavior and already underlines important aspects of multi-prototype problems: the non-trivial competition between prototypes defining the decision boundary.
Chapter 3

Learning dynamics and robustness of vector quantization and neural gas

Abstract

Various alternatives have been developed to improve the Winner-Takes-All (WTA) mechanism in vector quantization, including the Neural Gas (NG). However, the behavior of these algorithms including their learning dynamics, robustness with respect to initialization, asymptotic results, etc. has only partially been studied in a rigorous mathematical analysis. The theory of on-line learning allows for an exact mathematical description of the training dynamics in model situations. We analyse both WTA and NG schemes using a system of three competing prototypes trained from a mixture of Gaussian clusters and demonstrate that the Neural Gas can improve convergence speed and achieves robustness to initial conditions. However, depending on the structure of the data, the Neural Gas does not always obtain the best asymptotic quantization error.

3.1 Introduction

Vector quantization (VQ) as developed by Kohonen in (Kohonen 1997) is an important unsupervised learning algorithm, widely used in different areas such as data mining, medical analysis, image compression and speech or handwriting recognition. An up-to-date repository of applications can be found in (Neural Networks Research Centre, Helsinki 2002). The main objective of VQ is to represent the data points by a small number of prototypes or codebook vectors, measured by a distortion or quantization error. This can directly be used for compression, clustering, data mining or (with post-labeling of the prototypes) classification (Gersho and Gray 1991, Jain et al. 1999).

The basic “winner-takes-all” (WTA) or batch algorithms such as the popular $k$-means clustering (Bottou and Bengio 1995) directly optimize the quantization error underlying vector quantization. However, these methods can easily be subject to confinement in local minima of the quantization error and produce suboptimal results. Consequently, the initialization of prototypes undesirably plays a critical role.
role in the success of training. A variety of alternatives to overcome this problem has been proposed, some of which are heuristically motivated while others are based on the minimization of a cost function related to the quantization error: the self-organizing map (SOM) (Kohonen 1997), fuzzy-k-means (Bezdek 1981), stochastic optimization (Buhmann 1998), to name just a few. These algorithms have in common that each pattern influences more than one prototype at a time through a “winner-takes-most” paradigm.

Neural gas (NG) as proposed in (Martinetz et al. 1993) is a particularly robust variation of vector quantization with the introduction of neighborhood relations. The NG system takes into account the relative distances between all prototypes and a given pattern and ranks the prototypes accordingly. The rank-based adaptation steps are hence affected directly by the data topology. This is contrast to the self-organizing map (Kohonen 1997) which utilizes a predefined lattice. While the NG procedure is potentially more expensive than SOM, it also reduces topology mismatches.

In practice, given proper choices of the learning parameters, NG algorithms yield better solutions than WTA; however, the effect of this strategy on convergence speed or asymptotic behavior has hardly been rigorously investigated so far.

Methods from statistical physics and the theory of on-line learning (Engel and van den Broeck 2001) allow for an exact mathematical description of learning systems for high dimensional data. In the limit of infinite dimensionality, such systems can be fully described in terms of a few characteristic quantities, the so-called order parameters. The evolution of these order parameters along the training procedure is characterized by a set of coupled ordinary differential equations (ODE). By integrating these ODEs, it is possible to analyse the performance of VQ algorithms in terms of stability, sensitivity to initial conditions, and achievable quantization error. This successful approach has also been reviewed in (Engel and van den Broeck 2001, Watkins et al. 1993), among others.

The extension of the theoretical analysis of simple (WTA-based) vector quantization with two prototypes and two clusters introduced in an earlier work (Biehl et al. 2006) is not straightforward. Additional prototypes and clusters introduce more complex interactions in the system that can result in radically different behaviors. Also, the mathematical treatment becomes more involved and requires, for instance, several numerical integrations. In this work we introduce an additional prototype and a mixture of clusters. We investigate not only WTA but also the popular Neural Gas approach (Martinetz et al. 1993) for vector quantization. This is an important step towards the investigation of general VQ approaches based on neighborhood interaction such as self-organizing maps.
3.2 Winner-Takes-All and Neural Gas

Assume a set of $P$ input data denoted by $\{\xi^\mu \in \mathbb{R}^N\}_{\mu=1}^P$, generated according to a given probability density function $P(\xi)$. In our work, we choose the input density to be a mixture of $M$ spherical Gaussian clusters, as written in Eq. (2.5):

$$P(\xi) = \sum_{\sigma=1}^M p_\sigma P(\xi|\sigma) = \sum_{\sigma=1}^M p_\sigma \frac{1}{(2\pi v_\sigma)^{N/2}} \exp\left(-\frac{1}{2v_\sigma}(\xi - \ell_\sigma B_\sigma)^2\right)$$

where $p_\sigma$ are the prior probabilities of each cluster. The components of vector $\xi^\mu$ are random numbers according to a Gaussian distribution with mean vectors $\ell_\sigma B_\sigma$ and variance $v_\sigma$. The mean vectors are orthogonal, i.e. $B_i \cdot B_j = \delta_{ij}$ where $\delta$ is the Kronecker delta. The parameters $\ell_\sigma$ describe the separation between the clusters. We refer to the details of this input density in Chapter 2.2.

Vector Quantization represents the input data in the same $N$-dim. space by a set of prototypes $W = \{w_S \in \mathbb{R}^N\}_{S=1}^K$. The primary goal of VQ is to find a faithful representation by minimizing the so-called quantization or distortion error

$$E(W) = \frac{1}{2} \sum_{\mu=1}^P \sum_{S=1}^K d(\xi^\mu, w_S) \prod_{T \neq S} \Theta_{ST} - \frac{1}{2} \sum_{\mu=1}^P (\xi^\mu)^2$$

where $\Theta_{ST} \equiv \Theta(d(\xi^\mu, w_S) - d(\xi^\mu, w_T))$. For each input vector $\xi^\mu$ the closest prototype $w_S$ is singled out by the product of Heaviside functions, $\Theta(x) = 0$ if $x < 0$; 1 else. Here we restrict ourselves to the quadratic Euclidean distance measure $d(\xi^\mu, w_S) = (\xi^\mu - w_S)^2$. The constant $\frac{1}{2} \sum_{\mu=1}^P (\xi^\mu)^2$ term is independent of prototype positions and is subtracted for convenience, this will be shown in Eq. (3.17).

The input data is presented sequentially during training and one or more prototypes are updated on-line. Algorithms studied here can be interpreted as stochastic gradient descent procedures with respect to a cost function $H(W)$ related to $E(W)$. The generalized form reads

$$H(W) = \frac{1}{2} \sum_{\mu=1}^P \sum_{S=1}^K d(\xi^\mu, w_S) f(r_S) - \frac{1}{2} \sum_{\mu=1}^P (\xi^\mu)^2$$

where $r_S$ is the rank of prototype $w_S$ with respect to the distance $d(\xi^\mu, w_S)$, i.e. $r_S = S - \sum_{T \neq S} \Theta_{ST}$. Rank $r_S = 1$ corresponds to the so-called winner, i.e. the prototype $w_j$ closest to the example $\xi^\mu$. The rank function $f(r_S)$ determines the update strength for the set of prototypes and satisfies the normalization $\sum_{S=1}^K f(r_S) = 1$; note that it does not depend explicitly on distances but only on the ordering of the prototypes with respect to the current example.
Stochastic gradient descent of $H(W)$ yields the online update rule

$$w^\mu_S = w^{\mu-1}_S + \frac{\eta}{N} f(r_S) (\xi^\mu - w^{\mu-1}_S), \quad (3.4)$$

where $\eta$ is the learning rate and $\xi^\mu$ is a single example drawn independently at time step $\mu$ of the sequential training process. We compare two different algorithms:

1. **Winner Takes All:**
   In this learning scheme only one prototype, the winner, is updated for each input. The cost function directly minimizes the quantization error with $H(W) = E(W)$. The corresponding rank function is

   $$f_{WTA}(r_S) = \prod_{T \neq S} \Theta_{ST}. \quad (3.5)$$

2. **Neural Gas:**
   The update strength decays exponentially with the rank controlled by a parameter $\lambda$. The rank function is $f(r_S) = \frac{1}{C(\lambda)} h_{\lambda}(r_S)$ where $h_{\lambda}(.) = \exp (-r_S/\lambda)$ and $C(\lambda) = \sum_{r_S=1}^{K} \exp (-r_S/\lambda)$ is a normalization constant. The parameter $\lambda$ is adjusted during training; it is frequently set large initially and decreased in the course of training. Note that for $\lambda \to 0$ the NG algorithm becomes identical with WTA. We decompose $f(r_S)$ according to its ranks as

   $$f_{NG}(r_S) = \frac{1}{C(\lambda)} \sum_{k=1}^{K} h_{\lambda}(k) g_S(k) \quad (3.6)$$

where $g_S(k) = 1$ if $r_S = k$; 0 else and $\sum_k g_S(k) = 1$. In a system of three prototypes, this can be written in terms of Heaviside functions, defined below Eq. (3.2):

$$g_S(1) = \prod_{T \neq S} \Theta_{ST}$$

$$g_S(2) = \sum_{U \neq S} \prod_{T \neq U,S} \Theta_{ST}(1 - \Theta_{SU})$$

$$g_S(3) = \prod_{T \neq S} (1 - \Theta_{ST}). \quad (3.7)$$
3.3 Analysis of learning dynamics

In this section we give a description of the theoretical framework for analysis of online LVQ training. Following the lines of the theory of online learning, e.g. (Biehl and Caticha 2003, Biehl et al. 2004, Biehl et al. 2007, Engel and van den Broeck 2001, Saad 1999), in the thermodynamic limit $N \to \infty$ the system can be fully described in terms of a few characteristic quantities, or so-called order parameters. A suitable set of characteristic quantities for the considered learning model is:

$$R_{S\sigma}^\mu = w_S^\mu \cdot B_{\sigma} \quad Q_{ST}^\mu = w_S^\mu \cdot w_T^\mu.$$  \hspace{1cm} (3.8)

Note that $R_{S\sigma}$ measure the projections of prototype vectors $w_S^\mu$ on the center vectors $B_{\sigma}$ and $Q_{ST}^\mu$ correspond to the self- and cross- overlaps of the prototype vectors.

From the generic update rule defined above, Eq. (3.4), we can derive the following recursions in terms of the order parameters:

$$\frac{R_{S\sigma}^\mu - R_{S\sigma}^{\mu-1}}{1/N} = \eta f(r_S)(b^\sigma - R_{S\sigma}^{\mu-1})$$

$$\frac{Q_{ST}^\mu - Q_{ST}^{\mu-1}}{1/N} = \eta [f(r_T)(h_S^\mu - Q_{ST}^{\mu-1}) + f(r_S)(h_T^\mu - Q_{ST}^{\mu-1})]$$

$$+ \eta^2 \frac{f(r_S)f(r_T)(\xi^\mu)^2}{N} + O\left(\frac{1}{N}\right)$$  \hspace{1cm} (3.9)

where the input data vectors $\xi^\mu$ enter the system only as their projections $h_S^\mu$ and $b_{\sigma}^\mu$, defined as

$$h_S^\mu = w_S^{\mu-1} \cdot \xi^\mu \quad b_{\sigma}^\mu = B_{\sigma} \cdot \xi^\mu.$$  \hspace{1cm} (3.10)

Note that the last two terms in Eq. (3.9) come from

$$\left(\frac{\eta}{N}\right)^2 f(r_S)f(r_T)(\xi^\mu - w_S^{\mu-1})(\xi^\mu - w_T^{\mu-1})$$

$$= \frac{\eta^2}{N} \frac{f(r_S)f(r_T)}{N} \left[ (\xi^\mu)^2 - h_S^\mu - h_T^\mu + Q_{ST}^{\mu-1} \right]$$  \hspace{1cm} (3.11)

where $(\xi^\mu)^2$ is the only term that scales with $N$.

In the limit $N \to \infty$, the $O(1/N)$ term can be neglected and the order parameters become self-averaging with respect to the random sequence of examples. This means that fluctuations of the order parameters vanish and the system dynamics can be described exactly in terms of their mean values. A mathematically rigorous analysis of this property is provided in (Reents and Urbanczik 1998), describing the necessary bounds of the magnitude of the learning steps.
We denote the average over the density $P(\xi)$ as $\langle \cdots \rangle = \sum_\sigma p_\sigma \langle \cdots \rangle_\sigma$, where $\langle \cdots \rangle_\sigma$ is the conditional average over $P(\xi | \sigma)$, and exploit the thermodynamic limit to find the following relation

$$
\lim_{N \to \infty} \frac{\langle f(r_S)f(r_T)\xi^2 \rangle}{N} = \lim_{N \to \infty} \frac{1}{N} \sum_\sigma p_\sigma \langle f(r_S)f(r_T)\langle \xi^2 \rangle_\sigma \rangle_\sigma = \sum_\sigma p_\sigma v_\sigma \langle f(r_S)f(r_T) \rangle_\sigma.
$$

(3.12)

Here we used the relation $\xi^2 = \langle \xi^2 \rangle_\sigma + \Gamma$, where $\Gamma$ is a stochastic quantity with $O(\Gamma) \ll N$ which can be neglected. Also, we calculated the conditional average of the input vector length $\langle \xi^2 \rangle_\sigma / N = (v_\sigma N + \ell^2_\sigma) / N$, see the derivations in Eq. (2.6).

Furthermore, for $N \to \infty$, we can conceive learning time as a continuous variable by using the rescaled quantity

$$
\alpha \equiv \mu / N.
$$

(3.13)

Here it is implied that successful training requires a number of examples which grows linearly with the number of dimensions. Accordingly, by combining Eqs. (3.9), (3.12) and rescaling with (3.13), the dynamics can be described by a set of coupled ODE after performing an average over the sequence of input data, see also (Biehl et al. 2004, Ghosh et al. 2006):

$$
\frac{dR_{S\sigma}}{d\tilde{\alpha}} = \eta \left( \langle b_\sigma f(r_S) \rangle - \langle f(r_S) \rangle R_{S\sigma} \right)

\frac{dQ_{ST}}{d\tilde{\alpha}} = \eta \left( \langle h_S f(r_T) \rangle - \langle f(r_T) \rangle Q_{ST} + \langle h_T f(r_S) \rangle - \langle f(r_S) \rangle Q_{ST} \right) + \eta^2 \sum_\sigma p_\sigma v_\sigma \langle f(r_S)f(r_T) \rangle_\sigma.
$$

(3.14)

In various sections in this thesis, we investigate learning behaviors using small learning rates $\eta \to 0$ and neglect the $\eta^2$ terms in Eq. (3.14). Non trivial behavior is only expected by rescaling the learning time while taking the simultaneous limits

$$
\eta \to 0, \ \alpha \to \infty, \ \tilde{\alpha} = \eta \alpha.
$$

In the limit of small learning rates, Eq. (3.14) is rescaled as

$$
\frac{dR_{S\sigma}}{d\tilde{\alpha}} = \langle b_\sigma f(r_S) \rangle - \langle f(r_S) \rangle R_{S\sigma}

\frac{dQ_{ST}}{d\tilde{\alpha}} = \langle h_S f(r_T) \rangle - \langle f(r_T) \rangle Q_{ST} + \langle h_T f(r_S) \rangle - \langle f(r_S) \rangle Q_{ST}.
$$

(3.15)
3.3. Analysis of learning dynamics

Exploiting the limit $N \to \infty$ once more, the quantities $h^\mu_{Si}, b^\mu_\sigma$ in Eq. (3.14) or Eq. (3.15) become correlated Gaussian quantities by means of the Central Limit Theorem. Therefore, they are fully specified by first and second moments, detailed in Appendix A.

$$
\langle h^\mu_{Si}\rangle = \ell_\sigma R^\mu_{Si}^{-1}, \quad \langle b^\mu_\sigma \rangle = \ell_\sigma \delta_{\tau \sigma}, \quad \langle h^\mu_{Ti}\rangle - \langle h^\mu_{Si}\rangle = v_\sigma Q^\mu_{ST}^{-1} \langle b^\mu_\sigma \rangle - \langle b^\mu_\sigma \rangle \quad \langle h^\mu_{Ti}\rangle - \langle h^\mu_{Si}\rangle = v_\sigma R^\mu_{Ti}^{-1}.
$$

where $S, T$ are prototype indices, $\tau, \rho, \sigma$ are cluster indices, $\delta$ is the Kronecker delta and $T_{\tau \rho} \equiv B_\tau \cdot B_\rho$ is an overlap measure between clusters. For orthonormal $B_\sigma$ vectors, we can use $T_{\tau \rho} = 1$, if $\tau = \rho$; 0 else.

Given the averages for a specific rank function $f(r_S)$, cf. Eqs. (B.38) and (B.46) we obtain a closed form expression of ODE. Using the initial conditions $R_{Si}(0), Q_{ST}(0)$, we integrate this system for a given algorithm and get the evolution of order parameters in the course of training, $R_{Si}(\alpha), Q_{ST}(\alpha)$. The behavior of the system depends on the characteristic of the data and the parameters of the learning scheme, i.e. offset of the clusters $\ell_\sigma$, variance within the clusters $v_\sigma$, learning rate $\eta$, and for NG, the rank function parameter $\lambda$. This method of analysis is in good agreement with large scale Monte Carlo simulations of the same learning systems for dimensionality as low as $N = 200$, see e.g. in (Biehl et al. 2007).

Analogously, the average quantization error, Eq. (3.2), over the probability density expressed in terms of order parameters

$$
E(W) = \frac{1}{2} \sum_{S=1}^{K} \left( \prod_{T \neq S} \Theta_{ST} \right) Q_{SS} - \sum_{S=1}^{K} \left( h^\mu_{Si} \prod_{T \neq S} \Theta_{ST} \right)
$$

Note that $E(W)$ does not depend explicitly on $\xi$; here it is shown how the subtracted constant term described in Eq. (3.2) and Eq. (3.3) becomes useful. $E(W)$ is fully expressed in terms of order parameters. For instance, in two prototype systems $W = \{w_1, w_2\}$ we can calculate the above quantity $E(W) = \sum_\sigma P_\sigma E_\sigma(W)$ as follows:

$$
E_\sigma(W) = -\frac{\sqrt{v_{\sigma}} \Delta Q}{\sqrt{2\pi}} \exp\left(-\frac{Z^2}{2}\right) + \left(\frac{Q_{11}}{2} - \ell_\sigma R_{1\sigma}\right) \Phi(Z) + \left(\frac{Q_{22}}{2} - \ell_\sigma R_{2\sigma}\right) \Phi(-Z),
$$

with

$$
\Delta Q = \sqrt{Q_{11} - 2Q_{12} + Q_{22}},
$$

$$
Z = \left(2\tau_\sigma (R_{1\sigma} - R_{2\sigma}) - Q_{11} + Q_{22}\right) / (2\sqrt{\tau_\sigma \Delta Q}).
$$

We refer to the details of the calculations in Appendix B.4. The form of $E_\sigma(W)$ for systems with more prototypes is more involved, and requires numerical integrations over a $(K - 2)$ subspace. Plugging in the values of the order parameters $\{R_{Si}(\alpha), Q_{ST}(\alpha)\}$, we can study the so-called learning curve $E(W)$ in dependence of the training time $\alpha$ for a given VQ algorithm.
3.4 Results

3.4.1 Learning Dynamics

We study the performance of both WTA and NG in several cases using three prototypes and up to three clusters. Stochastic gradient descent procedures approach a (local) minimum of the objective function in the limit $\eta \to 0$. We can consider this limit exactly by rescaling the learning time as $\tilde{\alpha} = \eta \alpha$. Then, the $O(\eta^2)$ terms in Eq. (3.14) can be neglected and the set of ODEs is simplified. For all demonstrations, the NG algorithm is studied for decreasing $\lambda$ with

$$\lambda(\tilde{\alpha}) = \lambda_i (\lambda_f / \lambda_i)^{\tilde{\alpha} / \tilde{\alpha}_f} \quad (3.19)$$

where $\lambda_i$ and $\lambda_f$ are respectively the initial and final settings of the rank parameter and $\tilde{\alpha}_f$ is a learning time parameter. The influence of the initial set of prototypes on the learning curves is investigated by choosing different values of $\{R_{S\sigma}(0), Q_{ST}(0)\}$.

Figure 3.1 presents the prototype dynamics in a system with three prototypes

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3_1}
\caption{Trajectories of prototypes $\{w_i\}_{i=1}^3$ on the plane spanned by $B_1$ and $B_2$, displayed with the order parameters $R_{i\sigma} = w_i \cdot B_\sigma$. The cluster centers $\ell_\sigma B_\sigma$ are marked by crosses. The trajectories are marked by solid lines (WTA) and dashed lines (NG). The prototypes at initialization are marked with squares and at $\tilde{\alpha} = 10$ with circles (WTA) and dots (NG). Both algorithms converge at the triangles, where two prototypes coincide at $\{0.07, 1.07\}$. The set of prototypes is initially set (a) near the cluster centers, and (b) far away from the cluster centers. In both figures the parameters are $p_1 = 0.45, \ell_1 = \ell_2 = 1, \nu_1 = \nu_2 = 1, \eta \to 0, \lambda_i = 2, \lambda_f = 0.01$ and $t_f = 50$.}
\end{figure}
3.4. Results

Figure 3.2: The corresponding order parameters $R_{i2}$ at learning time $\tilde{\alpha} = \eta \alpha$ for WTA (solid lines) and NG (dashed lines) algorithms in the system described in Fig. 3.1. The initial sets of prototypes are defined in Figs. 3.1(a) and (b), respectively.

and two clusters. We examine two different initial sets of prototypes: close to the origin at $\{R_{S1}(0), R_{S2}(0)\} \approx \{0, 0\}, Q_{ST}(0) \approx 0, \forall\{S, T\}$ in Fig. 3.1 and far away from the origin on the side of the weaker cluster, viz. $p_i$, at $\{R_{S1}(0), R_{S2}(0)\} \approx \{3, -2\}$, $Q_{ST}(0) = R_{Sx}(0) \cdot R_{Ty}(0), \forall\{S, T\}$ in Fig. 3.1. While the prototypes have different trajectories in WTA and NG algorithms, they converge at the identical configuration at large $\alpha$ and $\lambda \to 0$. Here, the projections of two prototypes converge near the center of the stronger cluster. The advantage of NG is apparent in Fig. 3.1 where all prototypes already reach the area near the cluster centers at an intermediate learning stage $\tilde{\alpha} = 10$.

This can be illustrated with the evolution of the order parameters $R_{S2}(\alpha)$ in Fig. 3.2. In Fig. 3.2, the order parameters of both algorithms converge relatively fast. In Fig. 3.2, the order parameters of one prototype change rapidly compared to that of other prototypes in WTA algorithm. One prototype dominates as the winner and gets frequent updates towards the cluster centers, while the other prototypes are rarely updated. The NG algorithm partially solves this problem by updating all prototypes at the initial stages of learning.

The quantization error obtained from the order parameters $\{R_{Sx}(\tilde{\alpha}), Q_{ST}(\tilde{\alpha})\}$ is displayed in Fig. 3.3. We observe that the quantization error decreases faster in the WTA algorithm compared to NG methods at the initial stages of the learning. This behavior can be explained by the fact that the $H_{NC}$ differs from $E(W)$ by smoothing terms in particular in early stages of training. We observe that WTA yields the best overall quantization error in the first set of initial values in Fig. 3.3. This is mirrored by the fact that, for large $\tilde{\alpha}$ and $\lambda_f \to 0$, both algorithms yield the same quantization error.
3. Learning dynamics and robustness of vector quantization and neural gas

**Figure 3.3**: Evolution of the quantization error $E(W)$ in Fig. 3.1 at learning time $\tilde{\alpha} = \eta \alpha$ for WTA (solid line) and NG (dashed line) algorithms. The prototypes are initialized (a) near the cluster centers and (b) far away from the cluster centers.

For WTA training, the prototypes reach $\tilde{\alpha} \to \infty$ asymptotic positions corresponding to the global minimum of $E(W)$ for small learning rates $\eta \to 0$. However, learning can slow down significantly at intermediate stages of the training process. Transient configurations may persist in the vicinity of local minima and can indeed dominate the training process. The NG is more robust w.r.t. the initial position of prototypes than WTA while achieving the best quantization error asymptotically.

### 3.4.2 Asymptotic configuration

The dynamics of the prototypes while learning on a model data with a larger separation between the clusters are presented in Fig. 3.4. The initial configurations correspond to the following values of $\{R_{S1}(0), R_{S2}(0)\}$: (a) $\{-1, 2\}$, (b) $\{-0.5, 2\}$, (c) $\{-1, 1.5\}$, (d) $\{-0.5, 1.5\}$, (e) $\{-1, 1\}$ and (f) $\{-0.5, 1\}$. In all panels, $Q_{ST}(0) = R_{S\sigma}(0) \cdot R_{T\sigma}(0)$.

In this case, the optimal configuration of prototypes is with two prototypes representing the stronger cluster as in Figs. 3.4(a to c). However, the asymptotic configuration of the prototypes in the WTA algorithm are sensitive to the initial conditions. In some cases, viz. Figs. 3.4(d to f), this configuration is not the optimal set of prototypes. Therefore, even in this comparably simple model, prototypes in WTA can be confined in suboptimal local minima of the cost function $E(W)$. The issue of different regions of initialization which lead to different asymptotic configurations are to be discussed in forthcoming projects.

The asymptotic configurations for the NG algorithm are independent of initial conditions as shown in Figs. 3.4(a to f). During the learning process with $\lambda > 0$ the
3.4. Results

Figure 3.4: Trajectories of the prototypes on the plane spanned by $\mathbf{b}_1$ and $\mathbf{b}_2$, corresponding to the WTA (solid lines) and the NG (dashed lines) algorithms. Here, $p_1 = 0.45$, $p_2 = 0.55$, $\nu_1 = 1$, $\nu_2 = 1.21$, $\ell_1 = 1$ and $\ell_2 = 5$. The cluster centers $\ell_{\sigma} \mathbf{b}_{\sigma}$ are marked by $\times$. The initial prototype configurations for both algorithms are marked with $\square$. While the asymptotic configurations of WTA (circles) algorithm depends on initialization, the NG (dots) always produces identical asymptotic configurations. In these cases, the NG algorithm always finds the optimal quantization error.

The system moves towards intermediate configurations with minimum $H_{\text{NG}}(\mathbf{W})$. Given sufficiently large $\lambda$ and $\tilde{\alpha}$, these configurations are identical and therefore the NG algorithm is robust with respect to initial conditions. In these cases, the asymptotic configuration is the optimal configuration and thus the NG algorithm achieves optimal performance.

We demonstrate a model where the NG algorithm does not yield optimal per-
Learning dynamics and robustness of vector quantization and neural gas

Figure 3.5: (a) The optimal set of prototypes (solid dots) in a system with three clusters projected on the plane space spanned by \( \{B_1, B_3\} \). The values of \( R_2 \) are not shown here. The cluster centers \( \ell_\sigma B_\sigma \) are marked by \( \times \). (b,c,d) Trajectories of the prototypes using the NG algorithm with different initial conditions. Their initial (squares) and asymptotic (solid dots) configuration of the prototypes are indicated. The parameters are \( p_1 = 0.25, p_2 = 0.20, p_3 = 0.55, \nu_1 = \nu_2 = \nu_3 = 1, \ell_1 = \ell_2 = 1 \) and \( \ell_3 = 5 \).

In this more complex situation, the weaker cluster (\( p_\sigma = 0.45 \)) is divided into two Gaussian clusters with \( p_{1,2} = \{0.25, 0.20\} \). This corresponds to a system of three clusters, with \( \ell_\sigma = \{1, 1, 5\} \) and \( p_\sigma = \{0.25, 0.20, 0.55\} \). The distance between the first two clusters is small compared to their distance to the third cluster. In comparison to the previous case, where the weaker cluster spreads out evenly in all directions, here it has a particular orientation along the vector \( (B_1 - B_2) \). Because of this structure, the best quantization error is obtained when one prototype is placed near each cluster center, as in Fig. 3.5 even though one cluster has a very large prior (\( p_3 > p_1 + p_2 \)).

Similar to the previous case, the asymptotic configuration for the NG algorithm is independent of initial conditions. However, this configuration with two prototypes near the center of the stronger cluster in Figs. 3.5(b to d), is not the optimal configuration. Even with prototypes initialized at the optimal set as in Fig. 3.5, the NG algorithm may still lead to suboptimal configurations.

The characteristics of the cost function \( H(W) \) of NG, i.e. its minima, can be radically apart with different values of \( \lambda \). While the NG may find the configuration of the global minima of \( H(W) \) for large \( \lambda \), these configurations do not always lead to the global minima for smaller \( \lambda \). Consequently, the asymptotic configuration may correspond to a local minimum of \( E(W) \) and the NG algorithm does not always
yield the optimal quantization error.

3.5 Conclusion

We have presented an exact mathematical analysis of the dynamics of vector quantization for high dimensional data. Performance is measured by the evolution of the quantization error. In a learning scenario with no sub-optimal local minima of the quantization error, the WTA always converges to the best quantization error. However, learning can slow down significantly if the prototypes are initialized far from the region of high data density. The NG is less sensitive to the initial conditions and achieves both robustness and optimal asymptotic quantization error. Thereby, the convergence speed of NG algorithms is comparable or (for initialization outside the clusters) better than the convergence speed of simple WTA mechanisms, while achieving the same final quantization error.

In the presence of local minima, the WTA algorithm may converge into different asymptotic configurations depending on its initial conditions. The NG algorithm is very robust, i.e. relatively insensitive to initial conditions. However, we demonstrate a test case where it does not find the best asymptotic quantization error. The above discussed sub-optimal outcome of NG training might result from the specific schedule at which $\lambda$ is decreased in the course of training. The influence of both schedules for $\eta$ and $\lambda$ will be studied in greater detail in forthcoming projects.

The formalism allows for the design of optimal schemes in the framework of the model situation. While this model clearly does not describe the complexity of real world problems, it is useful to demonstrate certain characteristics of both algorithms. Immediate extensions of the model towards realistic data structures could include additional or non-spherical clusters. An important investigation for rank-based learning schemes could include the analysis of the popular Self Organising Maps (SOM) schemes, which apply a predefined lattice of prototype. Comparisons of SOM to Neural Gas systems would provide insight on the importance of preserving the topology of prototypes.
Chapter 4

Window-based example selection in learning vector quantization

Abstract

A variety of modifications has been employed to Learning Vector Quantization (LVQ) algorithms using either crisp or soft windows for selection of data. Although these schemes have been shown in practice to improve performance, a theoretical study on the influence of windows has so far been limited. Here we rigorously analyse the influence of windows in a controlled environment of Gaussian mixtures in high dimensions. Concepts from statistical physics and the theory of on-line learning allow for an exact description of the training dynamics, yielding typical learning curves, convergence properties and achievable generalization abilities. We compare the performance and demonstrate the advantages of various algorithms, including LVQ 2.1, Generalized LVQ (GLVQ), Learning From Mistakes (LFM) and Robust Soft LVQ (RSLVQ). We find that the selection of the window parameter highly influences the learning curves, but surprisingly not the asymptotic performances of LVQ 2.1 and RSLVQ. Although the prototypes of LVQ 2.1 exhibit divergent behavior, the resulting decision boundary coincides with the optimal decision boundary thus yielding optimal generalization ability.

4.1 Introduction

Learning Vector Quantization (LVQ) constitutes a family of learning algorithms for nearest prototype classification of potentially high dimensional data (Kohonen 1997). The intuitive approach and computational efficiency of LVQ classifiers have motivated its application in various disciplines, see e.g. (Neural Networks Research Centre, Helsinki 2002). Prototypes in LVQ algorithms represent typical features within a data set using the same feature space instead of the black-box approach practiced in many other classifiers, e.g. feedforward neural networks or support vector machines. This makes them attractive to researchers outside the field of machine learning. Other advantages of LVQ algorithms are (1) they are easy to implement for multi-class classification problems and (2) the algorithm complexity can be adjusted during training as required.
Numerous variants of the original LVQ prescriptions have been proposed towards achieving better performance, such as LVQ 2.1 (Kohonen 1990, Kohonen 1997), LVQ 3 (Kohonen 1990, Kohonen 1997), Generalized LVQ (GLVQ) (Hammer and Villmann 2002, Sato and Yamada 1995) and Robust Soft LVQ (RSLVQ) (Seo and Obermayer 2003). Common themes of these modifications include an additional parameter which controls the selection of data to which the system is adapted and variation of the magnitude of prototype updates. We refer to these in general as window schemes. In the limiting case of hard or crisp learning schemes, updates are restricted only to examples which fall into this window. For instance, LVQ 2.1 allows updates as long as the example is in the vicinity of the current decision boundary. Alternatively, learning schemes can implement a soft window, e.g. RSLVQ and GLVQ, which considers all examples but adapts the magnitude of the update according to their relative distances to the current decision boundary.

In general, the learning behavior of these strategies is not well understood. It is unclear how the convergence, stability and achievable generalization ability compare for the different strategies. Fortunately, methods from statistical physics and theory of on-line learning recently allowed a systematic investigation of very large systems in the so-called thermodynamic limit. This has been successfully applied in, among others, feedforward neural networks, perceptron training and principal component analysis (Biehl and Caticha 2003, Engel and van den Broeck 2001, Saad 1999). A similar approach to LVQ-type algorithms, e.g. LVQ 1, unsupervised VQ and rank-based Neural Gas, was treated in (Biehl et al. 2007, Witoelar et al. 2008).

In this work, we closely examine the influence of window schemes for LVQ algorithms. Typical learning behavior is studied within a model situation of high dimensional Gaussian clusters and competing prototypes. From this analysis, we can observe typical learning curves and the convergence properties, i.e. the asymptotic behavior in the limit of an arbitrarily large number of examples.

Typically the window parameters are selected either heuristically or derived from prior knowledge of the data and kept fixed during training. The optimal parameter settings are chosen according to a computationally expensive validation procedure. It is also possible to treat the hyperparameters as dynamic properties during learning, e.g. by means of an annealing schedule (Seo and Obermayer 2006) or a gradient-based optimization method (Bengio 2000). Using the model described in this paper, one can investigate the optimality of the parameters for both fixed and dynamic settings in representative model situations.
4.2 Model

Throughout the paper, we study LVQ algorithms in a model situation as described in Section 2.2: high dimensional data are generated from a mixture of $M$ Gaussian clusters and presented to a system of two or three prototypes. We restrict ourselves to the analysis of isotropic and homogeneous clusters, i.e. each cluster $\sigma$ generates only data with one of the class labels $y_\sigma = \{1, 2, \ldots, Y\}$ where $Y$ is the number of classes. Examples $\{\xi^\mu, y^\mu_\sigma\}$ with $\xi^\mu \in \mathbb{R}^N$ are drawn independently according to the probability density function

$$P(\xi) = \sum_{\sigma=1}^{M} p_\sigma P(\xi|\sigma)$$

with $P(\xi|\sigma) = \frac{1}{(2\pi\upsilon_\sigma)^{N/2}} \exp \left[ -\frac{1}{2\upsilon_\sigma} (\xi - \ell_\sigma B_\sigma)^2 \right]$ (4.1)

where $p_\sigma$ are the cluster-wise prior probabilities and $\sum_\sigma p_\sigma = 1$. The components of vectors $\xi^\mu$ from cluster $\sigma^\mu$ are random numbers with mean vectors $\ell_\sigma B_\sigma$ and variance $\upsilon_\sigma$. The unit vectors $B_\sigma$ determine the orientation of cluster centers.

In this framework we formally exploit the thermodynamic limit $N \to \infty$ corresponding to very high dimensional data. This has simplifying consequences which will be present throughout the paper. Note that on random subspace projections, data from different clusters completely overlap and are not separable. The clusters become apparent only in the, at most, $M$-dimensional space spanned by vectors $\{B_\sigma\}_{\sigma=1}^{M}$. The non-trivial goal is to identify this subspace from the $N$-dimensional data.

4.3 Window-based LVQ algorithms

We shortly review LVQ algorithms and their corresponding window schemes. For the model defined in Section 4.2, we define an LVQ system as a set of $K$ prototypes $W = \{w_S, c_S\}_{S=1}^{K}$ with $w_S \in \mathbb{R}^N$ and $c_S = \{1, 2, \ldots, Y\}$. Classification is implemented through a nearest prototype scheme: novel examples will be assigned to the class of the closest prototype according to a dissimilarity measure. Here we restrict the measure to the squared Euclidean distance $d_S^\mu = (\xi^\mu - w_S)^2$ for a given novel example $\xi^\mu$. In the on-line algorithm, examples are presented sequentially to the system and the prototypes are adapted by the following update step

$$w^\mu_S = w^{\mu-1}_S + \eta N f_S [d_1^\mu, \ldots, d_K^\mu, c_1, \ldots, c_K, y^\mu_\sigma] \left( \xi^\mu - w^{\mu-1}_S \right),$$

(4.2)

where $w^{\mu}_S$ denotes the prototype after presentation of $\mu$ examples and the learning rate $\eta$ is rescaled with $N$. We use the shorthand $f_S$ for the modulation function.
which controls, along with the learning rate \( \eta \), the magnitude of the update of \( w_S \) towards or away from the current example. In this work, we investigate several LVQ prescriptions which include window schemes.

### 4.3.1 LVQ 2.1

LVQ 2.1 was proposed by Kohonen aiming at efficient separation between prototypes of different classes and has been shown to provide good classification results (Kohonen 1990, Neural Networks Research Centre, Helsinki 2002). Given an example \( \xi^\mu \), two nearest prototypes \( w_S \) and \( w_T \) are updated if the following conditions are met: (i) the classes \( c_S \) and \( c_T \) are different, and (ii) either \( c_S \) or \( c_T \) is equal to \( y_\mu^\sigma \).

The prototype with the correct class is moved towards the data while the other is moved farther away with 
\[
f_S = 1, 
\]f_T = -1
if \( c_S = y_\mu^\sigma \); 
\[
f_S = -1, 
\]f_T = +1
else.

It is well known that such learning rule has stability problems for unbalanced data sets, resulting in diverging prototypes with deteriorating performance (Kohonen 1990). Therefore, LVQ 2.1 restricts updates to examples \( \xi^\mu \) which fall into a window around the decision boundary.

\[
\min \left( \frac{d^\mu}{d_S}, \frac{d^\mu}{d_T} \right) > \rho, \quad \text{with} \quad \rho = \frac{1 - \omega}{1 + \omega} \quad (4.3)
\]

where \( \omega \) is a window parameter, \( 0 < \omega \leq 1 \) and therefore \( 1 > \rho \geq 0 \). However, this window is ineffective for very high dimensional data, as we obtain \( \lim_{N \to \infty} (\xi^\mu - w_S)^2 \approx (\xi^\mu)^2 \) because \( (\xi^\mu)^2 = O(N) \) terms dominate the other \( O(1) \)-terms, i.e. \( (w_S \cdot \xi^\mu) \) and \( (w_T^2) \). Consequently, this window definition does not work in very high dimensions, evidenced by

\[
\lim_{N \to \infty} \min \left( \frac{(\xi^\mu - w_S^\mu-1)^2}{(\xi^\mu - w_S^\mu-1)^2}, \frac{(\xi^\mu - w_T^\mu-1)^2}{(\xi^\mu - w_T^\mu-1)^2} \right) = 1, \quad (4.4)
\]

which implies that every example falls into the window. Therefore, in the following we implement the constraint

\[
| (\xi^\mu - w_T)^2 - (\xi^\mu - w_S)^2 | \leq k \min \left( (\xi^\mu - w_S)^2, (\xi^\mu - w_T)^2 \right) \quad (4.5)
\]

where \( k \) is a small positive number. Note that the term \( (\xi^\mu)^2 = O(N) \) cancels out on the left hand side, while it dominates on the right hand side for \( N \to \infty \). Thus, the right hand side becomes \( k \cdot (\xi^\mu)^2 \) and the condition is non-trivial only if \( k = O(1/N) \). We introduce the rescaled window parameter \( \delta = k \cdot (\xi^\mu)^2 = O(1) \) so that
the window scheme is $-\delta \leq (d_T^u - d_S^u) \leq \delta$; $\delta$ is positive. We describe these rules as the following modulation function

$$f_S = \chi(c_S, y_S^u) \sum_{T : c_T \neq c_S} (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) \prod_{U \neq S, T} \Theta_{SU} \Theta_{TU}$$ \hspace{1cm} (4.6)$$

with $\chi(c_S, y_S^u) = 1$ if $c_S = y_S^u$ and $\chi(c_S, y_S^u) = -1$ else. We use the shorthand notation $\Theta_{ji}^x \equiv \Theta(d_{i}^u - d_{j}^u - \delta)$, where $\Theta(x)$ is the Heaviside function $\Theta(x) = 1$ if $x > 0$; 0 else.

We sum over prototypes $\{w_T | c_T \neq c_S\}$ and terms $(\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) = \Theta(d_T^u - d_S^u + \delta) - \Theta(d_T^u - d_S^u - \delta)$ enforce the window condition. The product term $\prod_{U \neq S, T} \Theta_{SU} \Theta_{TU}$ singles out instances where $w_S$ and $w_T$ are the two closest prototypes. This form of $f_S$ allows for the analysis given in Section 4.4.

4.3.2 LFM-W

A simple modification to overcome the stability problems of LVQ 2.1 is restricting updates only on misclassified examples. Analogous to perceptron learning, we term this update rule as Learning From Mistakes (LFM). Here, the closest prototype $w_J$ with the same class $c_J = y_J^u$ (correct winner) and closest prototype $w_K$ with a different class $c_K \neq y_K^u$ (incorrect winner) are updated with $f_J = +1$ and $f_K = -1$, if the example is misclassified. On the contrary, if the winning prototype is already correct, the configuration is left unchanged. This prescription can be interpreted as a limiting case of cost function based Robust Soft LVQ (RSLVQ), which will be explained later in this section. Because the cost function of RSLVQ is bounded from below, stability can also be expected in LFM.

The performance of LFM can be improved by including data selection of data using the window rule in Eq. (4.5). We refer to this algorithm as LFM with a window (LFM-W), represented by the modulation function

$$f_S = \begin{cases} \sum_{K : c_K \neq y_S} (\Theta_{KS} - \Theta_{KS}^{+\delta}) \psi(S, K) & \text{if } c_S = y_S^u \\ \sum_{J : c_J = y_S} (\Theta_{SJ} - \Theta_{SJ}^{+\delta}) \psi(J, S) & \text{else.} \end{cases}$$ \hspace{1cm} (4.7)$$

with $\Theta_{ji} \equiv \Theta(d_{i}^u - d_{j}^u)$ and $\psi(J, K) = \prod_{T : c_T = y_T} \Theta_{JT} \prod_{U : c_U \neq y_U} \Theta_{KU}$ which identifies cases with $w_J$ being the correct winner and $w_K$ being the incorrect winner: $\psi(J, K) = 1$ if this condition is fulfilled and $\psi(J, K) = 0$ else. Terms in parentheses single out misclassified examples which fall into the window.
4.3.3 GLVQ

Earlier LVQ prescriptions, including LVQ 2.1, were based on heuristic grounds. In contrast, a popular variant termed the Generalized LVQ was proposed in (Sato and Yamada 1995) which introduced the cost function

\[ E = \sum_{\mu} \Phi(\tau(\xi^\mu)) \quad \text{with} \quad \tau(\xi^\mu) = C \frac{d^\mu_J - d^\mu_K}{d^\mu_J + d^\mu_K} \]

(4.8)

where \( \Phi(\tau) \) is a (usually non-linear) monotonically increasing function, \( w_J \) is the nearest correct prototype and \( w_K \) is the nearest incorrect prototype to the example \( \xi^\mu \). We insert the scaling parameter \( C \) which will be required for high dimensions.

Stochastic gradient procedure on (4.8) yields the learning rule

\[ f_J = 2C \frac{\partial \Phi(\tau)}{\partial \tau} \frac{d^\mu_K}{(d^\mu_J + d^\mu_K)^2}, \quad f_K = -2C \frac{\partial \Phi(\tau)}{\partial \tau} \frac{d^\mu_J}{(d^\mu_J + d^\mu_K)^2}. \]

(4.9)

Here the usefulness of selecting a non-linear \( \Phi(\tau) \) is shown. For instance, in (Hammer and Villmann 2002, Sato and Yamada 1995), the sigmoid function is chosen:

\[ \Phi(\tau) = \frac{1}{1 + \exp(-\tau)} \]

To obtain a non-zero argument, \( C \) must also be in the order \( O(N) \), and we rescale using \( v_G = (d^\mu_J + d^\mu_K)/C = O(1) \). The parameter \( v_G \) determines the softness of the window, provided that an appropriate non-linear \( \Phi(\tau) \) is chosen. Note that GLVQ can be simplified to LVQ 2.1 without a window using the identity function \( \Phi(\tau) = \tau \). The cost function in (4.8) becomes

\[ E = \sum_{\mu} \Phi \left( \frac{1}{v_G} (d^\mu_J - d^\mu_K) \right) \]

(4.10)

To obtain a non-zero argument, \( C \) must also be in the order \( O(N) \), and we rescale using \( v_G = (d^\mu_J + d^\mu_K)/C = O(1) \). The parameter \( v_G \) determines the softness of the window, provided that an appropriate non-linear \( \Phi(\tau) \) is chosen. Note that GLVQ can be simplified to LVQ 2.1 without a window using the identity function \( \Phi(\tau) = \tau \). The cost function in (4.8) becomes

\[ E = \sum_{\mu} \Phi \left( \frac{1}{v_G} (d^\mu_J - d^\mu_K) \right) \]

where \( \Phi(\tau) = \int_{-\infty}^{\tau} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{t^2}{2} \right) dt \)

(4.11)

Note that this form implements a Gaussian window similar to the sigmoidal cost described in (Hammer and Villmann...
4.3. Window-based LVQ algorithms

Figure 4.1: Left panel: The form of the chosen $\Phi(\tau)$ in GLVQ, in comparison to the sigmoidal function $\text{Sig}(\tau)$. The derivatives produce a soft window. Middle and right panel: The RSLVQ modulation function $f_S$ for class 1 ($\circ$) when presented with data from class 1. The figures display the difference between smaller $v_{\text{soft}}$ (left) and larger $v_{\text{soft}}$ (right).

Plugging in the form of (4.10), we obtain the learning rules

$$f_J = \frac{2}{v_G} \phi \left( \frac{d_J - d_K}{v_G} \right), \quad f_K = -\frac{2}{v_G} \phi \left( \frac{d_J - d_K}{v_G} \right)$$

(4.12)

We can write the modulation function as

$$f_S = \begin{cases} \sum_{K \neq K \neq y} \frac{2}{v_G} \phi \left( \frac{d_S - d_K}{v_G} \right) \psi(S, K) & \text{if } c_S = y^w \\ -\sum_{J \neq J \neq y} \frac{2}{v_G} \phi \left( \frac{d_J - d_S}{v_G} \right) \psi(J, S) & \text{else.} \end{cases}$$

(4.13)

with $\psi(J, K) = \prod_{T, c_T = y^w} \Theta_{JT} \prod_{U, c_U \neq y^w} \Theta_{KU}$.

4.3.4 RSLVQ

The Robust Soft LVQ algorithm (Seo and Obermayer 2003) was derived using a statistical modeling of the data and designed to overcome the stability problem of LVQ 2.1. RSLVQ introduces soft prototype assignments which act similarly to a soft window around the decision boundary. This algorithm minimizes a bounded cost function $E = -\ln(L)$ where $L$ is based on a likelihood ratio function of a mixture
model, described as

\[
L = \prod_{\mu=1}^{P} \frac{p(\xi^\mu, \sigma^\mu | W)}{p(\xi^\mu | W)} \quad \text{with} \quad \begin{cases} 
 p(\xi^\mu, \sigma^\mu | W) = \sum_{j:y_j=y^\mu} P_j p(\xi^\mu | j), \\
 p(\xi^\mu | W) = \sum_{j=1}^{K} P_j p(\xi^\mu | j),
\end{cases}
\] (4.14)

where \( p(\xi^\mu | W) \) approximates the actual probability density \( P(\xi) \), c.f. (4.1). It is assumed that every component \( j \) of the mixture generates examples which belong to one class, viz. \( c_j \). \( N_\sigma \) is the number of classes and \( P_j \) is the probability that the examples are generated by a particular component \( j \) and \( p(\xi^\mu | j) \) is the conditional probability that \( j \) generates a particular example \( \xi^\mu \).

The learning rule is obtained by performing stochastic gradient descent on the cost function \( E \) with respect to \( w_S \). We examine it for a Gaussian mixture ansatz as in (Seo and Obermayer 2003), where it is chosen \( p(\xi^\mu | j) = (2\pi v_j)^{N/2} \exp(-d_{\mu j}^2 / 2v_j) \).

Furthermore, every component is assumed to have equal probability \( P(j) = 1/K \), \( \forall j \) and equal variance \( v_j = v_{\text{soft}}, \forall j \) where \( v_{\text{soft}} \) is called the softness hyperparameter.

This gives the following modulation function

\[
f_S = \frac{1}{v_{\text{soft}}} \begin{cases} 
 P_\sigma(S|\xi^\mu) - P(S|\xi^\mu), & \text{if } c_S = y^\mu_{\sigma} \\
 -P(S|\xi^\mu), & \text{else,}
\end{cases}
\]

with the assignment probabilities

\[
P_\sigma(S|\xi^\mu) = \frac{\exp(-d_{S\mu}^2 / 2v_{\text{soft}})}{\sum_{j:y_j=\sigma} \exp(-d_{j\sigma}^2 / 2v_{\text{soft}})}, \quad P(S|\xi^\mu) = \frac{\exp(-d_{S\mu}^2 / 2v_{\text{soft}})}{\sum_j \exp(-d_{j\mu}^2 / 2v_{\text{soft}})},
\] (4.15)

see (Seo and Obermayer 2003) for the derivations. \( P_\sigma(S|\xi^\mu) \) describes the posterior probability that \( \xi^\mu \) is assigned to the component \( S \) of the mixture, given that the example is generated by the correct class. \( P(S|\xi^\mu) \) describes the posterior probability that \( \xi^\mu \) is assigned to the component \( S \) of the complete mixture using all classes. As \( v_{\text{soft}} \) becomes smaller, the updates become smaller for correctly classified examples and larger for incorrectly classified examples, see Fig. 4.1.

Note than the limiting case of \( v_{\text{soft}} \) is particularly simple. The assignments of Eq. (4.15) become hard assignments, i.e.

\[
P_\sigma(S|\xi^\mu) = \begin{cases} 
 1, & \text{if } d_{S\mu} = \min_{j:y_j=\sigma} \{d_{j\sigma}^2\} \\
 0, & \text{else}
\end{cases}, \quad P(S|\xi^\mu) = \begin{cases} 
 1, & \text{if } d_{S\mu} = \min_{(j)} \{d_{j\mu}^2\} \\
 0, & \text{else}
\end{cases}
\] (4.16)

Plugging the above into (4.15), we obtain the learning rule for learning from mistakes (LFM), described in section 4.3.2.
4.4 Analysis

The learning dynamics of window-based LVQ algorithms are studied along the lines of theory of on-line learning, as described in Chapter 3. We briefly summarize the method of analysis and refer to Section 3.3 for the full explanation.

In the thermodynamic limit \( N \to \infty \), the system can be fully described in terms of a few characteristic quantities, so-called order parameters. For this learning model, a suitable chosen set of order parameters is

\[
R^\mu_{S\sigma} = w^\mu_S \cdot B_\sigma \quad Q^\mu_{ST} = w^\mu_S \cdot w^\mu_T,
\]

where \( R^\mu_{S\sigma} \) are the projections of prototype vectors \( w^\mu_S \) on the cluster center vectors \( B_\sigma \) and \( Q^\mu_{ST} \) correspond to the self- and cross- overlaps of the prototype vectors.

Recursion relations wrt. arrival of examples can be derived from the learning algorithm. Following the lines detailed in Section 3.3, the recursions are converted into a set of coupled ordinary differential equations (ODEs) in the thermodynamic limit in the following form:

\[
\frac{dR^\mu_{S\sigma}}{d\alpha} = \eta \left( \langle b_\sigma f_S \rangle - \langle f_S \rangle R^\mu_{S\sigma} \right)
\]

\[
\frac{dQ^\mu_{ST}}{d\alpha} = \eta \left( \langle h_S f_T \rangle - \langle f_T \rangle Q^\mu_{ST} + \langle h_T f_S \rangle - \langle f_S \rangle Q^\mu_{ST} \right) + \eta^2 \sum_\sigma p_\sigma \nu_\sigma \langle f_S f_T \rangle_\sigma,
\]

where \( \langle \ldots \rangle \) denotes averages over the full density \( P(\xi) \) and \( \langle \ldots \rangle_\sigma \) denotes conditional averages over \( P(\xi|\sigma) \); we have the relation \( \langle \ldots \rangle = \sum_\sigma p_\sigma \langle \ldots \rangle_\sigma \). The input data vectors enter the right side of Eq. (4.18) strictly as projections

\[
h_S = w_S \cdot \xi \quad b_\sigma = B_\sigma \cdot \xi.
\]

Note that the index \( \mu \) is omitted from \( \xi^\mu \), but the presented example at each learning step are implicitly assumed to be uncorrelated to \( w_S \). In various sections in this chapter, we investigate learning behaviors using small learning rates \( \eta \to 0 \) where we can neglect \( \eta^2 \)-terms in Eq. (4.18). To compensate for the small training steps, longer training times are required; the simultaneous limits \( \eta \to 0, \alpha \to \infty \) are taken and the number of examples are rescaled as \( \tilde{\alpha} = \eta \alpha \).

The quantities \( h_S, b_\sigma \) become correlated Gaussian quantities by means of the Central Limit Theorem. Therefore, they are fully specified by first and second moments, detailed in Appendix A. Thus, the above averages \( \langle f_S \rangle, \langle h_T f_S \rangle \) and \( \langle b_T f_S \rangle \) reduce to Gaussian integrations in \( K + M \) dimensions and can be expressed in...
4. Window-based example selection in learning vector quantization

Figure 4.2: Left panel: Evolution of the order parameters \( \{R_S, Q_{ST}\} \) for LVQ 2.1 with \( K = 2, M = 2, \ell_1 = \ell_2 = 1, p_1 = 0.7, v_1 = v_2 = 1 \) and learning parameters \( \eta = 0.1 \) and \( \delta = 1 \). Solid lines represent \( \{R_S, Q_{ST}\} \) obtained from the theoretical analysis, while bars represent the variance as produced by Monte Carlo simulations for \( N = 100 \) over 100 independent runs. Right panel: Influence of a window on LVQ 2.1 at learning time \( \alpha = 40 \). Prototypes are projected on the \((B_+, B_-)\) subspace for \( \delta = 1 \) (○), \( \delta = 5 \) (△) and unrestricted LVQ 2.1 (□). In the latter, one prototype strongly diverges. The resulting decision boundaries are indicated by chained lines. The origin is marked by (·) and the cluster centers are marked by (∗).

\( \{R_S, Q_{ST}\} \), see Appendix B.1. For various algorithms and a system with two competing prototypes, the averages can be calculated analytically. For three or more prototypes, the mathematical treatment becomes more involved and requires multiple numerical integrations.

Given the averages for a specific modulation function \( f_S \), we obtain a closed set of ODE. Using initial conditions \( \{R_{S\sigma}(0), Q_{ST}(0)\} \), we integrate this system for a given algorithm and obtain the evolution of order parameters in the course of training, \( \{R_{S\sigma}(\alpha), Q_{ST}(\alpha)\} \). The generalization error \( \epsilon_g \), i.e. the probability of the closest prototype \( w_\sigma \) carrying an incorrect label, is determined by considering the contribution from each cluster separately:

\[
\epsilon_g = \sum_{\sigma=1}^{M} p_\sigma \epsilon_{g,\sigma} \quad \text{with} \quad \epsilon_{g,\sigma} = \sum_{S: i_S \neq y_\sigma} \left( \prod_{T \neq S} \Theta_{ST} \right)_{\sigma},
\]

which can be calculated from \( \{R_{S\sigma}(\alpha), Q_{ST}(\alpha)\} \). For instance, for the simplest system with two clusters \( \sigma = \{+, -\} \) and prototypes \( w_+ \) and \( w_- \), the generalization
4.5. A simple case: two prototypes, two clusters

Error is written explicitly in terms of order parameters as

\[ \epsilon_{g,\sigma} = \Phi \left( \frac{Q_{\sigma\sigma} - Q_{-\sigma,-\sigma} - 2\epsilon\epsilon (R_{\sigma,\sigma} - R_{-\sigma,-\sigma})}{2\sqrt{\nu} \sqrt{Q_{\sigma\sigma} - 2Q_{\sigma,-\sigma} + Q_{-\sigma,-\sigma}}} \right), \]  

(4.21)

with \( \Phi(x) = \int_{-\infty}^{x} dt \frac{1}{\sqrt{2\pi}} \exp(-t^2/2) \), detailed in Appendix B.3. The form of \( \epsilon_{g,\sigma} \) for systems with more prototypes is more involved, and we refer the final result of the calculations to Appendix B.3. We obtain the learning curve \( \epsilon_{g}(\alpha) \) which quantifies the success of training. This method of analysis shows excellent agreement with Monte Carlo simulations of the learning system for dimensionality as low as \( N = 100 \), as demonstrated in Fig. 4.2.

4.5 A simple case: two prototypes, two clusters

In this section we discuss in detail the results of the analysis for the simplest non-trivial problem: two-prototype LVQ 2.1, GLVQ, LFM-W and RSLVQ systems and \( M = 2 \) with one Gaussian cluster per class. The model data is given in section 4.2. For simplicity, we denote the two clusters as \( \sigma = \{+, -\} \) and without loss of generality can choose \( \ell_+ = \ell_- = \ell \) and orthonormal \( B_\sigma \), i.e. \( B_i \cdot B_j = 1 \) if \( i = j \); 0 else.

We place an emphasis on the asymptotic behavior in the limit \( \alpha \to \infty \), i.e. the achieved performance for an arbitrarily large number of examples. The asymptotic generalization error \( \epsilon_g(\infty) \) scales with the learning rate, analogous to minimizing a cost function in stochastic gradient descent procedures. For LVQ 2.1 and RSLVQ, the best achievable generalization error is obtained in the simultaneous limit of small learning rates \( \eta \to 0 \), \( \alpha \to \infty \) and rescaling \( \tilde{\alpha} = \eta \alpha \to \infty \). However this limit is not meaningful for LFM, as will be explained later.

In this simple scenario, it is possible to exactly calculate the best linear decision boundaries (BLD) by linear approximation of the Bayesian optimal decision boundary, see (Biehl et al. 2004) for the calculations. We compare the results from each algorithm to the best linearly achievable error \( \epsilon_{g,\text{bd}} \).

4.5.1 LVQ 2.1

We first examine two-prototype systems, i.e. \( K = 2 \). Figures 4.2 illustrate the evolution of order parameters under the influence of a window and the trajectories of the prototypes projected onto the \( (B_+, B_-) \) subspace. Without additional constraints, LVQ 2.1 with two prototypes displays a strong divergent behavior in a system with unbalanced data, i.e. \( p_+ \neq p_- \). The repulsion factor dominates for the prototype representing the weaker cluster, here \( w_2 \). The order parameters associated with
4. Window-based example selection in learning vector quantization

Figure 4.3: Generalization error $\epsilon_g$ for LVQ 2.1 with $K = 2, M = 2, \ell = 1, p_+ = 0.8, p_- = 0.2$ and $\eta \to 0$. Left panel: $\epsilon_g$ vs $\tilde{\alpha}$ using $\delta = 1, 5, 20$ and without a window. Note the logarithmic scaling on the horizontal axis. The asymptotic errors for all settings of $\delta$ converge at $\epsilon_g^{\text{bld}}$, indicated by the dotted line. Right panel: $\epsilon_g$ at fixed learning times $\tilde{\alpha} = 2, 5$ and $20$ as a function of $\delta$.

Implementing the window scheme, $w_2$ is repulsed until the data densities of both classes within the window become more balanced. Subsequently, the order parameters change with more balance between both prototypes. The repulsion factor still dominates its counterpart, therefore both prototypes still diverge, viz. $R_{z,\sigma}$ for both prototypes display a linear change with $\tilde{\alpha}$ at large $\tilde{\alpha}$, but the decision boundary remains stable. Trivial classification is prevented, see the generalization error curves $\epsilon_g$ vs. $\tilde{\alpha}$ in the left panel of Fig. 4.3. Obviously, for smaller $\delta$ a considerable amount of data is filtered out and the initial learning stages slow down significantly. Meanwhile for large $\delta$, $\epsilon_g$ becomes non-monotonic and converges more slowly.

Hence the performance at finite $\tilde{\alpha}$ is dependent on $\delta$, displayed in Fig. 4.3 and parameter settings are highly critical in practical applications. Given learning time $\tilde{\alpha}$, an optimal choice of fixed $\delta$ exists, which clearly depends on the properties of the data. With larger $\tilde{\alpha}$, $\epsilon_g$ becomes less sensitive towards $\delta$ and the optimal setting of $\delta$ is smaller. Surprisingly, $\delta$ only influences the convergence speed while the non-trivial asymptotic generalization error $\epsilon_g(\infty)$ is insensitive to the choice of $\delta$ and equals the best achievable error $\epsilon_g^{\text{bld}}$ for each setting. This can be explained as follows. We can compare the asymptotic decision boundary to the BLD: the angle between them is equal to the angle between $(w_1 - w_2)$ and $(B_+ - B_-)$. This is calculated, using (4.17) and the orthonormality of $B_+$ and $B_-$, as

\[
\theta = \arccos \left( \frac{(w_1 - w_2)^T (B_+ - B_-)}{\|w_1 - w_2\| \|B_+ - B_-\|} \right).
\]
4.5. A simple case: two prototypes, two clusters

Figure 4.4: LFM-W with \( p_+ = 0.6, \ell = 1, v_+ = v_- = 1 \). Left: Asymptotic prototype configuration for LFM and LFM-W \( \delta = 5 \) and 4.5, projected on \{B\_+, B\_-\}. Cluster centers \( \ell(B\_+, B\_-) \) are indicated by *. The projection of \( w_1, w_2 \) lie parallel to the symmetry axis \( \ell(B\_+ - B\_-) \), although they retain components orthogonal to the \{B\_+, B\_-\} subspace. Right: \( \epsilon_g(\infty) \) as a function of the window size \( \delta \). The lines correspond to learning rates \( \eta = 0.2, 0.5 \) and 1.0.

\[
\varphi = \arccos \left( \frac{R_{1+} + R_{1-} - R_{2+} + R_{2-}}{\sqrt{2(Q_{11} - 2Q_{12} + Q_{22})}} \right),
\]

which is found to be zero for large \( \tilde{\alpha} \). Hence, the decision boundary becomes parallel to the BLD and only its offset produces the difference between \( \epsilon_g(\infty) \) and \( \epsilon_g^{\text{BLD}} \). In low dimensions, this offset oscillates around zero due to the window rule. In the thermodynamic limit, the fluctuations vanish and the LVQ 2.1 decision boundary coincides with the BLD.

4.5.2 LFM-W

The LFM scheme performs updates identical to LVQ 2.1 with the condition that the example is misclassified. A detailed investigation into the characteristics of \( K = 2 \) unrestricted LFM has been presented in (Biehl et al. 2007). There, it was shown that LFM produces stable prototype configurations for finite learning rates \( \eta \). The projection of the prototypes lies parallel to the symmetry axis \( \ell(B\_+ - B\_-) \), displayed in Fig. 4.3. However the prototypes \( w_1 \) and \( w_2 \) retain components orthogonal to the two dimensional subspace spanned by the cluster centers, indicated by \( Q_{ST} > R_{S+}R_{T+} + R_{S-}R_{T-} \) which implies

\[
|w_S|^2 > |R_{S+}B_+ + R_{S-}B_-|^2.
\]
The asymptotic generalization error $\epsilon_g(\infty)$ is suboptimal and insensitive to $\eta$: the asymptotic decision boundary remains at an angle $\varphi$ from the optimal hyperplane, c.f. Eq. (4.22), independent of $\eta$. The Euclidean distance between prototypes is given by the quantity

$$\Delta_q = \sqrt{(w_1 - w_2)^2} = \sqrt{Q_{11} - 2Q_{12} + Q_{22}},$$

(4.23)

which is found to be proportional to $\eta$ for $\alpha \to \infty$. At $\eta \to 0$, $\Delta_q \to 0$ and the prototypes coincide, and this limit is not meaningful in LFM.

In this analysis, we observe that window schemes can dramatically improve performance of LFM. Using a window, the tilt of the decision boundary from the optimal hyperplane, i.e $\varphi$ in (4.22), is reduced, resulting in lower $\epsilon_g(\infty)$. We observe that $\epsilon_g(\infty)$ decreases along with reducing $\delta$, displayed in the right panel of Fig. 4.4. However, a critical window size $\delta_c$ exists where the LFM unexpectedly becomes divergent and no stationary state exists. Smaller windows filter examples which produce more repulsion in the orientation of the cluster centers, and we observe asymptotically larger $\Delta_q$ as $\delta$ decreases. This is clearly observed in Fig. 4.4. Given a sufficiently small $\delta$, it is possible that the repulsion factor entirely outweighs the attractive factor. At $\delta < \delta_c$, it performs similar to LVQ 2.1: the angle $\varphi$ becomes zero and $\epsilon_g(\infty)$ is close to the best achievable error.

Unlike the unrestricted case, the learning rate $\eta$ can influence the asymptotic performance. The learning rate and window size are indirectly related, as shown in the right panel of Fig. 4.4. For example, learning with small learning rates requires smaller windows to achieve optimal asymptotic error. Note that the influence of the window size depends heavily on the structure of the data. For various data models, efficient window settings may only exist on a very limited range and window schemes may be ineffective to improve generalization performance while still maintaining stability.

4.5.3 GLVQ

Apart from the influence of $v_G$ to the overall learning rate, small $v_G$ corresponds to a sharp peak around the decision boundary while large $v_G$ corresponds to a very large window. Figure 4.5 displays the prototype lengths while using GLVQ: the soft window slows down the strong repulsion of the prototype of the weaker cluster, as opposed to unrestricted LVQ 2.1. While both prototypes still diverge because the cost function at $N \to \infty$ is not bounded, c.f. 4.10, the asymptotic $\epsilon_g$ remains non-trivial, see Fig. 4.5.

Note that $v_G$ directly relates to the overall learning rate $\eta/v_G$, refer to Eq. (4.13), which influences the level of noise in stochastic gradient procedures. We compare
4.5. A simple case: two prototypes, two clusters

**Figure 4.5:** Left panel: $Q_{11}$ and $Q_{22}$ for GLVQ (solid lines), compared to unrestricted LVQ 2.1 (dashed lines). The soft window of GLVQ slows down the repulsion of one prototype, but the prototypes remain divergent. Here $p_+ = 0.7$, $\ell = 1$, $v_+ = 2$, $v_- = 5$, $\eta = 0.25$. Right panel: Learning curves $\epsilon_g$ vs. $\alpha$ for softness $v_G = 2, 5$ and 50, note the logarithmic horizontal axis. The learning rates are maintained at $\eta/v_G = 0.1$. Large $v_G$ produces better asymptotic generalization error, but may exhibit non-monotonic behavior and require very long learning times.

**Figure 4.6:** $p_+ = 0.7$, $\ell = 1$, $v_+ = 1$, $v_- = 1$. The cost functions for GLVQ with $\eta \to 0$ (left panel) and RSLVQ with $\eta/v_{soft} = 1$ (right panel) decrease monotonically, corresponding to a stochastic gradient descent.

results with respect to $v_G$, while maintaining at equal overall learning rate by keeping $\eta/v_G$ constant, in Fig. 4.5. Performance deteriorates at smaller $v_G$, where training slows down at intermediate stages and converges at a higher error. However, very large $v_G$ allows strong repulsion of the weaker prototype which results in non-monotonic $\epsilon_g$ and long learning convergence times. Surprisingly, the soft GLVQ window is outperformed by the simple hard or crisp window of LVQ 2.1. This is caused by the long tail of the modulation function which sums up into a large repulsion, whereas in the crisp window, only data near the decision boundary are
considered.

Figure 4.6 displays the cost function during learning. In the initial learning stages, the minimization of the cost function $E$ leads to fast decrease of $\epsilon_g$. However, while the cost function continues to decrease monotonically, $\epsilon_g$ behaves non-monotonically. While many techniques are developed to improve minimization procedures of $E$, it is important to evaluate the choice of $E$ and its correlation to the desired generalization performance.

4.5.4 RSLVQ

Finally in this section, we study the influence of the softness parameter $v_{\text{soft}}$ in the RSLVQ algorithm.

We first investigate model scenarios with equal variance clusters $v_+ = v_-$ and unbalanced data $p_+ \neq p_-$. We observe the influence of $v_{\text{soft}}$ on the learning curves while maintaining a fixed overall learning rate $\eta/v_{\text{soft}}$, displayed on the left panel of Figure 4.7. The generalization error curve depends on $v_{\text{soft}}$: at large $v_{\text{soft}}$, $\epsilon_g$ may exhibit non-monotonic behavior, reminiscent of LVQ 2.1. Because of this behavior, the learning process may require long learning times before reaching the asymptotic configuration. This is an important consideration for practical applications which often uses early stopping strategies to avoid overtraining. Meanwhile, the algorithm minimizes the cost function $E$ in (4.14) monotonically, see Fig. 4.6. Thus, the decrease in $E$ does not always result in a decrease of $\epsilon_g$.

A major advantage of the RSLVQ algorithm is the convergence of prototypes, i.e. a stationary configuration of order parameters exists for finite $v_{\text{soft}}$. The asymptotic configuration of prototypes are displayed in Fig. 4.8. At $\tilde{\alpha} \to \infty$, the softness parameter controls only the distance between the two prototypes: $\Delta_q$ as defined in Eq. (4.23), decreases linearly with $v_{\text{soft}}$. Note that under the conditions $p_+ = 0.5$, $v_{\text{soft}} = v_+ = v_-$ and initialization of prototypes on the symmetry axis, each prototype is located at its corresponding cluster center, i.e. the RSLVQ mixture model matches exactly to the actual input density.

Figures 4.7 compare the asymptotic errors in the case of $\eta/v_{\text{soft}} = 1$ (left panel) and small learning rates $\eta/v_{\text{soft}} \to 0$ (right panel). In the former case, performance improves with large $v_{\text{soft}}$: at small $v_{\text{soft}}$, the system converges at high $\epsilon_g$ similar to LFM, while at larger $v_{\text{soft}}$, it approaches the best linear decision. Meanwhile, at small learning rates, the asymptotic error becomes independent to $v_{\text{soft}}$. Therefore, given sufficiently small learning rates, RSLVQ becomes robust wrt. its softness parameter.

In the equal variance scenario, the asymptotic decision boundary always converges to the best linear decision boundary for all settings of $\{p_+, p_-\}$ and RSLVQ outperforms both LFM and LVQ 2.1, as it provides robustness, stability and low
4.5. A simple case: two prototypes, two clusters

Figure 4.7: Learning curves $\epsilon_g$ for RSLVQ using softness parameter $v_{\text{soft}} = 1, 2, 10$ and 20. Left: $p_+ = 0.7$ and equal variance $v_+ = v_- = 1$ with fixed overall learning rate $\eta/v_{\text{soft}} = 1$. Right: $p_+ = 0.6$ and unequal variance $v_+ = 1, v_- = 4$ with $\eta/v_{\text{soft}} \to 0$. The asymptotic errors is independent of $v_{\text{soft}}$ at small learning rates, but at a suboptimal value. Note the the logarithmic scale of $\tilde{\alpha}$.

Figure 4.8: Trajectories of prototypes of the system in the left panel of Fig. 4.7. Prototypes are projected on the space $\text{Span}(B_+, B_-)$ for $v_{\text{soft}} = 1$ (circle), 2 (triangle) and 5 (square).

generalization error.

On the other hand, a scenario with unequal class variances presents an interesting case where RSLVQ with global $v_{\text{soft}}$ fails to match the model. RSLVQ remains robust, i.e. the decision boundary converges to identical configurations for all settings of $v_{\text{soft}}$, see Fig. 4.8. However, the asymptotic results are suboptimal. While RSLVQ is insensitive to the priors of the clusters, its performance wrt. the best achievable error is sensitive to the cluster variances, e.g. at highly unbalanced $\sigma_+ / \sigma_-$, RSLVQ generalizes poorly and is outperformed by the simpler LVQ 2.1. In practical applications, $v_{\text{soft}}$ may be set locally for each prototype to accomodate such scenarios, but
4.6 Optimal window schedules

We have observed in sections 4.5.1 and 4.5.2 the learning curves and asymptotics of LVQ 2.1 and LFM-W wrt. fixed window parameters. In this section we treat the window parameter as dynamic properties during learning, viz. $\delta(\alpha)$. Although small windows allow optimal $\epsilon_g(\alpha \to \infty)$, their obvious disadvantage is their slower initial learning and convergence speed. This suggests that dynamic performance can be improved by adjusting the window along with the number of examples presented.

We calculate the locally optimal $\delta^*(\alpha)$-schedule by formally minimizing $d\epsilon_g/d\alpha$ with respect to $\delta$ using the knowledge of the input density and finding the condition

$$\delta^*(\alpha) = \arg \min_\delta \left( u(\alpha) \cdot \frac{dO(\alpha)}{d\alpha} \right) = 0 \text{ with } u(\alpha) = \frac{M}{\sum_{\sigma=1}^M p_{\sigma} \frac{d\epsilon_{g,\sigma}(\alpha)}{dO}}$$

(4.24)

where we use the shorthand $O$ for the set of order parameters. For a system with two prototypes $\{w_+, w_-\}$ and two clusters $\sigma = \{+, -\}, O = \{R_{++}, R_{+-}, R_{-+}, R_{--}, Q_{++},$
Q_+− , Q_−− )^T and derivating from (4.21), we obtain

\[
\frac{d\epsilon_g(\alpha)}{d\Omega} = \frac{1}{2\sqrt{v_\sigma}\Delta_q} \phi \left( \frac{Z_\sigma}{2\sqrt{v_\sigma}\Delta_q} \right) \cdot A_\sigma \quad \text{with}
\]

\[
A_+ = \begin{bmatrix}
-2\ell \\
0 \\
+2\ell \\
0 \\
1 - Z_+/(2\Delta_q^2) \\
Z_+/(\Delta_q^2)
\end{bmatrix}, \quad A_- = \begin{bmatrix}
0 \\
+2\ell \\
0 \\
-2\ell \\
-1 - Z_-/(2\Delta_q^2) \\
Z_-/(\Delta_q^2)
\end{bmatrix}
\]

with \( Z_\sigma = Q_{\sigma\sigma} - Q_{-,\sigma} - 2\ell(R_{\sigma\sigma} - R_{-,\sigma}) \) and \( \Delta_q \) defined in Eq. (4.23), see Appendix B.3 for the calculations.

We plug in \( d\Omega/d\alpha \) for the corresponding algorithm and numerically calculate \( \delta^*(\alpha) \) from Eq. (4.24) at each learning step. We find that the learning curve is improved with initially large \( \delta \) which is decreased during training, following the curve in Fig. 4.9. This suggests that practical schedules with gradual reduction of window sizes are indeed suitable for this particular learning problem.

While this approach locally minimizes generalization error, this strategy does not always lead to minimization of \( \epsilon_g \) over a time span, i.e. a globally optimal schedule, which requires calculations along the lines of variational optimization, see e.g. (Biehl 1994, Saad and Rattray 1997) for its application of optimal learning rates in multilayered neural networks. Obviously, a priori knowledge of the input density is not available in practical situations. Nevertheless, this minimization technique provides an upper bound of the achievable performance of the learning scheme for a given model.

Figure 4.7 displays that although large \( v_{\text{soft}} \) for RSLVQ allows for a faster initial learning, it also can yield non-monotonic learning curves. We can avoid the non-monotonic behavior and maximize the decrease of \( \epsilon_g \) by applying a variational approach analogous to (4.24) in order to calculate the locally optimal softness parameter schedule \( v_{\text{soft}}^*(\alpha) \). While fixing the value of \( \eta/v_{\text{soft}} \), we produce the locally optimal softness schedule \( v_{\text{soft}}^*(\alpha) \) in Fig. 4.9 where \( v_{\text{soft}}^*(\alpha) \) is initially large and decreases to saturate at a constant value. Note that this value depends on the learning rate, e.g. it decreases with \( \eta/v_{\text{soft}} \). In calculations with \( \eta \to 0 \), we obtain the limit \( v_{\text{soft}}^*(\infty) \to 0 \), which is the clearly suboptimal LFM. Therefore an analysis of optimal RSLVQ schedule requires \( \eta > 0 \).
4. Window-based example selection in learning vector quantization

Figure 4.10: Left panel: Snapshot at $\alpha = 50$ of an LVQ 2.1 system, $\delta = 1$ with $K = 3$ and $M = 6$ randomly generated isotropic clusters projected on the $(B_1, B_3)$ subspace. The solid dot marks the initial position of all prototypes and solid lines mark the trajectories of the prototypes. Right panel: $p_\triangle = 0.5, p_\Box = 0.3, p_\circ = 0.2$. Solid lines represent, from bottom to top, prototype vector lengths $Q_{11}, Q_{22}, Q_{33}$ for LVQ2.1 $\delta = 10$. Dashed lines represent the result for RSLVQ $v_{\text{soft}} = 2$.

4.7 Three-prototype systems

In this section we look at more generic analyses of LVQ algorithms by extending the previous systems to $K = 3$ prototypes and $M$ clusters, requiring a much larger set of order parameters. This allows an initial study on two important issues concerning practical applications of LVQ: multi-class problems and the use of multiple prototypes within a class.

We first look at multi-class problems with $Y = 3$ classes, an example is shown in Fig. 4.11 for LVQ 2.1 with $M = 6$ clusters selected with random variances and random deviation from the original class centers. The clusters are separable only in $M$ out of $N$ dimensions. In all our observations, we find that the behaviors of $K = 3$ systems are qualitatively similar to $K = 2$ systems. For LVQ 2.1, the learning curves vary according to the window sizes, but its asymptotic generalization error is independent of $\delta$. Due to the presence of other prototypes, the repulsion on a weaker class prototype are reduced. However, the prototypes remain divergent, e.g. Fig. 4.11. Meanwhile for LFM-W, the asymptotic performance is sensitive to $\delta$ whose range of effective window sizes depend strongly on the learning parameters. For GLVQ, the prototypes are divergent with a higher asymptotic error than LVQ 2.1, and thus it performs poorly. Finally, for RSLVQ, the prototypes remain stable and the asymptotic generalization performance is robust wrt. settings of $v_{\text{soft}}$, but it is outperformed by LVQ 2.1. Hence, the results are consistent with the $K = 2$ system.
4.7. Three-prototype systems

Figure 4.11: Unspecialized phase induces long learning plateaus, shown with LFM-W $K = 3, c_K = \{\pm 1\}$ and input density $M = 6$ and $Y = 2, c_\sigma = \{\pm 1\}$. Left panel: Several order parameters display a specialization phase between prototypes of same class. Right panel: Generalization error.

and the preceding analysis is valid qualitatively to, at least, systems of $M$ clusters and one prototype per class within the model restrictions.

To allow more complex decision boundaries, practical LVQ applications frequently employ several prototypes within a class. We investigate a two-class system $Y = 2, y_\sigma = \{+,-\}$ using $K = 3$ prototypes with labels $c_S = \{+,-\}$ and observe the non-trivial interaction between similarly labeled prototypes, here $w_1$ and $w_2$. While prototypes of different classes immediately separate in the initial training phase, prototypes of the same class remain identical in the $M$ dimensional space, see Fig. 4.11. The latter prototypes differ only in dimensions which are not related for classification and produce a suboptimal decision boundary. This may proceed for a long learning period before these prototypes begin to specialize, i.e. each prototype produces a bigger overlap $R_{S\sigma}$ with a distinct group of clusters. The specialization phase produces a sudden decrease of $\epsilon_g$, displayed in the right panel of Fig. 4.11. This phenomenon is highly reminiscent of symmetry breaking effects observed in unsupervised learning, such as Winner-Takes-All vector quantization (VQ) (Biehl 1994, Witoelar et al. 2008) or multilayer neural networks (Saad and Solla 1995).

Learning parameters highly influence the nature of the transition, e.g. large learning rates and smaller windows prolong the unspecialized phase, and therefore they are critical to the success of learning. Symmetry breaking may require exceedingly long learning times, resulting in learning plateaus which dominate the training process and present a challenge in practical situations with very high dimensional data. In more extreme circumstances, the system may not escape the unspecialized state
Table 4.1: Asymptotic properties of LVQ algorithms

<table>
<thead>
<tr>
<th></th>
<th>LVQ 2.1</th>
<th>LFM-W</th>
<th>GLVQ</th>
<th>RSLVQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>divergent</td>
<td>convergent*</td>
<td>divergent</td>
<td>convergent</td>
</tr>
<tr>
<td>Sensitivity wrt. parameters</td>
<td>robust</td>
<td>dependent</td>
<td>dependent</td>
<td>robust</td>
</tr>
<tr>
<td>Gen. ability</td>
<td>optimal</td>
<td>suboptimal</td>
<td>suboptimal</td>
<td>suboptimal</td>
</tr>
</tbody>
</table>

* under the condition that $\delta$ is larger than critical window size $\delta_c$.

at all and the optimal classification cannot be obtained. Details of the symmetry breaking properties wrt. parameters will be investigated in following publications.

### 4.8 Conclusion

We have investigated the learning behavior of LVQ 2.1, GLVQ, LFM-W and RSLVQ using window schemes which work in high dimensions. The analysis is based on the theory of on-line learning on a model of high dimensional isotropic clusters. Our findings demonstrate that the selection of proper window sizes is critical to efficient learning for all algorithms. Given more available data and allowance for costly learning times, parameter selection becomes much less important.

Our analysis demonstrates the influence of windows on the learning curves and the advantages and drawbacks of each algorithm within the model scenarios. A summary is described in Table 4.1. Asymptotically, LVQ 2.1 achieves optimal performance in all scenarios, but stability remains an issue in terms of diverging prototypes. LFM-W shows a remarkable improvement in performance over LFM. Unfortunately, the introduction of a window may also influence its stability, and therefore it is highly parameter sensitive, i.e. only a narrow range of window size can improve the overall performance. GLVQ behaves similarly to LVQ 2.1. While GLVQ reduces the initial strong overshooting of LVQ 2.1, the prototypes remain divergent and GLVQ produces higher generalization errors or long convergence times. RSLVQ attempts to combine the advantages of both LFM and LVQ 2.1 by providing both stability and optimal performance. However, an important issue of RSLVQ lies on its approximation of the data structure, e.g. it performs well when the actual input density are isotropic Gaussian clusters with equal variance. If the assumptions depart from the input density, the results become suboptimal and RSLVQ can even be outperformed by the simpler LVQ 2.1 and LFM-W. In all scenarios, RSLVQ displays robustness of its classification behavior with respect to the softness parameter, given sufficiently low learning rates.

This analysis also allows a formal optimization of the window size during learn-
ing to ensure fast convergence. While in general, various window sizes for LVQ 2.1 produce equal asymptotic errors, initial window sizes should be chosen large for faster convergence speed and decreased in the course of learning. Similarly, an optimal schedule for RSLVQ points to a gradual decrease of the softness parameter to a particular saturation value, which agrees well with many practical scheduling schemes. However, locally optimal schedules do not always lead to the globally optimal schedules, see for instance (Saad and Rattray 1997). In further work, we will develop efficient dynamic parameter adaptations, i.e. optimal window schedules during on-line training along the lines of variational optimization.

We show that the analysis remains valid for multi-class systems and arbitrary number of isotropic clusters. Additionally, using multiple prototype assignments within a class, we already observe the presence of learning plateaus in this highly simplified scenario. These phenomena carry on and could dominate the training process in any practical situations with high degrees of freedom. Further investigations of more complex network architectures and non-trivial input distributions may also yield additional phenomena, e.g. competing stationary states of the system, and provide further insights to general LVQ behaviors.
Chapter 5

Phase transitions in off-line vector quantization and neural gas

Abstract

The statistical physics of off-learning is applied to Winner-Takes-All and rank-based vector quantization (VQ), including the Neural Gas (NG). The analysis is based on the limit of high training temperatures and the annealed approximation. The typical learning behavior is evaluated for systems of two and three prototypes with data drawn from a mixture of high dimensional Gaussian clusters. The learning curves exhibit phase transitions, i.e. a critical or discontinuous dependence of performances on the training set size and training temperature. We show how the nature and properties of the transition depend on the number of prototypes and the control parameter of rank based cost functions. The NG based systems are demonstrated to give an advantage over WTA in terms of robustness to initial conditions.

5.1 Introduction

Vector Quantization (VQ) is an important family of unsupervised learning algorithms and has been applied in many fields, e.g. data mining, image compression and clustering problems (Neural Networks Research Centre, Helsinki 2002). The main objective of VQ is to represent the data faithfully by a small number of prototypes or codebook vectors, measured by the quantization error.

Competitive learning schemes such as the basic “winner-takes-all” (WTA) approach or batch variants such as the k-means algorithm aim at direct minimization of the quantization error. However, such methods are susceptible to confinement in local minima, leading to far from optimal performance. Numerous extensions and modifications have been proposed in order to overcome this difficulty: the self-organizing map (SOM) (Kohonen 1997), fuzzy-k-means (Bezdek 1981) and neural gas (Martinetz et al. 1993), to name just a few. These algorithms have in common that each data point is assigned to more than one prototype. In particular, NG algorithms replace the quantization error by related rank based cost functions (Martinetz et al. 1993).
In general, two different adaptation schemes are investigated to achieve the objective: online learning, which adapt the prototypes on a sequence of single example data, and off-line or batch learning which adapt the prototypes according to all examples at once. Off-line approaches are shown to perform faster for high dimensional data, see e.g. (Bermejo and Cabestany 2000, Cottrell et al. 2006); however, the system remains to be more susceptible to local minima. In previous studies, e.g. (Witoeaar et al. 2008) we have addressed the dynamics of on-line schemes, specifically the analysis of three-prototype VQ and NG systems in Chapter 3. Here, we consider training from a set of examples by means of off-line or batch stochastic optimization of a cost function.

To this end, we apply methods from the equilibrium physics of learning which were earlier used to study, amongst others, feed-forward neural networks (Engel and van den Broeck 2001, Seung et al. 1992, Watkins et al. 1993). The approach allows us to investigate the typical behavior of off-line VQ learning schemes in non-trivial model situations.

Our analysis is based on the so-called annealed approximation which has proven to yield valuable insights into many training scenarios. In particular, it becomes exact in the limit of high training temperatures, which allows for a simplifying description of qualitative behavior. The theory and several applications of annealed approximation and high-temperature limit can be found in, e.g., (Biehl et al. 1998, Engel and van den Broeck 2001, Seung et al. 1992, Solla and Levin 1992, Watkins et al. 1993).

Our analysis of Vector Quantization and Neural Gas shows how invariances with respect to the permutation of prototypes lead to phase transitions which govern the training process: A critical number of examples is required for the successful detection of the underlying structure. Similar effects of "retarded learning" have been studied in several models and learning scenarios earlier, e.g. (Biehl et al. 1998, Buhot et al. 2002, Herschkowitz and Opper 2001, Lootens and van den Broeck 1995, Watkin and Nadal 1994).

Extending earlier studies of off-line competitive learning, see (Engel and van den Broeck 2001, Lootens and van den Broeck 1995) for an example and further references, we consider rank based training and scenarios with more than two prototypes. We show that the nature of the transition can change significantly under these modifications. Here we consider the extensions to rank based training and scenarios with more than two prototypes. We show that the nature of the transition can change significantly under these modifications.

In section 5.2 we describe the cost functions minimized in the respective VQ learning schemes. This includes the basic WTA and rank-based cost functions. The high-dimensional model data is explained in Chapter 2 while section 5.3 briefly
describes the analysis in the equilibrium physics framework. Sections 5.4 and 5.5 present the obtained results for systems with two and three prototypes with emphasis on phase transitions in the learning curves. A summary and outlook are given in section 5.6.

5.2 Vector Quantization Cost Functions

We study training processes where the unlabeled input vectors $\xi^\mu$ in the data set $\mathcal{D} = \{\xi^\mu\}_{\mu=1}^P$ are generated independently according to the model density given in Eq. (2.5). We will exploit the thermodynamic limit $N \to \infty$ and assume that the number of examples also grows linearly in $N$, i.e., $P \propto N$. In this chapter, we specifically consider a mixture of two spherical Gaussian clusters with unit variance:

$$P(\xi) = \sum_{\sigma=1}^2 p_\sigma P(\xi|\sigma)$$

where the prior weights satisfy $p_1 + p_2 = 1$. The cluster centers are given by $\ell B_1$ and $\ell B_2$ with the separation parameter $\ell$. Without loss of generality, we assume that the $B_m$ are orthonormal with $B_m \cdot B_n = \delta_{mn}$. We refer the discussion of this input density to Chapter 2.

We consider a system of $K$ prototype vectors $\mathbf{W} = \{\mathbf{w}_S \in \mathbb{R}^N\}_{S=1}^K$ with $K \ll P$. The cost functions considered here can be expressed as empirical averages of an error measure:

$$H(\mathbf{W}) = \sum_{\mu=1}^P e(\mathbf{W}, \xi^\mu)$$

with

$$e(\mathbf{W}, \xi^\mu) = \frac{1}{2} \sum_{k=1}^K d(\mathbf{w}_S, \xi^\mu) g(r_S) - \frac{1}{2} (\xi^\mu)^2.$$  (5.2)

Here the last term is constant w.r.t. the choice of $\mathbf{W}$ and is subtracted for convenience in later calculations. Throughout the following, we employ the squared Euclidean distance measure $d(\mathbf{w}_S, \xi^\mu) = (\xi^\mu - \mathbf{w}_S)^2$. In Eq. (5.2), the normalization $\sum_{S=1}^K g(r_S) = 1$ of the so-called rank function $g$ is assumed. The argument $r_S$ is the rank of prototype $\mathbf{w}_S$ with respect to its distance from input vector $\xi$. It can be written as

$$r_S = K - \sum_{T \neq S} \Theta_{TS}$$  (5.3)
with the shorthand $\Theta_{TS} = \Theta(d(\xi, w_T) - d(\xi, w_S))$ where $\Theta(.)$ is the Heaviside function. Specifically, we consider rank functions of the form

$$g_\lambda(r_S) = \frac{\exp[-r_S/\lambda]}{\sum_{k=1}^K \exp[-r_k/\lambda]}.$$  \hspace{1cm} (5.4)

where $\lambda$ controls the soft assignment of a given vector $\xi$ to the prototypes. In the limit $\lambda \to 0$, it becomes WTA, i.e. only the winner $w_J$ with $r_J = 1$ is taken into account, $g_\lambda(k) = \delta_{k,1}$. The costs, Eq. (5.2), reduce to the quantization error with

$$e_{VQ}(W, \xi) = \frac{1}{2} \sum_{S=1}^K d(w_S, \xi) \prod_{T \neq S} \Theta_{ST} - \frac{1}{2} \xi^2.$$  \hspace{1cm} (5.5)

Note that the cost functions considered here are invariant under exchange or permutations of prototypes. This is different from supervised learning where prototypes and data vectors carry class labels. In Learning Vector Quantization (Biehl et al. 2007, Kohonen 1997, Neural Networks Research Centre, Helsinki 2002), for instance, the permutation symmetry holds only within the classes.

### 5.3 Equilibrium Physics Approach

In the by now standard statistical physics analysis of off-line learning (Seung et al. 1992, Watkins et al. 1993), training is interpreted as a stochastic minimization of $H(W)$ on the data set $D$, where the formal temperature $T$ controls the degree of randomness. We provide an overview of the analysis as follows:

1. Determining the stationary densities of given prototype configurations from the stochastic process, giving the partition sum,

2. Calculating the average free energy function over thermal noise (stochastic training), derived from the logarithm of the partition sum ,

3. Calculating the average over the disorder in the random data sets using the high temperature limit or the annealed approximation,

4. Describing the system in terms of order parameters and finding the configurations of order parameters which dominate the distribution using the saddle-point method.

We elaborate on these steps in the following. Under the influence of thermal noise, the state $W$ of the system varies along learning time. In the initial stages or after a change of parameters, quantities which depend on the state, for instance
5.3. Equilibrium Physics Approach

$H(W)$, also fluctuate along a trend, such as decreasing $H(W)$. After a while, this trend ceases and the quantities fluctuate around constant average values, when the system reaches a well-defined thermal equilibrium: a configuration $W$ is observed with a probability given by the Gibbs density

$$P(W) = \exp[-\beta H(W)]/Z$$

(5.6)

Here $\beta = 1/T$ is the inverse temperature, the normalization $Z$ is called the partition sum and the measure $d\mu(W)$ is the $KN$-dim. volume element. The formal temperature controls the quality of minimization: low $T$ corresponds to densities with a sharp peak around the minima of $H(W)$, while high $T$ corresponds to very broad distribution of $W$ and little learning on the form of $H(W)$. The interpretation of our results can be based on the observation that Eq. (5.6) corresponds to the stationary density of $W$ under a so-called Langevin dynamics for well-behaved differentiable energies $H(W)$:

$$\partial W/\partial t = -\nabla_W H(W) + \Gamma(t),$$

(5.7)

see (Sompolinsky and Tishby 1990, Watkin et al. 1993) for a discussion in the context of learning. Here, $\Gamma(t)$ is a $KN$-dim. vector of $\delta$-correlated white noise:

$$\langle \Gamma_i(t)\Gamma_j(t') \rangle = 2T \delta_{ij} \delta(t - t').$$

(5.8)

The approach outlined above yields properties of the stationary density $P(W)$ resulting from (5.7), on average over the data set contained in $H(W)$. Practical algorithms will not have precisely the form (5.7), but one can expect that our results carry over, qualitatively, to more general learning schemes that are guided by the stochastic minimization of $H(W)$. The stationary density in Eq. (5.6) is also valid for non-differentiable forms of $H(W)$.

In the statistical mechanics approach, thermal averages $\langle \cdots \rangle$ over $P(W)$ can be calculated as derivatives of the so-called free energy $-\ln Z/\beta$, for instance:

$$\langle H \rangle = \frac{1}{Z} \int d\mu(W) H(W) \exp(-\beta H(W)) = -\frac{\partial}{\partial \beta} \ln Z$$

(5.9)

Note that this type of average describes the system trained on one specific data set. In order to obtain generic properties of the model scenario, an additional average over all possible $D$ is performed, yielding the so-called quenched free energy (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993)

$$F = -\langle \ln Z \rangle_D /\beta.$$  

(5.10)

Proper derivatives thereof yield quantities of interest on average over the randomness contained in $D$ and over the stochastic outcome of the training process. In general, the computation of $\langle \ln Z \rangle_D$ requires involved techniques from the theory of
disordered systems such as the replica method (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993)

The analysis of thermal equilibrium does not directly correspond to the application of a particular learning algorithm in practical situations. However, it relates to the use of a specific cost function which guides a stochastic training process.

We discuss two important simplifying approaches: the high temperature limit and the annealed approximation.

5.3.1 High temperature limit

A common technique to perform the quenched average $\langle \ln Z \rangle_{\mathcal{D}}$ at all $T$ is the so-called replica method (Edwards and P.W 1975), which becomes greatly simplified in the limit of high $T$. For the final form of $\langle \ln Z \rangle_{\mathcal{D}}$ without the mathematical intricacies, we advise the reader to proceed to Eq. (5.19).

Replica method

The replica method exploits the relation

$$\langle \ln Z \rangle_{\mathcal{D}} = \lim_{n \to 0} \frac{1}{n} \ln \langle Z^n \rangle_{\mathcal{D}}.$$  

(5.11)

For integer $n$, $Z^n$ corresponds to the partition sum of $n$ non-interacting replicas, i.e. identical copies of the system labeled $\gamma = 1, 2, \ldots, n$. The following steps are to compute $\langle Z^n \rangle_{\mathcal{D}}$ and analytically continue to $n \to 0$. For mathematical and conceptual subtleties, we refer the interested reader to e.g. (Mezard et al. 1987). Using Eqs. (5.7) and (5.11), we obtain

$$\langle Z^n \rangle_{\mathcal{D}} = \int \prod_{\gamma} d\mu(\mathbf{W}^\gamma) \exp \left[ -\beta \sum_{\gamma=1}^{n} H(\mathbf{W}^\gamma) \right],$$  

(5.12)

where $H(\mathbf{W}^\gamma)$ has to calculated for each replica. Replacing the $H(\mathbf{W}^\gamma)$ by an effective energy in the replicated space, we can rewrite the equation above as

$$\langle Z^n \rangle_{\mathcal{D}} = \int \prod_{\gamma} d\mu(\mathbf{W}^\gamma) \exp \left[ -\beta H_{\text{eff}} \left( \{ \mathbf{W}^\gamma \}_{\gamma=1}^{n} \right) \right],$$  

(5.13)

with $H_{\text{eff}} (\{ \mathbf{W}^\gamma \}) = -\frac{1}{\beta} \ln \int d\mu(\xi) \exp \left[ -\beta \sum_{\gamma=1}^{n} \sum_{\mu=1}^{P} e(\mathbf{W}^\gamma, \xi^\mu) \right]$.  

(5.13)

where $\mu(\xi)$ is the normalized distribution from which the examples are chosen. Because we assume that the examples are generated independently, we can treat
5.3. Equilibrium Physics Approach

\langle \cdots \rangle_{ID} \text{ separately for } \mu = 1, 2, \ldots, P. \text{ We obtain}

\begin{equation}
H_{\text{eff}}(\{W^\gamma\}) = -\frac{P}{\beta} \ln \int d\mu(\xi) \exp \left[ -\beta \sum_{\gamma=1}^{n} e(W^\gamma, \xi) \right]. \tag{5.14}
\end{equation}

Plugging the above into Eq. (5.13), we write the final form of \( \langle Z^n \rangle_{ID} \) as

\begin{equation}
\langle Z^n \rangle_{ID} = \int \left[ \prod_{\gamma} d\mu(W^\gamma) \right] \exp \left[ -P G(\{W^\gamma\}) \right]
\end{equation}

with

\begin{equation}
G(\{W^\gamma\}) = -\ln \int d\mu(\xi) \exp \left[ -\beta \sum_{\gamma=1}^{n} e(W^\gamma, \xi) \right]. \tag{5.15}
\end{equation}

where \( G(\{W^\gamma\}) \) does not depend on the number of examples, but on the forms of \( e(W^\gamma, \xi^\mu) \) and \( d\mu(\xi) \). The different replicas interact in \( G(\{W^\gamma\}) \) and the computations are generally difficult.

**High-\( T \) limit**

Here we resort to training at high temperatures which allows us to use simplifying relations in the limit \( \beta \to 0 \). This limit has proven to provide important insights into a variety of learning scenarios (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993). Non-trivial results can only be expected if the increased noise is compensated for by a larger number of examples \( P \), which scales like

\begin{equation}
P = \hat{\alpha} \left( \frac{N}{\beta} \right). \tag{5.16}
\end{equation}

Because large training sets sample the model density very well, the empirical average \( \frac{1}{P} \sum_{\mu} e(W, \xi^\mu) \) can be replaced by \( \langle e \rangle_{\xi} \), i.e. an average over the full \( P(\xi) \). Consequently, training set and test set performances coincide in this simplifying limit. Furthermore, we expand in \( \beta \) and exploit \( \exp \left[ -\beta \sum_{\gamma} e(W^\gamma, \xi) \right] \approx 1 - \beta \sum_{\gamma} e(W^\gamma, \xi) \) for \( \beta \to 0 \). so that Eq. (5.15) becomes

\begin{equation}
\langle Z^n \rangle_{ID} = \int \left[ \prod_{\gamma} d\mu(W^\gamma) \right] \exp \left[ -\beta P \sum_{\gamma=1}^{n} \langle e(W^\gamma, \xi^\mu) \rangle_{\xi} \right]. \tag{5.17}
\end{equation}

An important consequence of the high-\( T \) limit is that the replicas become uncoupled, see e.g. (Seung et al. 1992). The \( n \)-fold integral over \( \prod_{\gamma} d\mu(W^\gamma) \) are simply \( n \) multiplications between identical one-fold integral over \( d\mu(W) \). Therefore, we obtain

\begin{equation}
\langle Z^n \rangle_{ID} = \left( \int d\mu(W) \exp \left[ -\beta P \langle e(W, \xi^\mu) \rangle_{\xi} \right] \right)^n \tag{5.18}
\end{equation}
5. Phase transitions in off-line vector quantization and neural gas

Using the rescale $\tilde{\alpha} = \beta (P/N)$ as in (5.16), and inserting (5.18) into the replica relations (5.11), we obtain the form

$$\langle \ln Z \rangle_{I D} = \lim_{n \to 0} \frac{1}{n} \ln \left( \int d\mu(W) \exp \left[ -\tilde{\alpha} N \langle e(W, \xi) \rangle_{\xi} \right] \right)^n$$

$$= \ln \int d\mu(W) \exp \left[ -\tilde{\alpha} N \langle e(W, \xi) \rangle_{\xi} \right],$$

(5.19)

where the rescaled number of examples $\tilde{\alpha}$ plays the role of an effective inverse temperature and $N \langle e \rangle_{\xi}$ is the extensive energy of the system.

**Order parameters**

We now apply the thermodynamic limit to fully describe the system in terms of macroscopic properties. The mean cost $\langle e \rangle_{\xi}$ for high dimensional data can now be expressed, at least numerically, as a function of the order parameters as defined in Eq. (3.8)

$$R_{S\sigma} = w_S \cdot B_\sigma \quad \text{and} \quad Q_{ST} = w_S \cdot w_T,$$

(5.20)

see Appendix B.4 for the results and details of the calculation. The set of quantities (5.20) represents the structure imposed by the cluster center vectors $B_\sigma$. By inserting the integral over the order parameters, Eq. (5.19) becomes

$$\langle \ln Z \rangle_{I D} = \ln \int d\mu(W) \int \prod_{S,\sigma} dR_{S\sigma} \left[ \prod_{S,T \leq T} dQ_{ST} \right] \prod_{S,\sigma} \delta (R_{S\sigma} - w_S \cdot B_\sigma) \times \prod_{S,T \leq S} \delta (Q_{ST} - w_S \cdot w_T) \exp \left[ -\tilde{\alpha} N \langle e \rangle_{\xi} \right]$$

(5.21)

We introduce the entropy term $s$ which represents the phase space volume corresponding to a particular setting of order parameters.

$$s \{ R_{S\sigma}, Q_{ST} \} = \frac{1}{N} \ln \int d\mu(W) \prod_{S,\sigma} \delta (R_{S\sigma} - w_S \cdot B_\sigma) \prod_{S,T \leq S} \delta (Q_{ST} - w_S \cdot w_T)$$

$$= \frac{1}{2} \ln \det C + c$$

(5.22)

where $c$ is independent of the order parameters and $C$ is the $(K + 2)$-dim. square matrix of self- and cross-overlaps of the $\{ w_S, B_\sigma \}$ as given by Eq. (5.20) and $B_\sigma \cdot B_\rho = T_{\sigma\rho}$, see (Ahr, Biehl and Urbanczik 1999, Witoelar and Biehl 2008) for details. We
substitute (5.22) into (5.21), to obtain the final form

$$
\langle \ln Z \rangle_{\mathcal{D}} = \ln \int \left[ \prod_{S,\sigma} dR_{S\sigma} \right] \left[ \prod_{S,T \leq S} dQ_{ST} \right] \exp \left[ -N f (\{ R_{S\sigma}, Q_{ST} \}) \right], \tag{5.23}
$$

where $f$ is called the free energy function, written as

$$
f (\{ R_{S\sigma}, Q_{ST} \}) = \hat{\alpha} \langle e \rangle_\xi - s (\{ R_{S\sigma}, Q_{ST} \}). \tag{5.24}
$$

The right hand side can be obtained as a function of the order parameters in closed form. We can use the saddle-point method to evaluate (5.23) in the limit of large $N$. For $N \to \infty$, the integral is dominated by the maximum integrand, i.e. the minimum of $f$. The quenched free energy becomes

$$
- \langle \ln Z \rangle_{\mathcal{D}}/N = \beta \min f (\{ R_{S\sigma}, Q_{ST} \}). \tag{5.25}
$$

Hence, given a specific cost function and training set size $\hat{\alpha}$, we obtain the typical equilibrium properties of the system by minimizing the free energy function $f (\{ R_{S\sigma}, Q_{ST} \})$ with respect to the order parameters. The corresponding $\{ R_{S\sigma}, Q_{ST} \}$ describe the typical properties of the configurations that dominate the Gibbs ensemble.

### 5.3.2 Annealed Approximation

Practical training procedures aim at an efficient minimization of the cost function. In the statistical physics interpretation of the learning process, this corresponds to low temperatures. While the correct treatment of finite $T$ requires sophisticated techniques such as the replica trick, a useful approximation method which is technically less difficult, can be employed to perform the quenched average $\langle \ln Z \rangle_{\mathcal{D}}$.

In the so-called annealed approximation, $\langle \ln Z \rangle_{\mathcal{D}}$ is approximated by the logarithm of the averaged $Z$ instead. It is equivalent to the approximation

$$
\left\langle \frac{\exp (-\beta H(W))}{Z} \right\rangle_{\mathcal{D}} \approx \left\langle \frac{\exp (-\beta H(W))}{Z} \right\rangle_{\mathcal{D}} = \left\langle \frac{\exp (-\beta H(W))}{Z} \right\rangle_{\mathcal{D}} \tag{5.26}
$$

The annealed approximation becomes exact in the limit $\beta \to 0$ and coincides with the explicit treatment of this limit (Seung et al. 1992). At low temperatures the annealed free energy yields only an upper bound to the correct one, but the hope is that the position of minima in terms of the $\{ R_{S\sigma}, Q_{ST} \}$ is similar. The scheme has proven useful in predicting qualitative behavior of many learning systems, e.g. (Engel and van den Broeck 2001, Sompolinsky and Tishby 1990). The validity of the
annealed approximation is discussed systematically in, for instance, (Seung et al. 1992, Solla and Levin 1992).

The average partition function can be rewritten as

$$\langle Z \rangle_D = \int d\mu(W) \exp[-\alpha NG_A(W)]$$

with

$$G_A = -\ln \langle \exp(-\beta e(\xi^\mu, W)) \rangle_\xi$$

(5.27)

where $G_A$ involves an average over one random input only. Only for $\beta \to 0$ this average can be absorbed into the exponent and we recover the high temperature result.

The calculation of $G_A$ can be done analytically for two prototypes and arbitrary $\beta$, as outlined in (Witoelar and Biehl 2008). The corresponding free energy function as in (5.24) for the annealed approximation is

$$f = \alpha G_A - s(R_{SS}, Q_{ST})$$

(5.28)

where the rescaled number of examples $\alpha = P/N$ is independent of $\beta$. Unlike the high temperature limit, in the annealed approximation the empirical average $1/P \sum_{\mu=1}^P e(W, \xi^\mu)$ training set $D$ is distinguished from $\langle e \rangle_\xi$. The training set performance is given by

$$e_{\text{train}} = \alpha^{-1} \frac{\partial}{\partial \beta} f(R_{SS}, Q_{ST})$$

(5.29)

which has to be evaluated in the minimum of $f$.

### 5.4 Two-prototype systems

Here we discuss typical properties of the considered model situations as computed in the statistical physics framework. We first concentrate on the system with only two prototypes, which already displays non-trivial phenomena. Furthermore, significant differences between WTA and NG training can be observed.

Most of the discussion will be in terms of the high-temperature limit. We show, however, that the extension to lower temperatures by means of the Annealed Approximation gives similar results, qualitatively.

#### 5.4.1 Relevant configurations and minima of $f$

In a system with two prototypes, successful learning should lead to the representation of each cluster by one of the $w_s$. However, our analysis of WTA training shows
that this type of configuration competes with two other settings in thermal equilibrium. Figure 5.1 displays a sketch of all relevant situations: In configurations of type (a) only one of the prototypes is placed near the clusters while the second one diverges to infinity. Case (b) displays a situation with both prototypes in the region of high density but unspecialized, i.e. the specialization factor

$$\Delta_\sigma = |R_{1\sigma} - R_{2\sigma}|$$

(5.30)
is zero for all \(\sigma\). Panel (c) represents prototype configurations with \(\Delta_\sigma > 0\) which we will refer to as specialized.

We first investigate the role of the trivial situation, (a), in the thermal equilibrium of WTA systems. In Fig. 5.2 (a) we have used the squared length \(Q_{22}\) of the second prototype as a parameter, while the free energy function is minimized with respect to all other order parameters. For all \(\hat{\alpha}\), the global minimum of \(f\), i.e. the true equilibrium state, corresponds to a trivial configuration: Only \(w_1\) contributes significantly to the energy \(\hat{\alpha}(e)\xi\), while \(w_2\) does not represent any data as \(Q_{22} \to \infty\). Note that the entropy \(s\) grows like \(\ln Q_{22}\) while the energy approaches a constant value in this limit. Consequently, a trivial minimum of type (a) in Fig. 5.1 with \(f \to -\infty\) will always be present.

For large enough data sets, a local minimum appears at smaller \(Q_{22}\) where both prototypes play a non-trivial role, e.g. \(\hat{\alpha} = 1, 2\) in Fig. 5.2 (a). This corresponds to illustration (b) in Fig. 5.1. Both prototypes \(w_{1,2}\) coincide in the space spanned by \(B_{1,2}\), differences in the \((N - 2)\)-dim. orthogonal space are reflected by non-trivial configurations of \(\{Q_{ST}\}\).

Figure 5.1: Relevant configurations in the WTA scenario with two prototypes. The darker circle symbolizes the Gaussian cluster with larger prior weight. Graph (a) displays a configuration with one of the prototypes diverging. An unspecialized two-prototype configuration is shown in (b) and the optimal, specialized state is shown in panel (c). Note that the two prototypes in (b) would indeed coincide in the projection but are separated in the \((N - 2)\)-dim. orthogonal space.
5. Phase transitions in off-line vector quantization and neural gas

Figure 5.2: Free energy function \( f \) vs. \( Q_{22} \) in a two-prototype system at \( \beta \to 0 \). The cost functions are (a) WTA for \( \hat{\alpha} = 2, 1 \) and 0.1, (b) NG with \( \lambda = 0.5, 0.3, 0.2 \) and 0 at \( \hat{\alpha} = 1 \). For both cases, \( p_1 = p_2 = 0.5 \), and \( \ell = 1 \).

of the free energy are also relevant from a practical point of view: in a dynamical system approaching equilibrium, they correspond to metastable states. The time to leave a metastable state increases with the height of the free energy barriers confining it. The system can be trapped in a metastable configuration and typical escape times become prohibitively long in large systems, see e.g. (Watkin et al. 1993) for a discussion.

This finding corresponds to the observation that initialization is highly important in practical applications of WTA-based systems. Given a start configuration, the system will approach and reside in the nearest stable or metastable state. Consequently, we expect the local minimum to be relevant in situations where the prototypes are prepared close to the clusters. On the contrary, prototypes initialized in regions with very low density of data will receive virtually no updates and do not contribute to the representation of data.

In rank based updates, as for instance in Neural Gas algorithms, the situation should be more favorable with respect to initialization issues. Here, all prototypes are updated even if they are far away from the presented data.

The corresponding high temperature analysis of NG training shows that the trivial minimum with \( Q_{22} \to \infty \) disappears for all \( \lambda > 0 \), as shown in Fig. 5.2(b). The limit \( \lambda \to 0 \) is identical to WTA. In NG, all prototypes contribute to the extensive energy by a term on the order \( g_\lambda(r_s)Q_{ss} \) which grows faster than entropy at large \( Q_{ss} \). The trivial minimum is replaced by a local or global minimum at large but finite \( Q_{22} \). The latter disappears completely at large enough values of the control parameter \( \lambda \). The corresponding characteristic values of \( \lambda \) depend on the training
set size \( \hat{\alpha} \), the cluster geometry and parameters of the model density. Hence, NG systems can be expected to be less sensitive to initialization in practice. For large enough \( \lambda \), the system is forced to place prototypes close to the cluster centers, cf. Fig. 5.2(b). This reflects the benefits of annealing schemes in practical training with, initially, large \( \lambda \) to ensure that all prototypes converge.

In practice, the trivial states could also be avoided by means of setting proper boundaries to \( Q_{SS} \) or imposing a normalization. The latter would correspond to methods of directional clustering.

### 5.4.2 Specialization transition in the training process

The model parameters and the size of the training set determine which of the above discussed configurations are observed.

We first investigate in greater detail the WTA cost function with \( \lambda = 0 \). For small \( \hat{\alpha} \), only trivial configurations, (a) in Fig. 5.1 are stable. Above a characteristic value of \( \hat{\alpha} \), a non-trivial unspecialized state of type (b) becomes metastable. We assume now that the system resides in such a configuration and that the divergence of the second prototype has been avoided.

Fig. 5.3 shows that for small values of \( \hat{\alpha} \) the prototypes remain unspecialized, i.e. \( \Delta_{\sigma} = 0 \) for all \( \sigma \), see Eq. (5.30). Both prototypes \( w_{1,2} \) coincide in the space spanned...
66  5. Phase transitions in off-line vector quantization and neural gas

Figure 5.4: Critical training set size $\hat{\alpha}_c$ as a function of the rank control parameter $\lambda$, cf. Eq. (5.4). The parameters are set at $\ell = 1$, $p_1 = 0.8$, $p_2 = 0.2$.

by $B_{1,2}$, while their differences in the $(N - 2)$-dim. orthogonal space are reflected by non-trivial configurations of $\{Q_{ST}\}$.

The underlying cluster structure is not at all detected as long as $\hat{\alpha}$ is smaller than the critical value $\hat{\alpha}_c$. This parallels findings for supervised learning in neural networks with two hidden units (Biehl et al. 1998) or unsupervised learning scenarios (Lootens and van den Broeck 1995, Watkin and Nadal 1994). Above $\hat{\alpha}_c$, prototypes begin to align with the clusters and the system becomes specialized, i.e. each $w_S$ has a larger overlap with exactly one of the cluster centers. Obviously, exchange of the prototypes would not alter the value of $H$ or $f$ and the two configurations are completely equivalent. In the continuous symmetry breaking transition, one of the two states is selected as signaled by a sudden power law increase of $\delta_\sigma$ for $\hat{\sigma} \geq \hat{\alpha}_c$. Fig. 5.3 (a) shows the dependence of the equilibrium values of $R_{11}$ and $R_{21}$ on $\hat{\alpha}$ in an example situation. The transition results in a non-differentiable kink in the learning curve $\langle e_{VQ} \rangle_\xi$ vs. $\hat{\alpha}$ as shown in Fig. 5.3 (b). The critical value depends on the model settings. For instance, $\hat{\alpha}_c$ will be larger for smaller $\ell$.

In summary, the generic behavior of the two-prototype WTA system is characterized by a sequence (a) → (b) → (c) with growing $\hat{\alpha}$ in terms of the illustrations in Fig. 5.1.

Next we investigate the minimization of rank based cost functions with $\lambda > 0$ in Eq. (5.2). We observe the same behavior as in WTA learning at sufficiently small rank function parameter $\lambda$. However, the critical value $\hat{\alpha}_c$ needed for prototype specialization and thus successful training, increases with $\lambda$, see Fig. 5.4. On the other hand, as discussed in Sec. 5.4.1 choosing large values of $\lambda$ has the advantage of avoiding the divergence of one of the prototypes. Note that the slope $d\hat{\alpha}_c/d\lambda = 0$
5.4. Two-prototype systems

Figure 5.5: Results for a WTA-based system with $K = 2$ using the annealed approximation. (a) The average errors for training set and test set as function of $\tilde{\alpha}$ for $\beta = 0.1$. (b) The scaled critical training set sizes $\tilde{\alpha}$ as function of $\beta$. The parameters of the input density are $p_1 = p_2 = 0.5, \ell = 1$.

for $\lambda \to 0$. Thus, performing rank based training with an appropriate annealing of $\lambda$ appears to be a promising strategy for practical optimization of the quantization error.

5.4.3 Annealed approximation

We have investigated the specialization transition in the WTA system with $K = 2$ using the annealed approximation. This approach allows us to study the learning behavior at finite temperatures. In the annealed approximation, inverse temperature $\beta$ and training set size $\alpha$ can be chosen independently and the performance with respect to training and test data are distinguished. Figure 5.5(a) shows the average training errors and test errors for $\beta = 0.1$. The difference between $e_{\text{train}}$ and $\langle e \rangle_\xi$ decreases with decreasing $\beta$ and coincides with the results for the high temperature limit as $\beta \to 0$.

The annealed approximation exhibits qualitatively similar behavior as the high temperature limit: A continuous phase transition exists from unspecialized to specialized states. The continuous phase transition results in a kink in the learning curve for both the training error and test error.

For small $\beta$, we observe that $\alpha_c \propto \beta^{-1}$ which confirms the scaling $\tilde{\alpha} = \beta \alpha$ in the limit $\beta \to 0$. Furthermore, figure 5.5(b) shows $\tilde{\alpha}_c$ for a wider range of temperature. The annealed approximation predicts that the specialization transition is still
relevant at finite temperatures, with $\hat{\alpha}_c$ decreasing with increasing $\beta$.

While the annealed approximation becomes exact in the limit $\beta \to 0$, it cannot be expected to describe the low temperature regime accurately. However, it has been frequently confirmed to reproduce qualitatively correct behavior (Engel and van den Broeck 2001, Sompolinsky and Tishby 1990). The exact calculation of quenched averages at low temperatures would require, for instance, the full replica approach. Here we conclude that the simplifying high temperature limit already gives reliable insight into the qualitative behavior of the model and proceed by applying it to more complex three prototype systems.

5.5 Three-prototype systems

The behavior of the phase transitions for $K = 3$ is qualitatively different compared to systems with $K = 2$. Figures 6 (1-6) show the relevant configurations which represent local or global minima of $f$. Depending on the cluster structure, for instance, the separation $\ell$, distinct transition scenarios are observed.

5.5.1 Small cluster separation

The first example we analyze is a WTA system with $p_1 = 0.8, p_2 = 0.2$ and $\ell = 1$.

Similar to the two-prototype WTA system, for all $\hat{\alpha}$, the trivial global minimum of $f$ is the configuration with only one prototype converging at the region of high density and two others diverging (Fig. 5.5(1)). At their respective characteristic $\hat{\alpha}$, new minima appear, in order, with two converging prototypes (Fig 5.6(2)) and three converging prototypes (Fig 5.6(3)). The relevant metastable configurations so far remain unspecialized.

At a critical value $\hat{\alpha}_s$, a specialized configuration with lower $\langle e_{VQ} \rangle_\xi$ appears. However, the phase transition is discontinuous or first order, i.e. a sudden jump occurs from an unspecialized to a specialized state, see Fig. 5.7. This behavior was also observed in multilayer neural networks with three or more hidden units (Biehl et al. 1998). The projections of two of the three prototypes into the span($B_1, B_2$) coincide close to the center of the cluster with larger prior weight. Thus the behavior in this example is as follows, illustrated in Figs. 5.6: (1) $\to$ (2) $\to$ (3) $\to$ (5).

Note that, in a generic discontinuous phase transition, one expects a range of values $\hat{\alpha}_s \leq \hat{\alpha} < \hat{\alpha}_c$ where the specialized configuration corresponds to a local minimum of $f$, see (Biehl et al. 1998) for an example in supervised learning. However, for the setting of parameters considered here, $\hat{\alpha}_s = \hat{\alpha}_c$ within the achievable numerical precision and we find that the free energy of the specialized configuration is always smaller than that of the system with $\Delta S = 0$. However, a local minimum of
5.5. Three-prototype systems

Figure 5.6: Possible configurations in the WTA scenario with three prototypes. The darker circle symbolizes the Gaussian cluster with larger prior weight. Graphs (1-3) display unspecialized configurations, with two prototypes diverging (1), one prototype diverging (2) and no prototypes diverging (3). Specialized configurations are shown in graphs (4-6), with one prototype diverging (4), low quantization error (5) and suboptimal performance (6). Note that unspecialized prototypes or prototypes representing the same cluster would indeed coincide in the projection but separate in the $(N - 2)$-dim. orthogonal space.

$f$ corresponding to unspecialized $w$, persists in the range $\hat{\alpha}_c \leq \hat{\alpha} < \hat{\alpha}_d$. While such a metastable state does not represent thermal equilibrium, its existence can have strong delaying effects in the practical optimization of $H$.

5.5.2 Large cluster separation

A different behavior can be observed in the $K = 3$ system in a scenario with relatively large separation, e.g. $\ell = 4$. Again at small $\hat{\alpha}$, metastable configurations (1) and (2) in Fig. 5.6 appear as $\hat{\alpha}$ grows.

However, the specialization transition occurs at a much earlier stage, as the clusters are more easily identified. A specialized two-prototype configuration (Fig. 5.6(4)) then becomes metastable, even though no non-trivial three-prototype configuration exists. The latter emerges directly in a specialized state (Fig. 5.6(5)) at a lower quantization error. Here the three-prototype unspecialized configuration state is absent: the sequence is $(1) \rightarrow (2) \rightarrow (4) \rightarrow (5)$ as illustrated in Figs. 5.6.

An important feature in this scenario is the presence of local minima of $H(W)$. 
5. Phase transitions in off-line vector quantization and neural gas

Figure 5.7: (a) The order parameters \( R_{s1} \) of the stable configuration for \( K = 3 \), with two of the three values coinciding in the upper curve. The transition is discontinuous; solid (dashed) lines mark global (local) minima of \( f \). Here, \( \hat{\alpha}_{s}(K = 3) \approx 4.37 \) and \( \hat{\alpha}_{d}(K = 3) \approx 4.40 \). (b) The mean error \( \langle e \rangle_{\xi} \). The transition results in a discontinuous drop for \( K = 3 \) at the respective \( \hat{\alpha}_{c} \). The parameters of the input density (5.1) are \( p_1 = 0.8, p_2 = 0.2 \) and \( \ell = 1 \) in both panels.

The lowest non-trivial minimum of \( f \), i.e. the optimal configuration of prototypes, has two prototypes representing the cluster with larger prior, see Fig. 5.6(5) for an illustration. A suboptimal local minimum of \( f \) corresponds to the inverted configuration with two prototypes representing the weaker cluster, see Fig. 5.6(6). The sequence (1) \( \rightarrow \) (2) \( \rightarrow \) (4) \( \rightarrow \) (6) in Figs. 5.6 can also be observed.

In this setting, the minima with configurations of type (5) and (6) are similar to global and local minima of the energy \( H(W) \), respectively. Due to its lower value of \( f \), the optimal state is also the more stable state at all \( \hat{\alpha} \). However, in a dynamical context, the system may still be trapped near the local minima of \( H(W) \) for long learning times.

5.5.3 Neural Gas

Figure 5.8(b) relates to an NG system with \( \lambda = 0.25 \) and \( \lambda = 0.5 \) using the previously described example with larger separated clusters, \( \ell = 4 \). The characteristics of the system with \( \lambda = 0.25 \) are similar to WTA: non-trivial optimal and suboptimal configurations appear at certain values of \( \hat{\alpha} \), depending on \( \lambda \). Note that while NG with large \( \lambda \) forces the prototypes to be in a non-trivial state, it may also has its drawback in terms of \( \langle e \rangle_{\xi} \), i.e. it is higher than with WTA.

For sufficiently large \( \lambda \), the energy landscape of \( H(W) \) changes drastically. In contrast to WTA and NG with small \( \lambda \), the suboptimal local minimum of \( f \) disap-
5.6 Summary

We have investigated the equilibrium properties of WTA and rank-based VQ systems along the lines of statistical physics of off-line learning. The analysis of the learning behavior is based on two approaches: the high temperature limit and the annealed approximation. The simplifying high temperature limit provides exact analyses, yet already demonstrates the qualitative behavior of the annealed approximation at finite temperatures.

Both methods allow the study of the landscape of the cost function, which provides important insights into the training process. While the analysis concerns equilibrium properties of a hypothetical training process, its results are indeed relevant for practical situations. The existence of local metastable states, for instance, can influence their outcome drastically.

For, both, two- and three-prototype systems, a critical number of examples is required before the underlying structure can be detected at all. While it is obvious...
that any practical algorithm should give better performance with larger data sets, our results imply a more drastic effect: even the best optimization strategies will fail below this critical number of data. This parallels findings for various other training scenarios and is highly relevant from a practical point of view: even the best optimization strategies will fail completely if too few example data are available. The nature of the phase transition is continuous for two prototypes and discontinuous for $K \geq 3$. The metastable states for $K \geq 3$ show that long delays may happen in practice even if the critical number of examples is exceeded.

In WTA-based systems, divergent behavior may be observed if the training set is too small or the prototypes are initialized far from the region of high density. Furthermore, the system can be trapped in suboptimal local minima of the cost function. The NG-based system is more robust, i.e. for practical algorithms one can expect NG to be less sensitive to initial conditions.

Note that the phase transitions discussed here are found in the high temperature limit and annealed approximation, which makes qualitatively correct predictions for high temperatures. In the exact analysis of training at low temperatures, the nature of the phase transitions may be different. This could be treated in further extensions using the replica approach.

The formalism explained in this paper can also be applied to supervised learning schemes based on cost functions, which will be discussed in the following chapter.
Chapter 6

Equilibrium properties of off-line LVQ

Abstract

The statistical physics analysis of off-line learning is applied to supervised learning vector quantization (LVQ) schemes. Typical learning behavior is obtained from a model with data drawn from high dimensional Gaussian mixtures and a system of two or three competing prototypes. The analytic approach becomes exact in the limit of high training temperature. We study three cost function related LVQ algorithms, including a limiting case of generalized LVQ (GLVQ) termed LVQ 2.1, learning from mistakes (LFM) and Robust Soft LVQ (RSLVQ), with the additional influence of a weight decay. Performance is measured from the generalization error at different sizes of the training set. In our findings, LFM achieves poor classification ability and RSLVQ is robust with respect to learning parameter. LVQ 2.1 outperforms both algorithms, given a properly chosen weight decay. For three-prototype systems, we find continuous phase transitions between different prototype configurations at critical numbers of examples.

6.1 Introduction

Various modifications have been developed to the standard Learning Vector Quantization (LVQ) methods (Kohonen 1990) to achieve, for instance, faster convergence and better approximation of Bayesian decision boundaries. In its original form, LVQ1 is based on heuristics. Several extensions aim at substituting the heuristics with adaptation rules derived from a well-chosen cost function. Thereby, the generalization behavior is governed by the choice of the cost function, which is expected to yield low error rates.

The construction of an appropriate cost function, however, is particularly challenging because it cannot be directly derived from the generalization error. Cost functions can be inspired by the concept of large margins as in, e.g., Generalized LVQ (GLVQ) (Sato and Yamada 1995). Other algorithms relate to explicit assumptions about the statistics of data. For instance, RSLVQ maximizes the likelihood ratio of a Gaussian mixture model ansatz (Seo and Obermayer 2003). While these cost functions may appear plausible, their relation with the generalization ability remains unclear in practical situations.
Furthermore the landscape of the cost function plays an important role in the success of training. Systems may be trapped in local minima of the cost function, resulting in suboptimal results. Several algorithms are specifically developed to escape from the local minima such as Supervised Neural Gas (Hammer et al. 2006). Nevertheless, these problems remain and we need to investigate the cost function.

Methods from statistical physics allow for investigation of equilibrium properties of large systems, such as neural networks (Seung et al. 1992, Watkin et al. 1993). In the by now standard analysis of off-line (batch) learning, training is interpreted as stochastic minimization of a cost function. Here, our analysis is based on the limit of high training temperature, a simplification that has provided useful insights into many learning scenarios (Biehl et al. 1998, Seung et al. 1992, Watkin et al. 1993, Witoelar and Biehl 2009).

In this chapter, we analyze several important LVQ schemes: a limiting case of GLVQ, Learning From Mistakes (LFM) and Robust Soft LVQ (RSLVQ) in systems of two and three competing prototypes. In addition we study weight decay as a control parameter against instabilities. Both learning schemes have been studied in terms of training dynamics in (Biehl et al. 2007). Here we explicitly treat off-line learning from a given fixed data set. Our analysis shows how successful learning depends on the size of the training set. Also, the learning process exhibits phase transitions which are reminiscent of those observed in unsupervised VQ (Witoelar and Biehl 2009) or multilayer neural networks (Ahr, Biehl and Urbanczik 1999, Biehl et al. 1998, Seung et al. 1992).

### 6.2 Cost functions

Consider a data set of \( P \) examples given as \( \mathcal{D} = \{(\xi^\mu, y^\mu) \in \mathbb{R}^N \times \{+1, -1\}\}^P_{\mu=1} \). We exploit the thermodynamic limit \( N \rightarrow \infty \) and assume that the number of examples also grows linearly in \( N \), i.e. \( P \propto N \). Examples are generated independently according to a model density given in Eq. (2.5). We restrict the system to \( M \) Gaussians:

\[
P(\xi) = \sum_{\sigma=1}^{M} p_\sigma P(\xi|\sigma) \quad \text{with} \quad P(\xi|\sigma) = \frac{1}{(2\pi)^{N/2}} \exp \left[ -\frac{1}{2\nu_\sigma} (\xi - f_{BS})^2 \right] \quad (6.1)
\]

where each cluster generates data with only one class. For a major section of this chapter, we consider \( M = 2 \) systems and denote \( y^\mu = \sigma^\mu = \pm 1 \). We consider a system of \( K \) prototype vectors \( \mathbf{W} = \{(\mathbf{w}_S, c_S)\}^K_{S=1} \). Cost functions considered here
are expressed as empirical averages

\[ H(W) = \frac{1}{P} \sum_{\mu=1}^{P} e(W, \xi^{\mu}) / P. \]  

(6.2)

We study the following specific examples:

1. **Unrestricted LVQ 2.1**

   \[ e_{\text{LVQ 2.1}}(W, \xi^{\mu}) = d_{j}^{\mu} - d_{K}^{\mu}. \]  

   (6.3)

   We restrict the analysis to the Euclidean distance \( d_{S}^{\mu} = (\xi^{\mu} - w_{S})^2 \). The prototype \( w_{J} \) is the closest correctly labeled prototype, while \( w_{K} \) is the closest incorrectly labeled prototype. Generally, LVQ 2.1 prescription selects data from a window about the decision boundary in the form of (4.3). Here we study the limit of infinite window size. GLVQ (Sato and Yamada 1995) with \( e(W, \xi^{\mu}) = \Phi((d_{S}^{\mu} - d_{T}^{\mu})/(d_{S}^{\mu} + d_{T}^{\mu})) \) reduces to the above for \( N \to \infty \) and \( \Phi(x) = x \). Note that for \( N \to \infty \) the term \((\xi^{\mu})^2 = \mathcal{O}(N)\) dominates \( d_{S}^{\mu} - d_{T}^{\mu} \) yielding a constant prefactor, while it cancels out in the numerator \( d_{S}^{\mu} - d_{T}^{\mu} \).

2. **LFM**

   \[ e_{\text{LFM}}(W, \xi^{\mu}) = (d_{S}^{\mu} - d_{T}^{\mu}) \Theta(d_{j}^{\mu} - d_{K}^{\mu}). \]  

   (6.4)

   The prototypes \( w_{J} \) and \( w_{K} \) are defined as in LVQ 2.1 and \( \Theta(x) \) is the Heaviside function. Here only misclassified data contribute to the cost reminiscent of perceptron training (Engel and van den Broeck 2001, Watkins et al. 1993). We refer to this as learning from mistakes (LFM). Note that this cost function is indirectly derived from the crisp version of Robust Soft LVQ (Seo and Obermayer 2003).

3. **RSLVQ**

   Robust Soft LVQ (RSLVQ) is developed based on statistical modeling of the input data (Seo and Obermayer 2003). The algorithm minimizes a cost function based on a mixture model, described as

   \[ e_{\text{RSLVQ}}(W, \xi^{\mu}) = -\ln \frac{p(\xi^{\mu}, \sigma^{\mu}|W)}{p(\xi^{\mu}|W)} \text{ with } \begin{cases} 
   p(\xi^{\mu}, \sigma^{\mu}|W) = \sum_{S: x_{S} = y^{\mu}}^{K} P_{S\mu}(\xi^{\mu}|S), \\
   p(\xi^{\mu}|W) = \sum_{S=1}^{K} P_{S\mu}(\xi^{\mu}|S), 
\end{cases} \]  

   (6.5)

   where \( p(\xi^{\mu}|W) \) approximates the actual probability density \( P(\xi) \), c.f. (6.1). It is assumed that every component \( j \) of the mixture generates examples which
belong to one class, viz. $c_S$. $N_\sigma$ is the number of classes and $P_S$ is the probability that the examples are generated by a particular component $S$ and $p(\xi^\mu|S)$ is the conditional probability that $S$ generates a particular example $\xi^\mu$.

Using a Gaussian mixture model ansatz, $p(\xi^\mu|S) = (2\pi v_S)^{(N/2)} \exp(-\frac{1}{2}d^\mu_S/v_S)$ and choosing equal probabilities $P(S) = 1/K$ and equal variances $v_S = v_{soft}$, we obtain the following cost function:

$$e_{RSLVQ}(W, \xi^\mu) = -\ln \left( \frac{\sum_{S,\xi^\mu} \exp \left( \frac{1}{2}d^\mu_S/v_{soft} \right)}{\sum_{S=1}^{K} \exp \left( -\frac{1}{2}d^\mu_S/v_{soft} \right)} \right)$$

(6.6)

where $v_{soft}$ is termed the softness hyperparameter.

Note that stochastic gradient on the above LVQ cost functions yields the online adaptation rules treated in Chapter 4. Several algorithms, in particular LVQ 2.1, are known to be unstable and thus usually require an additional restriction such as a window around the decision boundary for selection of examples. These schemes are studied in Chapter 4. Without the additional constraints, one or more prototype vectors may diverge to very large lengths.

In this work, we prevent instability by means of a regularization method which punishes configurations with large lengths using an additional energy term called weight decay (Ahr, Biehl and Schloesser 1999). This approach is widely used in feedforward neural networks. This regularization penalizes the length of each prototype $w_S$ by the measure $\gamma (w_S - O_\gamma)^2$ where $\gamma$ is a weight decay parameter and $O_\gamma$ is a chosen origin of the weight decay. We place $O_\gamma$ at the center of mass of the input density, $\ell(p, B_+ + p, B_-)$, and preserve the symmetry axis $\ell(B_+ - B_-)$. In practice, this is equivalent to transforming the data into zero mean and calculating the weight decay wrt. the transformed origin. Hence we obtain the modified cost function

$$H(W) = \sum_{\mu=1}^{P} e(W, \xi^\mu)/P + \gamma \sum_{S=1}^{K} [w_S - \ell(p, B_+ + p, B_-)]^2.$$

(6.7)

6.3 Analysis

We apply the statistical mechanics of off-line learning framework. In this chapter, we only briefly go over the steps and refer to Section 5.3 for the detailed explanation of analysis. In particular, we analyse using the high temperature limit as described
6.4 Results

in Section 5.3.1. Stochastic minimization on $H(W)$ under training temperature $T$ yields the equilibrium density

$$P(W) = \frac{\exp [-\beta H(W)]}{Z} \text{ where } Z = \int d\mu(W) \exp [-\beta H(W)].$$

where $\beta = 1/T$, the normalization $Z$ is called the partition sum, and the measure $d\mu(W)$ is the $NK$-dim. volume element. Thermal averages $\langle \rangle$ over $P(W)$ can be evaluated as derivatives of the so-called free energy $-\ln Z/\beta$. To observe the typical learning behavior, we require the computation of the quenched free energy (Engel and van den Broeck 2001, Seung et al. 1992, Watkin et al. 1993), i.e. an average over all possible data set $D$,

$$F = -\langle \ln Z \rangle_D / \beta$$

In general, the computation of $\langle \ln Z \rangle_D$ requires tools from theory of disordered systems (Seung et al. 1992); however, it is greatly simplified in the limit of high temperatures, $\beta \rightarrow 0$. Non-trivial results can only be expected if the increased noise is compensated for by a large number of examples $P = \hat{\alpha}N/\beta$.

We can fully express averages in terms of order parameters $R_{S\sigma} = w_S \cdot B_\sigma$ and $Q_{ST} = w_S \cdot w_T$. (6.8)

where $R_{S\sigma}$ are the projections of prototype vectors $w_S$ on the center vectors $B_\sigma$ and $Q_{ST}$ correspond to the self- and cross- overlaps of the prototype vectors. Using the modified cost function in Eq. (6.7), we define the free energy function

$$f(\{R_{S\sigma},Q_{ST}\}) = \hat{\alpha} \langle e(\{R_{S\sigma},Q_{ST}\})\rangle_\xi + \hat{\alpha} \gamma \sum_{S=1}^{K} [Q_{SS} - 2\ell(p_+R_{S+} + p_-R_{S-}) + \ell^2]$$

$$-s(\{R_{S\sigma},Q_{ST}\}).$$ (6.9)

The quenched free energy becomes $-\langle \ln Z \rangle_D / \beta = \min f(\{R_{S\sigma},Q_{ST}\})$. Given a fixed training set size $\hat{\alpha}$, we minimize Eq. (6.9) to obtain configurations of order parameters $\{R_{S\sigma},Q_{ST}\}$ which dominate the equilibrium density and describe the typical learning curve.

6.4 Results

We discuss the results for two- and three-protoype LVQ 2.1, LFM and RSLVQ systems and compare to the minimum limiting $\epsilon_g$ for linear decision boundaries.
Figure 6.1: LVQ 2.1 with the system \( p_+ = 0.8 \) and \( \ell = 1 \). Left panel: \( R_{S_+} \) vs. \( R_{S_-} \) plot with \( \hat{\alpha} = 10 \) and weight decay \( \gamma = 1, 0.8 \) and 0.65, displays the projection of prototypes on the \((B_+, B_-)\) plane. Prototype \( w_- \) diverges without a weight decay. Right panel: \( \epsilon_g \) vs weight decay \( \gamma \) for LVQ 2.1 with \( K = 2 \) for \( \hat{\alpha} = 1, 4 \) and 10 (solid lines, top to bottom). Learning systems with \( \gamma < |p_+ - p_-| \) produce divergent behaviour. The performance approaches the best linear decision boundary (dotted lines) as \( \hat{\alpha} \to \infty \) with proper settings of \( \gamma \).

### 6.4.1 Two-prototype systems

Without regularization, the unrestricted LVQ 2.1 system generally exhibits divergent behavior. Given unequal priors, only the trivial minimum of the free energy function \( f = -\infty \) exists, because the cost function \( (d_J - d_K) \) is minimized with infinite length of the prototype representing the weaker class, i.e. \( Q_{KK} \to \infty \). This results in trivial classification \( \epsilon_g = \min\{p_+, p_-\} \). A large weight decay \( \gamma \), depending on \( |p_+ - p_-| \), is required for LVQ 2.1 to obtain stable configurations \( \{R_{S_+}(\hat{\alpha}), Q_{ST}(\hat{\alpha})\} \). Figure 6.1 displays the prototype projections on the plane spanned by \((B_+, B_-)\). The performance is improved with proper settings of \( \gamma \). At settings with small \( \gamma \), the prototypes have very large lengths, which is not the desired result of training. Conversely, large \( \gamma \) places greater importance of the weight decay at the expense of higher generalization error. A pronounced optimal \( \gamma \) exists, which is fairly robust wrt. the size of the training set, displayed in Fig. 6.1 (right). For all settings of \( \gamma \), generalization improves as the number of training examples increase.

Given a proper weight decay, \( \epsilon_g \) reaches the optimal generalization error at \( \hat{\alpha} \to \infty \).

Contrary to LVQ 2.1, LFM is based on a limiting case of a bound cost function and therefore expected to be stable. Still, note that the form of the LFM cost function in (6.4) only calculates the difference of distances \( (d_J - d_K) \). This is in contrast to unsupervised VQ, which explicitly includes distances as a quantization measure and attracts prototypes towards regions of high input density, see (5.2). The supervised learning cost functions considered here emphasize on classification without regard-
6.4. Results

Figure 6.2: Order parameters \( R_{S^+}, S = 1, 2 \) for LFM and RSLVQ with \( p_+ = 0.7, p_- = 0.3, v_+ = 1.5, v_- = 1 \) and small weight decay. Left: The order parameters \( R_{S^+} \) vs. \( \alpha \). Right: The corresponding generalization error \( \epsilon_g \) vs. \( \alpha \) converges for all \( v_{\text{soft}} \) in RSLVQ, while LFM performs poorly compared to the best linear classification with \( \epsilon_{g,\text{opt}} \).

Regarding data representation. The cost function \( e(\{R_{S^\sigma}, Q_{ST}\}) \) is constant wrt. translations of all prototypes along the decision boundary, producing degenerate global minima. This is in accordance to the findings from on-line LFM in (Biehl et al. 2007), where the asymptotic configurations depend on the initialization, while the projections of the prototype assume a position along a line parallel to the symmetry axis \( \ell(B_+, B_-) \). The difference in off-line learning is that these configurations produce varying free energy function \( f(\{R_{S^\sigma}, Q_{ST}\}) \): the entropy term \( s \) can be increased arbitrarily by moving all prototypes away from the cluster centers without affecting classification. Thus for the analysis, a small weight decay \( \gamma \) is still required for the analysis to prevent diverging prototypes.

Assuming a small \( \gamma \), figure [6.2](left) displays the equilibrium \( \{R_{S^+(\hat{\alpha})}\}_{S=1}^{K} \), where the projections of the two prototypes converge at large \( \hat{\alpha} \). The actual distance between prototypes \( w_1 \) and \( w_2 \) is calculated by

\[
\Delta_q = \sqrt{(w_1 - w_2)^T} = \sqrt{Q_{11} - 2Q_{12} + Q_{22}},
\]

which decreases as \( \hat{\alpha} \) grows for off-line LFM. At \( \hat{\alpha} \to \infty \), \( \Delta_q = 0 \) and both prototypes coincide. This is identical to the convergence analysis of on-line LFM (Biehl et al. 2007) for small learning rates. Although its cost function appears reasonable, LFM exhibits surprisingly poor generalization ability compared to the best linear decision boundary, see \( \epsilon_g \) vs. \( \hat{\alpha} \) in Fig. 6.2(right). This is due to large orthogonal components to the \((B_+, B_-)\) plane. The LFM is outperformed by LVQ 2.1 and for highly unbalanced priors, \( \epsilon_g \) even exceeds that of the trivial classification. These LFM results apply only to the crisp version of RSLVQ. For the soft version, the per-
performance is improved significantly, as explained in the following.

Next we investigate the learning behavior of RSL VQ. Again, the cost function can be interpreted as the difference of distances, for instance by rewriting (6.6) for two-prototype RSL VQ as

\[
e_{\text{RSL VQ}}(W, \xi^\mu) = \ln \left[ 1 + \exp \left( -\frac{d_{J\mu}^2 - d_{KJ}^2}{2v_{\text{soft}}} \right) \right].
\]

(6.11)

where \(d_{J\mu}\) and \(d_{K\mu}\) are the correct and incorrect winners given example with class \(\sigma^\mu\). The quantity of Eq. (6.11) can be calculated numerically as a 1-dim. integration in Appendix B.2. Similar to LFM, the minima of \(H(W)\) for RSL VQ are degenerate and a small weight decay is required for the analysis. Figure 6.3 displays the cost functions for different settings of \(v_{\text{soft}}\), which converge at large \(\hat{\alpha}\), except in the LFM limit \(v_{\text{soft}} \rightarrow 0\). This is also observed in on-line RSL VQ, where the asymptotic value of the cost function is independent of \(v_{\text{soft}}\) at small learning rates.

In Figure 6.3 (right), we see that the generalization performance is independent of \(v_{\text{soft}}\). It is important to point out that while the stationary errors of RSL VQ is indeed robust, the learning dynamics and convergence speed remains highly dependent on learning parameter settings. As we have shown in Section 4.5.4, the learning curve \(\epsilon_g\) vs. learning time for on-line RSL VQ may become non-monotonic at large \(v_{\text{soft}}\) and learning plateaus may be encountered which drastically increase convergence times.

As opposed to LFM, where both prototypes collide at large \(\hat{\alpha}\), the prototypes remain separated for RSL VQ, as shown in Fig. 6.2 (left). This allows analysis at the limit \(\hat{\alpha} \rightarrow \infty\). We observe that the distance between prototypes defined in (6.10)
scales proportionally to \( v_{\text{soft}} \). In the model with unequal variance here with \( v_+ > v_- \), the generalization error of RSLVQ remains suboptimal even at \( \hat{\alpha} \to \infty \). At highly unbalanced variances, performance deteriorates for RSLVQ with global \( v_{\text{soft}} \).

Although various performances of one specific cost function at different \( \hat{\alpha} \) can be analyzed, the comparison between two different cost functions is generally not valid in the high temperature limit. Any multipliers of \( H(W) \) are rescaled into \( \beta \to 0 \) and, consequently, the scale of \( \hat{\alpha} \) is not consistently defined between different cost functions. The only exception which allows direct comparison is at the limit \( \hat{\alpha} \to \infty \). In this limit, LVQ 2.1 outperforms LFM and RSLVQ, given a properly chosen weight decay.

The behavior described by all the learning schemes above differs from that of unsupervised vector quantization in Chapter 5. In the latter, permutations between prototypes lead to effects of \textit{retarded learning}, i.e. a minimum number of examples is required to have any chance of successful learning, see e.g. (Biehl et al. 1998, Lootens and van den Broeck 1995). In the supervised learning scenario, the permutation symmetry between prototypes is broken by the class assignment of each prototype. Therefore, as long as \( \hat{\alpha} \) is in the order \( O(1) \), each prototype immediately aligns itself towards its respective class mean.

### 6.4.2 Three-prototype systems

The learning behavior is qualitatively different for \( K = 3 \) systems and data with two classes. Here two out of three prototypes share the same class, for instance \( c_S = \{+, +, -\} \).

Different configurations between the two prototypes produce distinct stable states, illustrated in Fig. 6.4. In Fig. 6.4(a), the prototypes \( \mathbf{w}_1 \) and \( \mathbf{w}_2 \) have unequal lengths and lie asymmetrically along \( \ell(B_+ - B_-) \). We represent this state with \( R_{1\sigma} \neq R_{2\sigma}, \forall \sigma \) and \( Q_{11} \neq Q_{22} \). The decision boundary is predominantly defined by only two prototypes. This configuration does not provide \( K = 3 \) systems with an advantage over \( K = 2 \) systems in terms of classification. In Fig. 6.4(b), these prototypes have equal lengths, and lie symmetrically around \( \ell(B_+ - B_-) \). The three prototypes form a wedge-shaped decision boundary which allows for better generalization ability, in general. For symmetrical reasons, we can rotate the system around the symmetry axis and represent this state with a configuration with \( R_{1\sigma} = R_{2\sigma}, \forall \sigma \) and \( Q_{11} = Q_{22} \neq Q_{12} \).

Due to the large repulsion from the stronger class, LVQ 2.1 requires large values of \( \gamma \) to prevent divergence. With this decay, the asymmetric configuration is unfavorable in terms of free energy and therefore only the symmetric state is found at any \( \hat{\alpha} \). Contrarily, for LFM systems, the asymmetric configuration is found at small
Figure 6.4: Possible configurations of a $K = 3$ system with two similarly labeled prototypes $c_S = \{+, +, -\}$, providing distinct decision boundaries. Graphs (a-b) displays systems with two clusters. In (a), all available prototypes are utilized for the decision boundary, while in (b), one of the prototypes is redundant. Graphs (c-d) displays systems with three clusters, with unspecialized (c) and specialized (d) configurations. Two prototypes completely overlap in (c).

$\gamma$ and small $\hat{\alpha}$. The prototype with the larger length in this configuration has a potential to diverge: the entropy scales with $Q_{SS}$ while its contributions to $e$ vanish, leading to $Q_{SS} \rightarrow \infty$. Thus, a sufficiently large weight decay is necessary even for three-prototype LFM at small $\hat{\alpha}$. A continuous phase transition occurs at a critical number of examples, where the system switches from the asymmetric to the symmetric configuration, see Fig. 6.5 (left). The critical number of examples depends on $\gamma$: while generally larger weight decay degrade the performance of LFM, it also has a lower critical value and allows the use of more prototypes at small training sets.

In systems with multiple clusters, permutations between prototypes with the same class produce the specialization phase transition. Here we investigate three prototype LFM learning on three clusters of data $\sigma = \{1, 2, 3\}$ with classes $y_\sigma = \{+, +, -\}$, see Fig. 6.4 (c-d). Given sufficient training examples, we obtain a classifier system using three prototypes. While cluster $\sigma = 3$ is already identified by the prototypes, the distinction between the remaining clusters is not detected: $R_{S1} = R_{S2}, \forall S$. This configuration is illustrated in Fig. 6.4(c). Permutation symmetry between same-class prototypes, which exchange gives identical states, is responsible for the effects of retarded learning.
6.5 Discussion

We have investigated LVQ systems along the lines of statistical physics of off-line learning using the limit of high training temperatures for three important cost functions: LVQ 2.1, LFM and RSLVQ. Typical behavior of the prescribed schemes is obtained for fixed training set sizes.

While LFM appears plausible, its performance is unexpectedly poor compared to the optimum achievable error for this learning problem, even with very large training sets. As expected, LVQ 2.1 displays divergent behavior and the performance is highly sensitive to the setting of weight decay. RSLVQ is shown to be robust with respect to the softness parameters, but does not produce optimal performance for data with highly unbalanced variances. Optimizing the performance wrt. weight decay,
LVQ 2.1 exhibits better generalization ability than LFM and RSLVQ for both two- and three prototype systems and obtains close to optimal errors, given sufficiently large training sets.

In three-prototype systems, we find continuous phase transitions between prototype configurations. While larger data sets should improve the performance of practical algorithms, a critical size of the training set is required to effectively utilize all available prototypes. Another critical number of examples exists for the system to find substructures to improve classification, producing more continuous phase transitions. Treatment of systems with more prototypes and higher degrees of symmetry can lead to the existence of other phase transitions. The nature of the transition has been shown to be qualitatively different in multilayered neural networks and soft-committee machines in (Biehl et al. 1998, Opper 1994, Schottky 1995), displaying first order phase transitions for many hidden units. Analogous to these findings, we also expect discontinuous transitions for LVQ systems with three or more prototypes possessing the same class label.

The analysis at finite training temperatures is important for practical applications, e.g. annealing schemes which end with low temperatures. This analysis requires additional techniques such as the annealed approximation or replica method with independent variation of the number of examples \( P/N \) and temperature. This allows for comparisons between algorithms at certain training set sizes and provides, amongst others, the relation between training error and generalization error.
Chapter 7

Summary and Outlook

7.1 Summary

In this thesis we present a theoretical framework to investigate prototype-based training prescriptions. The analysis is performed using concepts from statistical physics which allow for an exact mathematical description of the system in terms of characteristic quantities, so-called order parameters. The set of order parameters fully describes the training dynamics or equilibrium learning states. The model scenario we utilize to compare LVQ algorithms is described in Chapter 2.

In chapters 3 and 4, we examine on-line training and observe the learning dynamics of (unsupervised) VQ and (supervised) LVQ. The learning behavior is monitored in terms of the evolution of the order parameters described by ordinary differential equations. In the unsupervised data clustering analysis, we demonstrate the advantages of Neural Gas (NG) over Winner-Takes-All algorithms. First, NG can improve convergence speed in comparison to Winner-Takes-All for poor initialization of prototypes. Furthermore, NG achieves robustness with respect to initial conditions. However, the Neural Gas may still converge into local minima and does not always obtain the best possible quantization error.

Among the supervised LVQ schemes, we compare in detail the advantages and drawbacks of various window-based example selection schemes including LVQ 2.1, Learning From Mistakes (LFM), Generalized LVQ (GLVQ) and Robust Soft LVQ (RSLVQ). The sensitivity of the learning curves with respect to parameters is studied for all algorithms. Surprisingly, LVQ 2.1 produces the optimal linear decision boundary yielding optimal generalization ability. We also find learning plateaus in the learning stage in multi-prototype systems.

In chapters 5 and 6 we apply statistical physics of off-learning to analyse LVQ algorithms. Training is interpreted as a stochastic minimization of the cost function on the data set $D$, where the formal temperature $T$ controls the degree of randomness. We investigate the equilibrium properties of WTA and rank-based VQ systems using the high temperature limit and the so-called annealed approximation. We find that Neural Gas is more robust than WTA with respect to initialization, which
agrees well with the results obtained from on-line learning analysis. We find phase transitions in the learning phase: a critical number of examples must be presented to the system before it can identify any underlying structure within the data. The nature of transition is continuous in two-prototype systems and discontinuous in three-prototype systems. This is highly relevant from a practical point of view: any optimization strategy will ultimately fail completely if too few example data are available and metastable states for three-prototype systems may cause long delays in learning.

Finally, we analyse the cost function of LFM, LVQ 2.1 and RSLVQ in the high temperature limit. For this learning problem, the performance of LFM is unexpectedly poor compared to the optimum achievable error. Given properly chosen weight decay, LVQ 2.1 exhibits better generalization ability than LFM and RSLVQ for both two- and three-prototype systems. In three-prototype systems, we find continuous phase transitions between prototype configurations. We observe that critical sizes of the training set are required to effectively utilize all available prototypes and find substructures within the data. Treatment of systems with more prototypes may demonstrate additional phase transitions.

7.2 Relation between on-line and off-line analyses

While the general approaches of on-line and off-line analyses are technically different, the emphasis of both techniques highly complement each other. On-line analysis allows for investigation of convergence speed depending on learning parameters and initial conditions, which are essentially absent in off-line analysis. Furthermore it allows the study of heuristic practical LVQ prescriptions which are not based on cost functions. Additionally, unbounded cost functions such as LVQ 2.1 can lead to highly divergent behaviors which cannot be treated using the off-line analysis. From on-line analysis, we can approximate their non-trivial asymptotic learning behaviors. Alternatively, off-line analysis rigorously evaluates the characteristics of the cost function landscape in order to find all possible equilibrium states, without explicitly considering convergence times. The stable and metastable states uncover the presence of fixed points that may be encountered in on-line analysis.

Qualitatively, the resemblance between the findings of both methods is already apparent. Our on-line analysis of unsupervised learning in Chapter 3 reveals that learning slows down considerably for WTA with poor initialization and demonstrates the advantage of rank-based NG algorithms. In off-line learning, this corresponds to suboptimal equilibrium states discussed in Section 5.4.1 where trivial minima exist at large prototype lengths. We also demonstrate in off-line NG how
the rank parameter $\lambda$ changes the energy landscape, which will affect typical escape times from the suboptimal configuration. From both analyses we demonstrate that NG is more robust than WTA. In on-line learning, the asymptotic configurations of NG are independent of initial conditions. This is mirrored in off-line learning by the smooth energy landscape where the metastable states disappear, given sufficiently large $\lambda$.

In general, the results of both analyses become identical at long learning times $\hat{\alpha} \to \infty$ for on-line and large training sets $\hat{\alpha} \to \infty$ for off-line training. We compare supervised problems in Chapter 4 and Chapter 6 and confirm identical results for RSLVQ algorithms. The dependence of asymptotic configurations on the initial conditions is explained by the degeneracy of the cost function minima. The comparison of LFM is more restricted, because the prototypes coincide in the aforementioned limits. We observe that its behavior at large $\tilde{\alpha}$ approximates that of large $\hat{\alpha}$.

In order to fully compare both approaches, several imposed limits have to be resolved. For instance, we assume a small learning rate in the on-line analysis. However, this would correspond to low temperature off-line analysis which requires a full treatment of the replica method. In this work, this can only be approached with the annealed approximation. Conversely, off-line analysis allows investigation of learning from restricted data sets and we can observe distinct training and generalization errors at finite temperatures, see the annealed approximation results in Section 5.4.3. In on-line analysis, this corresponds to recycling examples from a finite data set which imposes correlations between training examples and the system. The subject of on-line learning for finite training sets in neural networks are investigated in, for example, (Barber and Sollich 1998, Rae et al. 1999).

The presence of symmetries produces learning behaviors which carry over to both on-line and off-line analysis. This appears during the specialization phase of prototypes in unsupervised learning. Permutation symmetry between two or more prototypes, i.e. equivalent configurations from exchange of prototypes, greatly inhibits the learning process in the thermodynamic limit. In on-line learning, prototypes require long learning times to escape from symmetry-induced fixed points, which result in plateaus in the learning curves. In off-line learning, symmetries create competition of states with varying entropy, where the unspecialized state is favored at small training sets. This in turn produces retarded learning, where specialization only occurs for training sets larger than a critical size. In supervised learning problems, permutation symmetries are broken by different class assignments to prototypes, but they remain among prototypes with the same class. Hence, learning plateaus and retarded learning are observed only in multi-prototype LVQ systems.
7.3 Outlook

While our analysis focuses on an idealized scenario for this training scheme, the results uncover underlying non-trivial effects which would carry on to any learning problems. Nevertheless, other effects may be discovered in more realistic models, which may actually become more relevant in practical situations. In this outlook, we explore several possibilities to analyse more complex models and provide more general learning characteristics.

We have shown in this thesis that it is possible to extend the model to accommodate arbitrarily many isotropic clusters, provided that the number of clusters is small compared to the dimensionality. However, we assume that the features of the data, i.e. the dimensions, are completely uncorrelated, which would be unreasonable in practical situations. An important step would include learning systems with anisotropic clusters. Along these lines, we can also interpret data with highly correlated components as non-isotropic clusters with uncorrelated components, e.g. after processing by principal component analysis, or other methods. This can be studied along with the selection of distance measures, which are critical to successful learning. A straightforward extension is weighing each dimension separately, leading to the so-called Relevance LVQ, see e.g. (Hammer and Villmann 2002). By providing additional order parameters, one can quantify the overlap of the relevance vectors to the discriminative orientation. The evolution of the overlap during learning is subsequently analysed by ordinary differential equations.

The statistical physics framework in this thesis will be useful to study various prototype-based learning methods, among others the popular Self Organising Maps (SOM) (Kohonen 1997). In principle, SOMs can be treated in a similar manner to our analysis of Neural Gas and would require systems with very many prototypes. While numerically these problems are solvable within our framework, the computations become exceedingly expensive, as the number of order parameters grows and higher dimensional numerical integrations are required. Alternative approaches may be called upon to analyse many-prototype systems, for instance by treating the prototypes as densities instead of individuals. These important extensions will provide meaningful insights into prototype-based learning schemes.
Appendix A

Statistics of the projections

For convenience, we combine the projections $h_S = w_S \cdot \xi$ and $b_\sigma = B_\sigma \cdot \xi$ defined in (3.10) into a $D$-dimensional vector, where $D = K + M$, as

$$x = \begin{pmatrix} h_1^\mu & h_2^\mu & \cdots & h_K^\mu & b_1^\mu & b_2^\mu & \cdots & b_M^\mu \end{pmatrix}^T$$  \hspace{1cm} (A.1)

In our analysis of on-line learning, we assume that $\xi$ is statistically independent from $w_S$, because $\xi^\mu$ is uncorrelated to all previous data and $w_S^{\mu-1}$. Therefore we observe that $h_S$ and $b_\sigma$ become correlated Gaussian random quantities following the Central Limit Theorem and can be fully described by their first and second moments, i.e. its conditional averages $\mu_\sigma = \langle x \rangle_\sigma$ and conditional covariance matrix $C_\sigma = \langle x \cdot x^T \rangle_\sigma$. We compute these averages in the following.

A.1 First order statistics

We compute the averages of the components of $x$ as follows:

$$\langle h_i \rangle_\sigma = \int_{R_N} \xi \cdot w_i \cdot p(\xi|\sigma) d\xi = w_i \cdot \int_{R_N} \xi \cdot p(\xi|\sigma) d\xi = w_i \cdot \ell_\sigma B_\sigma = \ell_\sigma R_\sigma$$  \hspace{1cm} (A.2)

$$\langle b_\tau \rangle_\sigma = \int_{R_N} \xi \cdot B_\tau \cdot p(\xi|\sigma) d\xi = B_\tau \cdot \int_{R_N} \xi \cdot p(\xi|\sigma) d\xi = B_\tau \cdot \ell_\sigma B_\sigma = \ell_\sigma T_\tau$$  \hspace{1cm} (A.3)

with $T_\tau = B_\tau \cdot B_\sigma$. To a large extent, we utilize orthonormal cluster center vectors, i.e. $B_\tau \cdot B_\sigma = \delta_{\tau\sigma}$ where $\delta$ is the Kronecker delta. The conditional first order moments $\mu_\sigma = \langle x \rangle_\sigma$ can be expressed in terms of order parameters as

$$\mu = \ell_\sigma \begin{pmatrix} R_1 \sigma & R_2 \sigma & \cdots & R_K \sigma & T_1 \sigma & T_2 \sigma & \cdots & T_M \sigma \end{pmatrix}^T$$  \hspace{1cm} (A.4)
A.2 Second order statistics

To compute the conditional variance $\langle x_n x_m \rangle_\sigma - \langle x_n \rangle_\sigma \langle x_m \rangle_\sigma$ we first look at the average

$$
\langle h_i h_j \rangle_\sigma = \left\langle \left( \sum_{k=1}^{N} (w_i)_k (\xi)_k \right) \left( \sum_{l=1}^{N} (w_j)_l (\xi)_l \right) \right\rangle_\sigma
$$

$$
= \left( \sum_{k=1}^{N} (w_i)_k (w_j)_k (\xi)_k (\xi)_k + \sum_{k=1}^{N} \sum_{l=1, l \neq k}^{N} (w_i)_k (w_j)_l (\xi)_k (\xi)_l \right)_\sigma
$$

$$
= \sum_{k=1}^{N} (w_i)_k (w_j)_k \langle (\xi)_k (\xi)_k \rangle_\sigma + \sum_{k=1}^{N} \sum_{l=1, l \neq k}^{N} (w_i)_k (w_j)_l \langle (\xi)_k (\xi)_l \rangle_\sigma
$$

$$
= \sum_{k=1}^{N} (w_i)_k (w_j)_k (\sigma + \ell^2_\sigma (B_\sigma)_k (B_\sigma)_k) + \sum_{k=1}^{N} \sum_{l=1, l \neq k}^{N} (w_i)_k (w_j)_l \ell^2_\sigma (B_\sigma)_k (B_\sigma)_l
$$

$$
= \nu_\sigma \sum_{k=1}^{N} (w_i)_k (w_j)_k + \ell^2_\sigma \sum_{k=1}^{N} \sum_{l=1}^{N} (w_i)_k (w_j)_l (B_\sigma)_k (B_\sigma)_l
$$

$$
= \nu_\sigma w_i \cdot w_j + \ell^2_\sigma (w_i \cdot B_\sigma)(w_j \cdot B_\sigma) = \nu_\sigma Q_{ij} + \ell^2_\sigma R_{\sigma} R_{\sigma} (A.5)
$$

Here we exploit the following

$$
\langle (\xi)_k (\xi)_k \rangle_\sigma = \nu_\sigma + \langle (\xi)_k \rangle_\sigma \langle (\xi)_k \rangle_\sigma = \nu_\sigma + \ell^2_\sigma (B_\sigma)_k (B_\sigma)_k
$$

and

$$
\langle (\xi)_k (\xi)_l \rangle_\sigma = \langle (\xi)_k \rangle_\sigma \langle (\xi)_l \rangle_\sigma = \ell^2_\sigma (B_\sigma)_k (B_\sigma)_l
$$

Hence we obtain the conditional second order moment, from Eqs. (A.5) and (A.2),

$$
\langle h_i h_j \rangle_\sigma - \langle h_i \rangle_\sigma \langle h_j \rangle_\sigma = \nu_\sigma Q_{ij} + \ell^2_\sigma R_{\sigma} R_{\sigma} - \ell_\sigma R_{\sigma} \ell_\sigma R_{\sigma} = \nu_\sigma Q_{ij} (A.6)
$$

Analogously, we get the second order statistics of $b$ and the covariance as follows:

$$
\langle h_i b_\rho \rangle_\sigma - \langle h_i \rangle_\sigma \langle b_\rho \rangle_\sigma = \nu_\sigma T_{\sigma b} + \ell^2_\sigma T_{\sigma b} T_{\rho b} - \ell_\sigma T_{\sigma b} \ell_\sigma T_{\rho b} = \nu_\sigma T_{\sigma b} (A.7)
$$

$$
\langle h_i b_\tau \rangle_\sigma - \langle h_i \rangle_\sigma \langle b_\tau \rangle_\sigma = \nu_\sigma R_{\sigma} + \ell^2_\sigma R_{\sigma} R_{\sigma} - \ell_\sigma R_{\sigma} \ell_\sigma R_{\sigma} = \nu_\sigma R_{\sigma} (A.8)
$$

The conditional covariance matrix $C_\sigma = \langle x_i x_j \rangle_\sigma$ can be written in terms of order parameters as

$$
C_\sigma = \nu_\sigma
\begin{pmatrix}
Q_{11} & \cdots & Q_{1K} & R_{11} & \cdots & R_{1M} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
Q_{1K} & \cdots & Q_{KK} & R_{K1} & \cdots & R_{KM} \\
R_{11} & \cdots & R_{K1} & T_{11} & \cdots & T_{I1} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
R_{1M} & \cdots & R_{KM} & T_{M1} & \cdots & T_{MM}
\end{pmatrix}
\tag{A.9}
$$
Appendix B

Averages

B.1 Form of the Differential Equations

In order to perform the ordinary differential equations described in (3.14), we need to plug in the values of 

\[ \langle f_S \rangle, \langle x_n f_S \rangle \text{ and } \langle f_S f_T \rangle \] (B.1)

Note that \( \langle f_S f_T \rangle \) is not required in the limit \( \eta \to 0 \), where terms proportional to \( \eta^2 \) can be neglected. We write the forms for the following algorithms: LVQ 2.1, LFM-W, GLVQ and RSLVQ.

LVQ 2.1

The general modulation function for LVQ 2.1 is described in Eq. (4.6) as

\[ f_S = \chi(c_S, y^\mu) \sum_{T : c_T \neq c_S} (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) \prod_{U \neq S, T} \Theta_{SU} \Theta_{TU}, \]

with \( \chi(c_S, y^\mu) = 1 \) if \( c_S = y^\mu \) and \( \chi(c_S, y^\mu) = -1 \) else. We can rewrite

\[ \Theta^\delta_{ST} = \Theta (d_T - d_S - \delta) \]

\[ = \Theta (\xi T - \xi S + \mathbf{w}_S \cdot \xi - \mathbf{w}_T \cdot \xi - \mathbf{w}_S^2 - \mathbf{w}_T^2 - \delta) \]

\[ = \Theta (-2h_{ST}^T + 2h_{ST}^S + Q_{TT} - Q_{SS} - \delta) \]

\[ = \Theta (\alpha_{ST} \cdot \mathbf{x} - \beta^\delta_{ST}), \] (B.2)

with \( \alpha_{ST} = (0, \ldots, +2, \ldots, -2, \ldots, 0) \) and \( \beta^\delta_{ST} = Q_{SS} - Q_{TT} - \delta \).

For two prototype systems with labels \( \mathbf{w}_S \) and \( \mathbf{w}_T \), we can simplify the above as

\[ f_S = \chi(c_S, y^\mu) (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}). \] (B.3)
And the required averages over the joint density (B.1) are calculated as

\[ \langle f_S \rangle = \langle \chi(c_S, y^\sigma_\sigma) (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) \rangle = \sum_{\sigma=1}^M p_\sigma \chi(c_S, y_\sigma) \langle \Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST} \rangle_\sigma \]

\[ \langle x_n f_S \rangle = \sum_{\sigma=1}^M p_\sigma \chi(c_S, y_\sigma) \langle x_n (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) \rangle_\sigma \]

\[ \langle f_S f_S \rangle = \langle \chi(c_S, y^\sigma_\sigma)^2 (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST})^2 \rangle = \sum_{\sigma=1}^M p_\sigma \langle (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) \rangle_\sigma \]

\[ \langle f_S f_T \rangle = \langle \chi(c_S, y^\sigma_\sigma) \chi(c_T, y^\sigma_T) (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST})^2 \rangle = -\sum_{\sigma=1}^M p_\sigma \langle (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) \rangle_\sigma \] (B.4)

The quantities \( \langle (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) \rangle_\sigma \) and \( \langle x_n (\Theta^{-\delta}_{ST} - \Theta^{+\delta}_{ST}) \rangle_\sigma \) are calculated in Appendix B.2

**LFM-W**

The general modulation function for LFM-W is described in Eq. (4.7) as

\[ f_S = \begin{cases} 
\sum_{K: c_T=y_\sigma} (\Theta_{KS} - \Theta^\delta_{KS}) \psi(S, K) & \text{if } c_S = y_\sigma \\
\sum_{J: c_T=y_\sigma} (\Theta^\delta_{SJ} - \Theta_{SJ}) \psi(J, S) & \text{else.}
\end{cases} \] (B.5)

with \( \psi(J, K) = \prod_{c_T=y_\sigma} \Theta_{JT} \prod_{U: c_U \neq y_\sigma} \Theta_{KU} \). With only two prototypes, both \( w_S \) and \( w_T \) are winners of their respective class, thus \( \psi(.) = 1 \) and the averages are

\[ \langle f_S \rangle = \sum_{\sigma: y_\sigma = c_S} p_\sigma (\Theta_{TS} - \Theta^\delta_{TS})_\sigma + \sum_{\sigma: y_\sigma \neq c_S} p_\sigma (\Theta_{ST} - \Theta^\delta_{ST})_\sigma \]

\[ \langle x_n f_S \rangle = \sum_{\sigma: y_\sigma = c_S} p_\sigma \langle x_n (\Theta_{TS} - \Theta^\delta_{TS}) \rangle_\sigma + \sum_{\sigma: y_\sigma \neq c_S} p_\sigma \langle x_n (\Theta_{ST} - \Theta^\delta_{ST}) \rangle_\sigma \] (B.6)

**GLVQ**

The general modulation function for GLVQ is described in Eq. (4.13) as

\[ f_S = \begin{cases} 
\sum_{K: c_T \neq y_\sigma} \left( \frac{2}{v_G} \phi \left( \frac{d_S - d_K}{v_G} \right) \right) \psi(S, K) & \text{if } c_S = y_\sigma \\
-\sum_{J: c_T = y_\sigma} \left( \frac{2}{v_G} \phi \left( \frac{d_J - d_S}{v_G} \right) \right) \psi(J, S) & \text{else.}
\end{cases} \] (B.7)
For two prototypes,

\[
\langle f_S \rangle = \sum_{\sigma : y_{\sigma} = c_S} p_{\sigma} \frac{2}{v_G} \langle \phi (\alpha_{ST} \cdot x - \beta_{ST}) \rangle_{\sigma} - \sum_{\sigma : y_{\sigma} \neq c_S} p_{\sigma} \frac{2}{v_G} \langle \phi (\alpha_{ST} \cdot x - \beta_{ST}) \rangle_{\sigma}
\]

\[
\langle x \circ f_S \rangle = \sum_{\sigma : y_{\sigma} = c_S} p_{\sigma} \frac{2}{v_G} \langle \phi (\alpha_{ST} \cdot x - \beta_{ST}) \rangle_{\sigma} - \sum_{\sigma : y_{\sigma} \neq c_S} p_{\sigma} \frac{2}{v_G} \langle \phi (\alpha_{ST} \cdot x - \beta_{ST}) \rangle_{\sigma}
\]

(B.8)

with \( \alpha_{ST} = \{ \ldots, -\frac{2}{v_G}, \ldots, \frac{2}{v_G}, 0, 0 \} \), \( \beta_{ST} = -\frac{Q_{SS} - Q_{TT}}{v_G} \).

The quantities \( \langle \phi (\alpha_{ST} \cdot x - \beta_{ST}) \rangle_{\sigma} \) are found in Eq. (B.26) in Appendix B.2.

**RSLVQ**

With one prototype representing each class, (4.15) become

\[
P_{c}(S|\xi^\mu) = \frac{\exp \left( -\frac{(\xi^\mu - w_S^\mu)^2}{2v_{soft}} \right)}{\exp \left( -\frac{(\xi^\mu - w_S^\mu)^2}{2v_{soft}} \right)} = 1
\]

\[
P(S|\xi^\mu) = \frac{\exp \left( -\frac{(\xi^\mu - w_S^\mu)^2}{2v_{soft}} \right)}{\sum_{T=1}^{K} \exp \left( -\frac{(\xi^\mu - w_T^\mu)^2}{2v_{soft}} \right)} = 1 + \sum_{T \neq S}^{K} \exp \left( \frac{-2\xi^\mu w_T^\mu + (w_S^\mu)^2 + 2\xi^\mu w_T^\mu - (w_T^\mu)^2}{2v_{soft}} \right)
\]

\[
= \frac{1}{1 + \sum_{T \neq S}^{K} \exp \left( \frac{-2h_S + Q_{SS} + 2h_T - Q_{TT}}{v_{soft}} \right)}
\]

(B.9)

where we defined

\( \alpha_{ST} = \{ \ldots, \frac{1}{v_{soft}}, \ldots, \frac{1}{v_{soft}}, 0, 0 \}, \ \beta_{ST} = -\frac{Q_{SS} - Q_{TT}}{2v_{soft}} \).

Therefore the RSLVQ modulation function becomes

\[
f_S = \frac{1}{v_{soft}} (\delta(c_S, y_S^\mu) - \Omega_S)
\]

(B.10)
B. Averages

where $\delta(x, y)$ is the Kronecker delta and

$$
\Omega_S = \frac{1}{1 + \sum_{T \neq S}^K \exp (\alpha_{ST} \cdot x - \beta_{ST})}
$$

We obtain the averages

$$
\langle f_S \rangle = \frac{1}{v_{\text{soft}}} \langle \delta(c_S, y_\sigma) - \Omega_S \rangle = \frac{1}{v_{\text{soft}}} \left( \sum_{\sigma: y_\sigma = c_S} p_\sigma - \sum_\sigma p_\sigma \langle \Omega_S \rangle_\sigma \right)
$$

$$
\langle x_n f_S \rangle = \begin{cases} 
\frac{1}{v_{\text{soft}}} \left( \sum_{\sigma: y_\sigma = c_S} p_\sigma \langle h_n \rangle_\sigma - \sum_\sigma p_\sigma \langle x_n \Omega_S \rangle_\sigma \right) & \text{if } n \leq K \\
\frac{1}{v_{\text{soft}}} \left( \sum_{\sigma: y_\sigma = c_S} p_\sigma \langle b_n - K \rangle_\sigma - \sum_\sigma p_\sigma \langle x_n \Omega_S \rangle_\sigma \right) & \text{if } n > K 
\end{cases}
$$

The required quantities $\langle \Omega_S \rangle_\sigma$ and $\langle x_n \Omega_S \rangle_\sigma$ are supplied in Appendix B.2.
B.2 Gaussian Averages

B.2.1 Two prototypes

For generic functions \( f_{ab} \equiv f(\alpha_{ab} \cdot x - \beta_{ab}) \), the quantities \( \langle f_{ab} \rangle_\sigma \) and \( \langle x_n f_{ab} \rangle_\sigma \) are computed as follows:

\[
\langle f_{ab} \rangle_\sigma = \frac{1}{(2\pi)^{D/2} \det(C_\sigma)^{1/2}} \int_{\mathbb{R}^D} f(\alpha_{ab} \cdot x - \beta_{ab}) \times \exp \left( -\frac{1}{2}(x - \mu_\sigma)^T C_\sigma^{-1}(x - \mu_\sigma) \right) \, dx
\]

\[
= \frac{1}{(2\pi)^{D/2} \det(C_\sigma)^{1/2}} \int_{\mathbb{R}^D} f(\alpha_{ab} \cdot x' + \alpha_{ab} \cdot \mu_\sigma - \beta_{ab}) \times \exp \left( -\frac{1}{2}(x')^T C_\sigma^{-1}(x') \right) \, dx',
\]

(B.13)

with the substitution \( x' = x - \mu_\sigma \). Because the covariance matrix \( C_\sigma \) is positive definite, \( C_\sigma^{1/2} \) exists. Defining \( x' = C_\sigma^{1/2} y \), we obtain \( x'^T C_\sigma^{-1} x' = y^2 \), \( dx' = (\det C_\sigma)^{1/2} dy \) and

\[
\langle f_{ab} \rangle_\sigma = \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} f(\alpha_{ab} C_\sigma^{1/2} y + \tilde{\beta}_{ab,\sigma}) \exp \left( -\frac{1}{2} y^2 \right) \, dy
\]

(B.14)

with \( \tilde{\beta}_{ab,\sigma} = \alpha_{ab} \cdot \mu_\sigma - \beta_{ab} \). Since \( \exp(-\frac{1}{2} y^2) \) has rotational invariance, it is possible to rotate the orthonormal coordinate system so that one axis, say \( \tilde{y} \), is aligned with vector \( \alpha_{ab} C_\sigma^{1/2} \). The remaining \( (D - 1) \) dimensions can be integrated over with \( f \exp(-\frac{1}{2} y^2) dy = \sqrt{2\pi} \) and we obtain

\[
\langle f_{ab} \rangle_\sigma = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(\|\alpha_{ab} C_\sigma^{1/2}\| \tilde{y} + \tilde{\beta}_{ab,\sigma}) \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y}
\]

\[
= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(\tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma}) \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y}
\]

(B.15)

with \( \tilde{\alpha}_{ab,\sigma} = \|\alpha_{ab} C_\sigma^{1/2}\| \). Next we calculate the quantity

\[
\langle x_n f_{ab} \rangle_\sigma = \frac{1}{(2\pi)^{D/2} \det(C_\sigma)^{1/2}} \int_{\mathbb{R}^D} x_n f(\alpha_{ab} \cdot x - \beta_{ab}) \times \exp \left( -\frac{1}{2}(x - \mu_\sigma)^T C_\sigma^{-1}(x - \mu_\sigma) \right) \, dx
\]

\[
= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} (C_\sigma^{1/2} y + \mu_\sigma) \cdot \alpha_{ab} C_\sigma^{1/2} y + \tilde{\beta}_{ab,\sigma} \exp \left( -\frac{1}{2} y^2 \right) \, dy
\]
B. Averages

\[
\frac{1}{(2\pi)^{D/2}} \int_{R^D} \left( C_{\sigma}^{1/2} \right) n f \left( \alpha_{\sigma} C_{\sigma}^{1/2} y + \bar{\beta}_{\sigma, \alpha} \right) \exp \left( -\frac{1}{2} y^2 \right) dy
\]
\[
+ (\mu_{\sigma})_n \langle f_{ab} \rangle_{\sigma}
\]

\[= \frac{1}{(2\pi)^{D/2}} I + (\mu_{\sigma})_n \langle f_{ab} \rangle_{\sigma} \tag{B.16} \]

where \( I \) is an integral to be computed. Consider \( I = (1/(2\pi)^{D/2}) \sum_{j=1}^{D} I_j \), we calculate

\[
I_j = \int_{R^D} \left( C_{\sigma}^{1/2} \right)_n j f \left( \alpha_{\sigma} C_{\sigma}^{1/2} y + \bar{\beta}_{\sigma, \alpha} \right) \exp \left( -\frac{1}{2} (y_j)^2 \right) d(y_j) \tag{B.17}
\]

by applying integration by parts \( \int u dv = uv - \int v du \) with

\[
u = -C_{\sigma}^{1/2} n_j \exp \left( -\frac{1}{2} (y_j)^2 \right)
\]

\[
du = f' \left( \alpha_{\sigma} C_{\sigma}^{1/2} y + \bar{\beta}_{\sigma, \alpha} \right) \left( \frac{\partial}{\partial (y_j)} \alpha_{\sigma} C_{\sigma}^{1/2} y \right) d(y_j)
\]

\[
= \sum_{i=1}^{D} (\alpha_{\sigma})_i (C_{\sigma}^{1/2})_{ij} f' \left( \alpha_{\sigma} C_{\sigma}^{1/2} y + \bar{\beta}_{\sigma, \alpha} \right) \exp \left( -\frac{1}{2} (y_j)^2 \right) d(y_j)
\]

\[
dv = (C_{\sigma}^{1/2})_n j (y_j) \exp \left( -\frac{1}{2} (y_j)^2 \right) d(y_j). \tag{B.18}
\]

Hence we have,

\[
I_j = \left[ (-) f \left( \alpha_{\sigma} C_{\sigma}^{1/2} y + \bar{\beta}_{\sigma, \alpha} \right) (C_{\sigma}^{1/2})_n j \exp \left( -\frac{1}{2} (y_j)^2 \right) \right]_{-\infty}^{\infty}
\]

\[- \left[ (-) \int_{R^D} (C_{\sigma}^{1/2})_n j \sum_{i=1}^{D} (\alpha_{\sigma})_i (C_{\sigma}^{1/2})_{ij} f' \left( \alpha_{\sigma} C_{\sigma}^{1/2} y + \bar{\beta}_{\sigma, \alpha} \right) \exp \left( -\frac{1}{2} (y_j)^2 \right) d(y_j) \right]
\]

\[= (C_{\sigma}^{1/2})_n j \sum_{i=1}^{D} (\alpha_{\sigma})_i (C_{\sigma}^{1/2})_{ij} \int_{R^D} f' \left( \alpha_{\sigma} C_{\sigma}^{1/2} y + \bar{\beta}_{\sigma, \alpha} \right) \exp \left( -\frac{1}{2} (y_j)^2 \right) d(y_j). \tag{B.19}
\]
Summing all $j$, we obtain

$$I = \frac{1}{(2\pi)^{D/2}} \sum_{j=1}^{D} \left( C_{\sigma}^{1/2}(\mu) \sum_{i=1}^{D} (\alpha_{ab})_{ij} \right) \int f'(\alpha_{ab} C_{\sigma}^{1/2} y + \tilde{\beta}_{ab,\sigma})$$

$$\times \exp\left(-\frac{1}{2}y^2\right) \, dy$$

$$= \frac{1}{(2\pi)^{D/2}} (C_{\sigma})_{n} \int_{\mathbb{R}^D} f'(\alpha_{ab} C_{\sigma}^{1/2} y + \tilde{\beta}_{ab,\sigma}) \exp\left(-\frac{1}{2}y^2\right) \, dy$$

(B.20)

Rotating the coordinate system similar to Eq. (B.15), we obtain the final form

$$\langle x_n f_{ab} \rangle_{\sigma} = \frac{(C_{\sigma} \alpha_{ab})_{n}}{\sqrt{2\pi}} \int_{\mathbb{R}^D} f'(\tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma}) \exp\left(-\frac{1}{2}\tilde{y}^2\right) \, d\tilde{y} + (\mu_{\sigma})_n \langle f_{ab} \rangle_{\sigma}.$$  

(B.21)

**LVQ 2.1, LFM-W**

The following quantities are required for two prototype LVQ2.1 and LFM-W:

$$\langle \Theta_{ab}^\delta - \Theta_{ab}^\gamma \rangle_{\sigma} \quad \text{and} \quad \langle x_n (\Theta_{ab}^\delta - \Theta_{ab}^\gamma) \rangle_{\sigma}.$$  

Averages of the form $\langle \Theta_{ab} \rangle_{\sigma}$ and $\langle x_n \Theta_{ab} \rangle_{\sigma}$ can be performed analytically; we refer to (Biehl et al. 2004) for the details and calculations and write down the final forms:

$$\langle \Theta_{ab} \rangle_{\sigma} = \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right).$$  

(B.22)

$$\langle x_n \Theta_{ab} \rangle_{\sigma} = \frac{(C_{\sigma} \alpha_{ab})_{n}}{\sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right)^2\right] + (\mu_{\sigma})_n \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right).$$  

(B.23)

where $\Phi(.)$ is the standard normal cumulative distribution function. Using the window rule, we obtain the required averages for LVQ 2.1 and LFM:

$$\langle \Theta_{ab}^\delta - \Theta_{ab}^\gamma \rangle_{\sigma} = \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right) - \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right).$$

$$\langle x_n (\Theta_{ab}^\delta - \Theta_{ab}^\gamma) \rangle_{\sigma} = \frac{(C_{\sigma} \alpha_{ab})_{n}}{\sqrt{2\pi}} \left\{ \exp\left[-\frac{1}{2} \left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right)^2\right] - \exp\left[-\frac{1}{2} \left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right)^2\right]\right\}$$

$$+ (\mu_{\sigma})_n \left[ \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right) - \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\right) \right].$$  

(B.24)
GLVQ

For GLVQ, the quantities $\langle \phi_{ab} \rangle_\sigma$ and $\langle x_n \phi_{ab} \rangle_\sigma$ are required to compute Eq. (B.8). From Eq. (B.15), we calculate the averages as follows:

$$\langle \phi_{ab} \rangle_\sigma = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \phi \left( \tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma} \right) \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y}$$

\[= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} \left( \tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma} \right)^2 \right) \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y} \]

\[= \frac{1}{\sqrt{2\pi}} \exp \left( \frac{1}{2} \tilde{\beta}_{ab,\sigma}^2 \right) \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} \left( \tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma} \right)^2 \right) \exp \left( -\frac{1}{2} \tilde{y}^2 - (\tilde{\alpha}_{ab,\sigma} \tilde{\beta}_{ab,\sigma}) \tilde{y} \right) d\tilde{y}. \]

\[= \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} \tilde{\beta}_{ab,\sigma}^2 \right) \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} \tilde{\alpha}_{ab,\sigma}^2 + 1 \right) \tilde{y} \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y}. \]

(B.25)

We use the substitution $\int_{-\infty}^\infty \frac{dx}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} ax^2 + bx \right) = \frac{1}{\sqrt{a}} \exp \left( \frac{b^2}{2a} \right)$ to obtain

$$\langle \phi_{ab} \rangle_\sigma = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} \tilde{\beta}_{ab,\sigma}^2 \right) \frac{1}{\sqrt{\tilde{\alpha}_{ab,\sigma}^2 + 1}} \exp \left( \frac{\tilde{\alpha}_{ab,\sigma}^2 \tilde{\beta}_{ab,\sigma}^2}{2(\tilde{\alpha}_{ab,\sigma}^2 + 1)} \right)$$

$$= \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\tilde{\alpha}_{ab,\sigma}^2 + 1}} \exp \left( \frac{1}{2} \tilde{\beta}_{ab,\sigma}^2 \right) \left( 1 - \frac{\tilde{\alpha}_{ab,\sigma}^2}{(\tilde{\alpha}_{ab,\sigma}^2 + 1)} \right) \exp \left( \frac{1}{2} \tilde{\beta}_{ab,\sigma}^2 \right). \quad \text{(B.26)}$$

The remaining average to be computed is

$$\langle x_n \phi_{ab} \rangle_\sigma = \frac{(C_\sigma \tilde{\alpha}_{ab})}{\sqrt{2\pi}} \int_{\mathbb{R}} \phi' \left( \tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma} \right) \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y} + (\mu_\sigma) \langle \phi_{ab} \rangle_\sigma. \quad \text{(B.27)}$$

Using $\frac{\partial}{\partial \tilde{y}} \exp(-y^2/2) = -y \exp(-y^2/2)$, we obtain

$$\langle x_n \phi_{ab} \rangle_\sigma = \frac{(C_\sigma \tilde{\alpha}_{ab})}{\sqrt{2\pi}} \int_{\mathbb{R}} \phi' \left( \tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma} \right) \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y}$$

\[+ \left( \mu_\sigma \right) \langle \phi_{ab} \rangle_\sigma \]

\[= -\frac{(C_\sigma \tilde{\alpha}_{ab})}{\sqrt{2\pi}} \int_{\mathbb{R}} (z + \tilde{\beta}_{ab,\sigma}) \phi' \left( z + \tilde{\beta}_{ab,\sigma} \right) \exp \left( \frac{1}{2} \tilde{y}^2 \right) d\tilde{y} \]

\[+ \left( \mu_\sigma \right) \langle \phi_{ab} \rangle_\sigma \]

(with $z = \tilde{\alpha}_{ab,\sigma} \tilde{y}$ and $dz = \tilde{\alpha}_{ab,\sigma} d\tilde{y}$).
B.2. Gaussian Averages

\[ -\frac{C_σ}{\sqrt{2π}} \exp\left(-\frac{1}{2} \tilde{β}_{ab,σ}^2\right) \int_R \frac{z + \tilde{β}_{ab,σ}}{\sqrt{2π}} \exp\left(-\frac{1}{2} \frac{z^2}{\tilde{α}_{ab,σ}^2} - z \tilde{β}_{ab,σ}\right) \, dz \]

+ (μ_σ)_n \langle φ_{ab}\rangle_σ .  

(B.28)

Now we use the substitutions

\[ \int_{-∞}^{∞} \frac{x}{\sqrt{2π}} \exp\left(-\frac{1}{2} ax^2 + bx\right) \, dx = \frac{b}{\sqrt{a^3}} \exp\left(\frac{b^2}{2a}\right) \]

\[ \int_{-∞}^{∞} \frac{x}{\sqrt{2π}} \exp\left(-\frac{1}{2} ax^2 + bx\right) = \frac{1}{\sqrt{a}} \exp\left(\frac{b^2}{2a}\right) . \]

Hence,

\[ \langle x_n φ_{ab}\rangle_σ = -\frac{C_σ}{\sqrt{2π}} \exp\left(-\frac{1}{2} \tilde{β}_{ab,σ}^2\right) \exp\left(-\frac{\tilde{β}_{ab,σ}^2}{2 \left(1 + 1/\tilde{α}_{ab,σ}^2\right)}\right) \]

\[ \times \left[-\frac{\tilde{β}_{ab,σ}}{\sqrt{1 + 1/\tilde{α}_{ab,σ}^2}} + \frac{\tilde{β}_{ab,σ}}{\sqrt{1 + 1/\tilde{α}_{ab,σ}^2}}\right] + (μ_σ)_n \langle φ_{ab}\rangle_σ . \]

(B.29)

and we obtain the final form

\[ \langle x_n φ_{ab}\rangle_σ = -\frac{C_σ}{\sqrt{2π}} \exp\left(-\frac{1}{2} \tilde{β}_{ab,σ}^2\right) \]

\[ \times \left[-\frac{\tilde{β}_{ab,σ}}{\sqrt{1 + 1/\tilde{α}_{ab,σ}^2}} + \frac{\tilde{β}_{ab,σ}}{\sqrt{1 + 1/\tilde{α}_{ab,σ}^2}}\right] + (μ_σ)_n \langle φ_{ab}\rangle_σ . \]

(B.30)

Note that both \( \langle φ_{ab}\rangle_σ \) and \( \langle x_n φ_{ab}\rangle_σ \) can be calculated analytically.

RSLVQ

For RSLVQ, the quantities \( \langle Ω_{ab}\rangle_σ \) and \( \langle x_n Ω_{ab}\rangle_σ \) are required.

\[ \langle Ω_{ab}\rangle_σ = \frac{1}{\sqrt{2π}} \int_R \frac{1}{1 + \exp(\tilde{α}_{ab,σ} \tilde{y} + \tilde{β}_{ab,σ})} \exp\left(-\frac{1}{2} \tilde{y}^2\right) \, d\tilde{y} \]

(B.31)

This one-dim. integration has to be solved numerically. Analogously, we obtain

\[ \langle x_n Ω_{ab}\rangle_σ = -\frac{C_σ}{\sqrt{2π}} \int_R \frac{\exp(\tilde{α}_{ab,σ} \tilde{y} + \tilde{β}_{ab,σ})}{(1 + \exp(\tilde{α}_{ab,σ} \tilde{y} + \tilde{β}_{ab,σ}))^2} \exp\left(-\frac{1}{2} \tilde{y}^2\right) \, d\tilde{y} \]

+ (μ_σ)_n \langle Ω_{ab}\rangle_σ , \]

(B.32)

which is also solved numerically.
B.2.2 Three prototypes

For generic function \( f_{ab} f_{cd} = f_1 (\alpha_{ab} \cdot x - \beta_{ab}) f_2 (\alpha_{cd} \cdot x - \beta_{cd}) \), the quantities \( (f_{ab} f_{cd})_k \) and \( (x, f_{ab} f_{cd})_k \) are required. For instance, the averages \( \langle \Theta_{ab} \Theta_{cd} \rangle \) are calculated as follows,

\[
\langle \Theta_{ab} \Theta_{cd} \rangle = \frac{1}{(2\pi)^{D/2}(\det C_\sigma)^{1/2}} \int_{\mathbb{R}^D} \Theta (\alpha_{ab} \cdot x - \beta_{ab}) \Theta (\alpha_{cd} \cdot x - \beta_{cd}) \\
\times \exp \left( -\frac{1}{2} (x - \mu_\sigma)^T C_\sigma^{-1} (x - \mu_\sigma) \right) dx
\]

\[
= \frac{1}{(2\pi)^{D/2}(\det C_\sigma)^{1/2}} \int_{\mathbb{R}^D} \Theta (\alpha_{ab} \cdot x' + \alpha_{ab} \cdot \mu_\sigma - \beta_{ab}) \\
\times \Theta (\alpha_{cd} \cdot x' + \alpha_{cd} \cdot \mu_\sigma - \beta_{cd}) \exp \left( -\frac{1}{2} x'^T C_\sigma^{-1} x' \right) dx'
\] (B.33)

(with the substitution \( x' = x - \mu_\sigma \)).

Because the covariance matrix \( C_\sigma \) is positive definite, \( C_\sigma^{1/2} \) exists. Defining \( x' = C_\sigma^{1/2} y \), we obtain \( x'^T C_\sigma^{-1} x' = y^2 \), \( dx' = (\det C_\sigma)^{1/2} dy \) and

\[
\langle \Theta_{ab} \Theta_{cd} \rangle = \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \Theta (\alpha_{ab} C_\sigma^{1/2} y + \alpha_{ab} \cdot \mu_\sigma - \beta_{ab}) \\
\times \Theta (\alpha_{cd} C_\sigma^{1/2} y + \alpha_{cd} \cdot \mu_\sigma - \beta_{cd}) \exp \left( -\frac{1}{2} y^2 \right) dy
\]

\[
= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \Theta (\alpha_{ab} C_\sigma^{1/2} y + \tilde{\beta}_{ab,\sigma}) \\
\times \Theta (\alpha_{cd} C_\sigma^{1/2} y + \tilde{\beta}_{cd,\sigma}) \exp \left( -\frac{1}{2} y^2 \right) dy
\] (B.34)

(where \( \tilde{\beta}_{ab,\sigma} = \alpha_{ab} \cdot \mu_\sigma - \beta_{ab} \)).

Since \( \exp(-\frac{1}{2} y^2) \) has rotational invariance, it is possible to rotate the orthonormal coordinate system \( y = (y_1 e_1 + y_2 e_2 + \ldots + y_N e_N) \) into \( y' = (y'_1 e'_1 + y'_2 e'_2 + \ldots + y'_N e'_N) \) where one axis, \( e'_1 \), is aligned with \( \alpha_{ab} C_\sigma^{1/2} \) and another axis, \( e'_2 \), lies on the plane spanned by \( \alpha_{ab} C_\sigma^{1/2} \) and \( \alpha_{cd} C_\sigma^{1/2} \). This is done by the Gram-Schmidt orthonormal transformation:

\[
e'_1 = \frac{\alpha_{ab} C_\sigma^{1/2}}{\| \alpha_{ab} C_\sigma^{1/2} \|}
\]

\[
e'_2 = \frac{\alpha_{cd} C_\sigma^{1/2} - (\alpha_{cd} C_\sigma^{1/2} \cdot e'_1)e'_1}{\| \alpha_{cd} C_\sigma^{1/2} - (\alpha_{cd} C_\sigma^{1/2} \cdot e'_1)e'_1 \|}
\] (B.35)

The other axes \( \{ e'_1, e'_2, \ldots, e'_N \} \) are orthogonal to both \( \alpha_{ab} C_\sigma^{1/2} \) and \( \alpha_{cd} C_\sigma^{1/2} \) and can be integrated over using the substitution \( \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \exp(-\frac{1}{2} z^2) dz = 1 \). We obtain from
B.2. Gaussian Averages

Eq. (B.34),

\[
\langle \Theta_{ab} \Theta_{cd} \rangle_{\sigma} = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \Theta \left( \alpha_{ab} C_{\sigma}^{1/2} \cdot \mathbf{y}'_1 + \tilde{\beta}_{ab,\sigma} \right) \\
\times \Theta \left( \alpha_{cd} C_{\sigma}^{1/2} \cdot \left( \mathbf{y}'_2 + \tilde{\beta}_{cd,\sigma} \right) \right) \exp \left( -\frac{1}{2}(y'^2_1 + y'^2_2) \right) dy'_1 dy'_2.
\]

We examine the Heaviside functions \( \Theta(x) = 1 \) if \( x > 0 \); 0 else. \( \Theta(\alpha_{ab} C_{\sigma}^{1/2} \mathbf{y}' + \tilde{\beta}_{ab,\sigma}) = 1 \) and \( \Theta(\alpha_{cd} C_{\sigma}^{1/2} \mathbf{y}' + \tilde{\beta}_{cd,\sigma}) = 1 \) if the following conditions are satisfied

\[
y' > y^*_1 \text{ with } y^*_1 = -\frac{\tilde{\beta}_{ab,\sigma}}{\alpha_{ab,\sigma}} - \alpha_{ab} C_{\sigma} y'_1,
\]

\[
y' > y^*_2 \text{ with } y^*_2 = -\frac{\tilde{\beta}_{cd,\sigma} \tilde{\alpha}_{ab,\sigma}}{\tilde{\alpha}_{cd,\sigma} \tilde{\alpha}_{ab,\sigma}} - \alpha_{cd} C_{\sigma} y'_1.
\]

where we defined \( \tilde{\alpha}_{ab,\sigma} = \|\alpha_{ab} C_{\sigma}^{1/2} \| \). Substituting the conditions into Eq. (B.36), we get

\[
\langle \Theta_{ab} \Theta_{cd} \rangle_{\sigma} = \frac{1}{(2\pi)^2} \int_{y'_1}^{\infty} \int_{y'_2}^{\infty} \exp \left( -\frac{1}{2}(y'^2_1 + y'^2_2) \right) dy'_2 dy'_1
\]

\[
= \frac{1}{(2\pi)^2} \int_{y'_1}^{\infty} \exp \left( -\frac{1}{2}y'^2_1 \right) \left( \int_{y'_2}^{\infty} \exp \left( -\frac{1}{2}y'^2_2 \right) dy'_2 \right) dy'_1.
\]

We get the final result in closed form as

\[
\langle \Theta_{ab} \Theta_{cd} \rangle_{\sigma} = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}}^{\infty} \exp \left( -\frac{1}{2}y'^2_1 \right) \Phi \left( \frac{\frac{\tilde{\beta}_{cd,\sigma} \tilde{\alpha}_{ab,\sigma} + \alpha_{cd} \alpha_{ab}}{\tilde{\alpha}_{cd,\sigma} \tilde{\alpha}_{ab,\sigma} - \alpha_{cd} \alpha_{ab}} y'_1}{\frac{1}{2} \alpha_{ab} C_{\sigma}^{1/2}} \right) dy'_1
\]

with \( \Phi(x) = \int_{x}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}a^2) da \). The one-fold integration in Eq. (B.38) has to be performed numerically.

The remaining average to be computed is

\[
\langle (x)_n \Theta_{ab} \Theta_{cd} \rangle_{\sigma} = \frac{1}{(2\pi)^{D/2}(\det C_{\sigma})^{1/2}} \int_{\mathbb{R}^D} (x)_n \Theta(\alpha_{ab} \cdot x - \beta_{ab}) \Theta(\alpha_{cd} \cdot x - \beta_{cd}) \\
\times \exp \left( -\frac{1}{2}(x - \mu_{\sigma})^T C_{\sigma}^{-1} (x - \mu_{\sigma}) \right) dx.
\]

(B.39)
Similar to Eq. (B.34), we obtain the form
\[ \langle x_n \Theta_{ab \Theta_{cd}} \rangle_\sigma = \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} (C^1_\sigma y)_n \Theta (\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \times \Theta (\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \exp(-\frac{1}{2} y^2) dy + (\mu_\sigma)_n (\Theta_{ab \Theta_{cd}})_\sigma \]
\[ = I + (\mu_\sigma)_n (\Theta_{ab \Theta_{cd}})_\sigma \] (B.40)

(\text{where } I \text{ is an integral to be computed}).

Consider the integrals contributing to I
\[ I_j = \int_{\mathbb{R}} (C^1_\sigma y)_j \Theta (\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \Theta (\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \times \exp(-\frac{1}{2} y^2_j) dy_j \] (B.41)

we perform integration by parts \( \int u dv = uv - \int v du \) with
\[ u = \Theta (\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \Theta (\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) , \]
\[ v = (C^1_\sigma y)_j \exp(-\frac{1}{2} y^2_j) , \]
\[ du = \frac{\partial}{\partial y_j} \left( \Theta (\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \right) \Theta (\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) d(y_j) \]
\[ + \Theta (\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \frac{\partial}{\partial y_j} \left( \Theta (\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \right) d(y_j) , \]
\[ dv = -(C^1_\sigma y)_j \exp(-\frac{1}{2} y^2_j) d(y_j) , \]

and we obtain
\[ I_j = \left[ -\Theta (\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \Theta (\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) (C^1_\sigma y)_j \exp(-\frac{1}{2} y^2_j) \right]_{-\infty}^{\infty} \]
\[ + \int_{\mathbb{R}} (C^1_\sigma y)_j \exp(-\frac{1}{2} y^2_j) \left[ \frac{\partial}{\partial y_j} \left( \Theta (\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \right) \Theta (\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \right] d(y_j) \]
\[ + \Theta (\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \frac{\partial}{\partial y_j} \left( \Theta (\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \right) d(y_j) \]
B.2. Gaussian Averages

\[\begin{align*}
B.2.1. \quad & \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \frac{\partial}{\partial y_j} \left( \Theta(\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \Theta(\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \right) \\
& \times \exp\left(-\frac{1}{2} y^2\right) d(y) \\
& \times \exp\left(-\frac{1}{2} y^2\right) d(y), \\
& \text{(B.42)}
\end{align*}\]

The sum over \( j \) gives

\[\begin{align*}
I &= \frac{1}{(2\pi)^{D/2}} \sum_{j=1}^{\mathbb{R}} (C^1_\sigma)_{nj} \left( \int_{\mathbb{R}^D} \frac{\partial}{\partial y_j} \left( \Theta(\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \Theta(\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \right) \\
& \times \exp\left(-\frac{1}{2} y^2\right) d(y) \\
& \times \exp\left(-\frac{1}{2} y^2\right) d(y) \right) \\
& \text{(B.43)}
\end{align*}\]

\[\begin{align*}
I &= \frac{1}{(2\pi)^{D/2}} \left( (C_\sigma \alpha_{ab}) n \int_{\mathbb{R}^D} \delta(\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \Theta(\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \\
& \times \exp\left(-\frac{1}{2} y^2\right) d(y) + (C_\sigma \alpha_{cd}) n \int_{\mathbb{R}^D} \delta(\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \\
& \times \Theta(\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \exp\left(-\frac{1}{2} y^2\right) d(y) \right), \\
& \text{(B.44)}
\end{align*}\]

where \( \delta(\cdot) \) is the Dirac-delta function. In the last step we have used

\[\begin{align*}
\frac{\partial}{\partial y_j} \left( \Theta(\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \right) &= \sum_{i=1}^{\mathbb{R}^D} (\alpha_{ab})_i (C^1_\sigma)_{ij} \left( \delta(\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \right).
\end{align*}\]

Calculating the first term only,

\[\begin{align*}
I_{ab} &= \frac{1}{(2\pi)^{D/2}} (C_\sigma \alpha_{ab}) n \int_{\mathbb{R}^D} \delta(\alpha_{ab} C^1_\sigma y + \tilde{\beta}_{ab,\sigma}) \Theta(\alpha_{cd} C^1_\sigma y + \tilde{\beta}_{cd,\sigma}) \exp\left(-\frac{1}{2} y^2\right) d(y),
\end{align*}\]
we rotate the coordinate system as in Eq. (B.36) and obtain the following

\[
I_{ab} = \frac{1}{2\pi} (C_\sigma \alpha_{ab}) \int_{\mathbb{R}^2} \delta \left( \alpha_{ab} C_\sigma^{1/2} \cdot y_1 e_1' + \tilde{\beta}_{ab,\sigma} \right) \\
\times \Theta (\alpha_{cd} C_\sigma^{1/2} \cdot (y_1 e_1' + y_2 e_2') + \tilde{\beta}_{cd,\sigma}) \exp \left( -\frac{1}{2} (y_1^2 + y_2^2) \right) \, dy_1 \, dy_2
\]

\[
= \frac{(C_\sigma \alpha_{ab})}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \delta \left( \tilde{\alpha}_{ab,\sigma} y_1' + \tilde{\beta}_{ab,\sigma} \right) \exp \left( -\frac{1}{2} y_1'^2 \right) \\
\times \Phi \left( \frac{\tilde{\beta}_{cd,\sigma} \tilde{\alpha}_{ab,\sigma} + (\alpha_{cd} C_\sigma \alpha_{ab}) y_1'}{\sqrt{\alpha_{ab,\sigma}^2 \alpha_{ab,\sigma}^2 - (\alpha_{cd} C_\sigma \alpha_{ab})^2}} \right) \, dy_1'.
\]

Substituting \( z = \tilde{\alpha}_{ab,\sigma} y_1' \),

\[
I_{ab} = \frac{(C_\sigma \alpha_{ab})}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \delta \left( z + \tilde{\beta}_{ab,\sigma} \right) \exp \left( -\frac{1}{2} \left( \frac{z}{\alpha_{ab,\sigma}} \right)^2 \right) \\
\times \Phi \left( \frac{\tilde{\beta}_{cd,\sigma} \alpha_{ab,\sigma} + (\alpha_{cd} C_\sigma \alpha_{ab}) z}{\alpha_{ab,\sigma} \sqrt{\alpha_{ab,\sigma}^2 \alpha_{ab,\sigma}^2 - (\alpha_{cd} C_\sigma \alpha_{ab})^2}} \right) \, dz.
\]

\[
= \frac{(C_\sigma \alpha_{ab})}{\sqrt{2\pi}} \alpha_{ab,\sigma} \exp \left( -\frac{1}{2} \frac{\tilde{\beta}_{ab,\sigma}^2}{\alpha_{ab,\sigma}^2} \right) \Phi \left( \frac{\tilde{\beta}_{cd,\sigma} \alpha_{ab,\sigma} - \tilde{\beta}_{ab,\sigma} (\alpha_{cd} C_\sigma \alpha_{ab})}{\alpha_{ab,\sigma} \sqrt{\alpha_{ab,\sigma}^2 \alpha_{ab,\sigma}^2 - (\alpha_{cd} C_\sigma \alpha_{ab})^2}} \right).
\]

(B.45)

Analogously we compute the second term in Eq. (B.44) and obtain the final form

\[
((x)_n \Theta_{ab} \Theta_{cd})_\sigma = \frac{(C_\sigma \alpha_{ab})}{\sqrt{2\pi}} \alpha_{ab,\sigma} \exp \left( -\frac{1}{2} \frac{\tilde{\beta}_{ab,\sigma}^2}{\alpha_{ab,\sigma}^2} \right) \Phi \left( \frac{\tilde{\beta}_{cd,\sigma} \alpha_{ab,\sigma} - \tilde{\beta}_{ab,\sigma} (\alpha_{cd} C_\sigma \alpha_{ab})}{\alpha_{ab,\sigma} \sqrt{\alpha_{ab,\sigma}^2 \alpha_{ab,\sigma}^2 - (\alpha_{cd} C_\sigma \alpha_{ab})^2}} \right) \\
+ (\mu_\sigma)_n (\Theta_{ab} \Theta_{cd})_\sigma.
\]

(B.46)

**Window**

With the addition of a window, these quantities are required:

\[
((\Theta_{ab} - \Theta_{ab}^*) \Theta_{cd})_\sigma = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left( -\frac{1}{2} y_1'^2 \right) \Phi \left( \frac{\tilde{\beta}_{cd,\sigma} \alpha_{ab,\sigma} + (\alpha_{cd} C_\sigma \alpha_{ab}) y_1'}{\sqrt{\alpha_{ab,\sigma}^2 \alpha_{ab,\sigma}^2 - (\alpha_{cd} C_\sigma \alpha_{ab})^2}} \right) \, dy_1'.
\]
\[
\langle (x)_n (\Theta^\delta_{ab} - \Theta^\gamma_{ab}) \Theta_{cd}\sigma \rangle
\]
\[
= \frac{(C_\sigma \alpha_{ab})_n}{\sqrt{2\pi} \alpha_{ab,\sigma}} \exp \left( -\frac{1}{2} \frac{\tilde{\beta}^2_{ab,\sigma}}{\overline{\alpha}_{ab,\sigma}^2} \right) \Phi \left( \frac{\tilde{\beta}_{cd,\sigma} \overline{\alpha}_{ab,\sigma} - \tilde{\beta}_{ab,\sigma} (\alpha_{cd} \sigma \alpha_{ab})}{\overline{\alpha}_{ab,\sigma} \sqrt{\overline{\alpha}_{cd,\sigma}^2 \overline{\alpha}_{ab,\sigma}^2 - (\alpha_{cd} \sigma \alpha_{ab})^2}} \right)
\]
\[
+ \frac{(C_\sigma \alpha_{cd})_n}{\sqrt{2\pi} \alpha_{cd,\sigma}} \exp \left( -\frac{1}{2} \frac{\tilde{\beta}^2_{cd,\sigma}}{\overline{\alpha}_{cd,\sigma}^2} \right) \Phi \left( \frac{\tilde{\beta}_{ab,\sigma} \overline{\alpha}_{cd,\sigma} - \tilde{\beta}_{ab,\sigma} (\alpha_{cd} \sigma \alpha_{ab})}{\overline{\alpha}_{ab,\sigma} \sqrt{\overline{\alpha}_{cd,\sigma}^2 \overline{\alpha}_{ab,\sigma}^2 - (\alpha_{cd} \sigma \alpha_{ab})^2}} \right)
\]
\[
- \frac{(C_\sigma \alpha_{ab})_n}{\sqrt{2\pi} \alpha_{ab,\sigma}} \exp \left( -\frac{1}{2} \frac{\tilde{\beta}^2_{ab,\sigma}}{\overline{\alpha}_{ab,\sigma}^2} \right) \Phi \left( \frac{\tilde{\beta}_{cd,\sigma} \overline{\alpha}_{ab,\sigma} - \tilde{\beta}_{ab,\sigma} (\alpha_{cd} \sigma \alpha_{ab})}{\overline{\alpha}_{ab,\sigma} \sqrt{\overline{\alpha}_{cd,\sigma}^2 \overline{\alpha}_{ab,\sigma}^2 - (\alpha_{cd} \sigma \alpha_{ab})^2}} \right)
\]
\[
- \frac{(C_\sigma \alpha_{cd})_n}{\sqrt{2\pi} \alpha_{cd,\sigma}} \exp \left( -\frac{1}{2} \frac{\tilde{\beta}^2_{cd,\sigma}}{\overline{\alpha}_{cd,\sigma}^2} \right) \Phi \left( \frac{\tilde{\beta}_{ab,\sigma} \overline{\alpha}_{cd,\sigma} - \tilde{\beta}_{ab,\sigma} (\alpha_{cd} \sigma \alpha_{ab})}{\overline{\alpha}_{ab,\sigma} \sqrt{\overline{\alpha}_{cd,\sigma}^2 \overline{\alpha}_{ab,\sigma}^2 - (\alpha_{cd} \sigma \alpha_{ab})^2}} \right)
\]
\[
+ (\mu_\sigma)_n \langle (\Theta^\delta_{ab} - \Theta^\gamma_{ab}) \Theta_{cd}\sigma \rangle. \quad (B.47)
\]
B.3 Generalization error

Two prototypes

We compute the generalization error from Eq. (4.20) as follows. For two prototypes \( w_+ \) and \( w_- \), we calculate \( \epsilon_g = \sum p_\sigma \epsilon_{g, \sigma} \) with

\[
\epsilon_{g, \sigma} = \left( \Theta_{-\sigma} \right)_+ = \Phi \left( \frac{\tilde{\beta}_{-\sigma, \sigma}}{\tilde{\alpha}_{-\sigma, \sigma}} \right)
\]

with \( \tilde{\alpha}_{ST, \sigma} = \sqrt{\alpha_{ST} C_{\sigma} \alpha_{ST}} \) and \( \tilde{\beta}_{ST, \sigma} = \alpha_{ST} \mu_{\sigma} - \beta_{ST} \). We refer the calculations to (Biehl et al. 2004). Plugging in the values, we obtain

\[
\epsilon_g = \Phi \left( \frac{Q_{\sigma} - Q_{-\sigma, \sigma} - 2 \ell_\sigma (R_{\sigma, \sigma} - R_{-\sigma, \sigma})}{2 \sqrt{\nu_\sigma} \sqrt{Q_{\sigma} - 2Q_{\sigma, \sigma} + Q_{-\sigma, \sigma}}} \right)
\]

By using \( Z_\sigma = Q_{\sigma} - Q_{-\sigma, \sigma} - 2 \ell(R_{\sigma, \sigma} - R_{-\sigma, \sigma}) \) and \( \Delta_q = \sqrt{Q_{++} - 2Q_{+-} + Q_{--}} \), we can calculate the derivative of the generalization error with respect to the order parameters \( \mathbf{O} = \{ R_{++}, R_{+-}, R_{-+}, R_{--}, Q_{++}, Q_{+-}, Q_{-+}, Q_{--} \}^T \) as follows:

\[
\frac{d \epsilon_g}{d \mathbf{O}} = \frac{1}{\sqrt{2\pi}2^{\sqrt{\nu_\sigma}}} \exp \left( -\frac{1}{2} \frac{Z_\sigma}{2 \sqrt{\nu_\sigma} \Delta_q} \right) \frac{d}{d \mathbf{O}} Z_\sigma \Delta_q
\]

where we used \( d \Phi(\tau)/d\tau = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}\tau^2) \). Derivations with respect to the order parameters yield

\[
\frac{d}{d \mathbf{O}} Z_+ = \begin{bmatrix}
-2\ell/\Delta_q \\
0 \\
+2\ell/\Delta_q \\
1/\Delta_q - Z_+/2\Delta_3^2 \\
-1/\Delta_q - Z_+/2\Delta_3^2
\end{bmatrix}, \quad
\frac{d}{d \mathbf{O}} Z_- = \begin{bmatrix}
0 \\
+2\ell/\Delta_q \\
0 \\
-1/\Delta_q - Z_-/(2\Delta_3^2) \\
1/\Delta_q - Z_-/(2\Delta_3^2)
\end{bmatrix}
\]

In the special case of \( p_+ = p_- = 0.5 \) and \( v_+ = v_- = v \), one obtains

\[
\frac{d \epsilon_{g, \sigma}}{d \mathbf{O}} = \sum \frac{d \epsilon_{g, \sigma}}{d \mathbf{O}} = \frac{1}{2\sqrt{2\pi}v} \exp \left( -\frac{1}{2} \frac{Z}{2 \sqrt{\nu} \Delta_q} \right) \begin{bmatrix}
-\ell/\Delta_q \\
+\ell/\Delta_q \\
+\ell/\Delta_q \\
-\ell/\Delta_q \\
-Z/(2\Delta_3^2) \\
Z/\Delta_3^2 \\
-Z/(2\Delta_3^2)
\end{bmatrix}
\]

(B.52)
B.4 Quantization error

Three prototypes
To compute the generalization error in systems with three prototypes \( w_S, w_T, w_U \), we require the quantity

\[
\epsilon_{g,\sigma} = \sum_{S:T\neq S} K \langle \Theta_{ST}\Theta_{SU} \rangle_{\sigma},
\]

(B.53)

where the averages are written in Eq. (B.38).

B.4 Quantization error

For two prototypes \( w_1 \) and \( w_2 \), we compute the quantization error

\[
E(W) = \frac{1}{2} \sum_{S=1}^{K} \left( \prod_{T\neq S} \Theta_{ST} \right) Q_{SS} - \sum_{S=1}^{K} \left( h_S \prod_{T\neq S} \Theta_{ST} \right)
\]

as follows. We calculate \( \epsilon_g = \sum_{\sigma} p_\sigma \epsilon_{g,\sigma} \) with

\[
E_\sigma(W) = \frac{1}{2} (\Theta_{12})_\sigma Q_{11} - (h_1 \Theta_{12})_\sigma + \frac{1}{2} (\Theta_{21})_\sigma Q_{22} - (h_2 \Theta_{21})_\sigma
\]

\[
= - (h_1 \Theta_{12})_\sigma + (\Theta_{21})_\sigma + \frac{Q_{11}}{2} (\Theta_{12})_\sigma + \frac{Q_{22}}{2} (\Theta_{21})_\sigma
\]

\[
= - (C_{12})_\sigma \exp \left( -\frac{Z^2}{2} \right) - (C_{21})_\sigma \exp \left( -\frac{(-Z)^2}{2} \right) - (\mu_\sigma)_1 \Phi(Z)
\]

\[
+ (\mu_\sigma)_2 \Phi(-Z) + \frac{Q_{11}}{2} \Phi(Z) + \frac{Q_{22}}{2} \Phi(-Z)
\]

\[
= \frac{\sqrt{\Delta Q}}{\sqrt{2\pi}} \exp \left( -\frac{Z^2}{2} \right) + \left( \frac{Q_{11}}{2} - \ell_\sigma R_{1\sigma} \right) \Phi(Z) + \left( \frac{Q_{22}}{2} - \ell_\sigma R_{2\sigma} \right) \Phi(-Z),
\]

with

\[
\Delta Q = \sqrt{Q_{11} - 2Q_{12} + Q_{22}},
\]

\[
Z = (2\ell_\sigma (R_{1\sigma} - R_{2\sigma}) - Q_{11} + Q_{22}) / (2\sqrt{\Delta Q}).
\]

(B.54)
Appendix C

Annealed approximation

Practical training procedures aim at an efficient minimization of the cost function. In the statistical physics interpretation of the learning process, this corresponds to low temperatures. While the correct treatment of finite $T$ requires sophisticated techniques such as the replica trick, a useful approximation method which is technically less difficult, can be employed to perform the quenched average $\langle \ln Z \rangle_{\mathcal{E}}$. In the annealed approximation, we approximate the function

$$\langle \ln Z \rangle_{\mathcal{E}} \approx \ln \langle Z \rangle_{\mathcal{E}} \quad (C.1)$$

Equivalently,

$$\langle \exp (-\beta H(W)) / Z \rangle_{\mathcal{E}} \approx \langle \exp (-\beta H(W)) \rangle_{\mathcal{E}} / \langle Z \rangle_{\mathcal{E}} \quad (C.2)$$

The annealed approximation becomes exact in the limit $\beta \to 0$ and coincides with the explicit treatment of this limit (Seung et al. 1992). At low temperatures the annealed free energy yields only an upper bound to the correct one, but the hope is that the position of minima in terms of the $\{R_{S\sigma}, Q_{ST}\}$ is similar. The scheme has proven useful in predicting qualitative behavior of many learning systems, e.g. (Engel and van den Broeck 2001, Sompolinsky and Tishby 1990). The validity of the annealed approximation is discussed systematically in, for instance, (Solla and Levin 1992, Seung et al. 1992). The average partition function can be rewritten as

$$\langle Z \rangle_{\mathcal{E}} = \int d\mu(W) \langle \exp \left[ -\beta \sum_{\mu=1}^{P} e(\xi^\mu, W) \right] \rangle_{\mathcal{E}}$$

$$= \int d\mu(W) \prod_{\mu=1}^{P} \langle \exp [-\beta e(\xi^\mu, W)] \rangle_{\xi}$$

$$= \int d\mu(W) \exp \left[ P \ln \langle \exp [-\beta e(\xi^\mu, W)] \rangle_{\xi} \right]$$

$$= \int d\mu(W) \exp [-\alpha NG_A(W)]$$

with $G_A = -\ln \langle \exp (-\beta e(\xi^\mu, W)) \rangle_{\xi}$

$$\langle Z \rangle_{\mathcal{E}} = \int d\mu(W) \exp [-\alpha NG_A(W)]$$

where $G_A$ involves an average over one random input only. Only for $\beta \to 0$ this average can be absorbed into the exponent and we recover the high temperature result.
The corresponding free energy function is
\[
f = \alpha G_A(\{R_{S\sigma}, Q_{ST}\}) - s(\{R_{S\sigma}, Q_{ST}\}) \tag{C.7}
\]
Unlike the high temperature limit, in the annealed approximation the empirical average over training set \(D\) is distinguished from the input density. The training and test set performances are given by
\[
\epsilon_{\text{train}} = \frac{1}{p} \sum_{j=1}^{p} e(W, \xi''_j) = \alpha^{-1} \frac{\partial}{\partial \beta} f(\{R_{S\sigma}, Q_{ST}\})
\]
\[
\epsilon_{\text{test}} = \langle e \rangle_{\xi} \tag{C.8}
\]
which have to be evaluated in the minimum of \(f\).

The calculation of \(G_A\) can be done analytically for arbitrary \(\beta\) and two prototypes with the WTA cost function
\[
e(W, \xi) = \frac{1}{2} \sum_{i=1}^{K} d(w_i, \xi) \prod_{j \neq i} \Theta_{ij} - \frac{1}{2} \xi^2 \tag{C.9}
\]
we can obtain, for two prototypes,
\[
G_A = \ln \int d\mu(\xi) \exp [-\beta e(\xi, W)]
\]
\[
= - \ln \langle \exp -\beta \left[ \frac{1}{2} \left( \Theta_{12} (\xi - w_1)^2 + \Theta_{21} (\xi - w_2)^2 \right) - \frac{1}{2} \xi^2 \right] \rangle_{\xi} \tag{C.10}
\]
\[
= - \ln \langle \exp \left[ \Theta_{12} \left( w_1 \cdot \xi - \frac{1}{2} w_1^2 \right) + \Theta_{21} \left( w_2 \cdot \xi - \frac{1}{2} w_2^2 \right) \right] \rangle_{\xi} \tag{C.11}
\]
Substituting
\[
\langle \exp(\Theta_{12} f + \Theta_{21} g) \rangle_{\xi} = \langle \Theta_{12} \exp(f) \rangle_{\xi} + \langle \Theta_{21} \exp(g) \rangle_{\xi} \tag{C.12}
\]
we obtain
\[
G_A = - \ln \left\{ \langle \Theta_{12} \exp \left( w_1 \cdot \xi - \frac{1}{2} w_1^2 \right) \rangle_{\xi} \right. \\
+ \left. \langle \Theta_{21} \exp \left( w_2 \cdot \xi - \frac{1}{2} w_2^2 \right) \rangle_{\xi} \right\} \tag{C.13}
\]
\[
= - \ln \left\{ \exp \left( -\beta w_1^2 \right) \langle \Theta_{12} \exp (\beta w_1 \cdot \xi) \rangle_{\xi} + \exp \left( -\beta w_2^2 \right) \langle \Theta_{21} \exp (\beta w_2 \cdot \xi) \rangle_{\xi} \right\} \tag{C.14}
\]
Converting into order parameters and projections,

\[ G_A = -\ln \left\{ \exp \left( -\frac{1}{2} \beta Q_{11} \right) \langle \Theta_{12} \exp (\beta h_1) \rangle \xi + \right. \]
\[ \left. + \exp \left( -\frac{1}{2} \beta Q_{22} \right) \langle \Theta_{21} \exp (\beta h_2) \rangle \xi \right\} \]

(C.16)

For the annealed approximation, the quantity of importance is

\[ \langle \Theta_{ab} \exp (\beta x_n) \rangle_k = \frac{1}{(2\pi)^{D/2}(\det C)_{1/2}} \int_{R^D} \exp (\beta x_n) \Theta_{ab} \]
\[ \exp \left( -\frac{1}{2} (x - \mu_k)^T C_{\sigma}^{-1} (x - \mu_k) \right) dx \]

(17)

\[ = \frac{1}{(2\pi)^{D/2}(\det C)_{1/2}} \int_{R^D} \exp (\beta (x'_n + \mu_k)) \Theta_{ab} \]
\[ \exp \left( -\frac{1}{2} x'^T C_{\sigma}^{-1} x' \right) dx' \]

(18)

We begin calculations by partitioning elements in the matrix \( C_{\sigma}^{-1} \) and vector \( x' \) associated with term \( n \) as follows

\[ x'^T C_{\sigma}^{-1} x' = \sum_j \sum_i x'_i (C_{\sigma}^{-1})_{ij} x'_j \]
\[ = \sum_{j \neq n} \sum_{i \neq n} x'_i (C_{\sigma}^{-1})_{ij} x'_j + \sum_{j \neq n} x'_n (C_{\sigma}^{-1})_{nj} x'_j + \sum_{i \neq n} x'_i (C_{\sigma}^{-1})_{in} x'_n \]
\[ + x'_n (C_{\sigma}^{-1})_{nn} x'_n \]
\[ = x'_n (C_{\sigma}^{-1})_{[n,n]} x_n + 2(C_{\sigma}^{-1})_{[n,n]} x'_n x'_n + (C_{\sigma}^{-1})_{[n,n]} (x'_n)^2 \]

(19)

where \( \bar{n} \) are elements complementary to \( n \). We use the shorthand

\[ \tilde{C}_k^{-1} = (C_{\sigma}^{-1})_{[\bar{n},\bar{n}]} \]

(20)

It can be easily shown that because \( C_{\sigma} \) is positive definite, \( \tilde{C}_k \) is also positive definite. Therefore \( \tilde{C}_k^{1/2} \) exists and we can define \( x'_n^T = \tilde{C}_k^{1/2} y \) to obtain

\[ x'^T C_{\sigma}^{-1} x' = y^2 + 2 \left( (C_{\sigma}^{-1})_{[n,n]} \tilde{C}_k^{1/2} y \right) x'_n + (C_{\sigma}^{-1})_{[n,n]} (x'_n)^2 \]

(21)
and from $d\mathbf{x} = \det(\tilde{C}_k^{1/2})d\mathbf{y} = (\det \tilde{C}_k)^{1/2}d\mathbf{y}$,

$$
(\Theta_{ab} \exp(\beta \mathbf{x}_n))_k
= \frac{\exp(\beta(\mu_b)_n) \det(\tilde{C}_k)^{1/2}}{(2\pi)^D/2 \det(C_\sigma)^{1/2}} \int_{\mathbb{R}^D} \exp(\beta \mathbf{x}_n') \Theta_{ab} \exp\left( -\frac{1}{2} \left( y^2 + 2(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k^{1/2} \mathbf{y} \cdot \mathbf{x}_n' + (C_\sigma^{-1})_{nn}(x_n')^2 \right) \right) dy \ dx_n'
= \frac{\exp(\beta(\mu_b)_n)}{(2\pi)^D/2} \sqrt{\det \tilde{C}_k} \int_\mathbb{R} \exp\left( \beta \mathbf{x}_n' - \frac{1}{2} (C_\sigma^{-1})_{nn}(x_n')^2 \right) \left\{ \int_{\mathbb{R}^{D-1}} \Theta_{ab} \exp\left( -\frac{1}{2} \left( y^2 + 2(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k^{1/2} \mathbf{y} \cdot \mathbf{x}_n' \right) \right) dy \right\} dx_n'
= \left\{ \int_{\mathbb{R}^{D-1}} \Theta_{ab} \exp\left( -\frac{1}{2} \left( y^2 + 2(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k^{1/2} \mathbf{y} \cdot \mathbf{x}_n' \right) \right) dy \right\}_{I}
$$

Calculating the integral $I$ by completing the square,

$$
I = \int_{\mathbb{R}^{D-1}} \Theta_{ab} \exp\left( -\frac{1}{2} \left( y^2 + 2(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k^{1/2} \mathbf{y} \cdot \mathbf{x}_n' \right) \right) dy
= \int_{\mathbb{R}^{D-1}} \Theta_{ab} \exp\left( -\frac{1}{2} \left( y + x_n'(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k^{1/2} \right)^2 + \frac{1}{2} (x_n')^2(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k(C_\sigma^{-1})_{[n,n]} \right) dy
= \exp\left( \frac{1}{2} (x_n')^2(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k(C_\sigma^{-1})_{[n,n]} \right) \times \int_{\mathbb{R}^{D-1}} \Theta_{ab} \exp\left( -\frac{1}{2} \left( y + x_n'(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k^{1/2} \right)^2 \right) dy
= \exp\left( \frac{1}{2} (x_n')^2(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k(C_\sigma^{-1})_{[n,n]} \right) \int_{\mathbb{R}^{D-1}} \Theta_{ab} \exp\left( -\frac{1}{2} \mathbf{z}^2 \right) d\mathbf{z}
$$

where we defined $\mathbf{z} = \mathbf{y} + x_n'(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k^{1/2}$. The corresponding coordinate transforms for the Heaviside function is as follows

$$
\Theta_{ab} = \Theta\left( \alpha_{ab} \cdot \mathbf{x} + \tilde{\beta}_{ab,k} \right)
= \Theta\left( (\alpha_{ab})_n \cdot \tilde{x} + (\alpha_{ab})_n x_n' + \tilde{\beta}_{ab,k} \right)
= \Theta\left( (\alpha_{ab})_n \tilde{C}_k^{1/2} \mathbf{y} + (\alpha_{ab})_n x_n' + \tilde{\beta}_{ab,k} \right)
= \Theta\left( (\alpha_{ab})_n \tilde{C}_k^{1/2} \left( \mathbf{z} - x_n'(C_\sigma^{-1})^T_{[n,n]} \tilde{C}_k^{1/2} \right) + (\alpha_{ab})_n x_n' + \tilde{\beta}_{ab,k} \right)
= \Theta\left( (\alpha_{ab})_n \tilde{C}_k^{1/2} \mathbf{z} + \tilde{\beta}_{ab,k} \right)
$$
where

$$\tilde{\beta}_{ab,k} = \beta_{ab,k} + (\alpha_{ab})_n x'_n - (\alpha_{ab})_n \tilde{C}_k(C^{-1})_{[n,n]} x'_n. \quad (25)$$

Rotating the coordinate system so that one of the axes \( \tilde{z} \) coincides with \( (\alpha_{ab})_n \tilde{C}_k^{1/2} \) and substituting \( \int_R \exp(-\frac{1}{2}z^2)dz = \sqrt{2\pi} \)

$$I = \exp \left( \frac{1}{2} (x'_n)^T (C^{-1})_{[n,n]} \tilde{C}_k(C^{-1})_{[n,n]} x'_n \right) (2\pi)^{(D-2)/2} \times \int_R \Theta \left( \| (\alpha_{ab})_n \tilde{C}_k^{1/2} \| \tilde{z} + \tilde{\beta}_{ab,k} \right) \exp \left( -\frac{1}{2} \tilde{z}^2 \right) d\tilde{z} \quad (26)$$

$$= \exp \left( \frac{1}{2} (x'_n)^T (C^{-1})_{[n,n]} \tilde{C}_k(C^{-1})_{[n,n]} x'_n \right) (2\pi)^{(D-2)/2} \times \int_{-\infty}^{\infty} \beta'_{ab,k} \exp \left( -\frac{1}{2} \tilde{z}^2 \right) d\tilde{z} \quad (27)$$

where

$$\tilde{\alpha}_{ab,k} = \| (\alpha_{ab})_n \tilde{C}_k^{1/2} \| \quad (28)$$

Substituting \( \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp \left( -\frac{1}{2} \tilde{z}^2 \right) dt \)

$$I = \exp \left( \frac{1}{2} (C^{-1})_{[n,n]} \tilde{C}_k(C^{-1})_{[n,n]} (x'_n)^2 \right) (2\pi)^{(D-1)/2} \Phi \left( \frac{\tilde{\beta}_{ab,k}}{\tilde{\alpha}_{ab,k}} \right) \quad (29)$$

Hence, we obtain the form

$$\langle \Theta_{ab} \exp (\beta x_n) \rangle_k = \frac{\exp (\beta(\mu_k)_n)}{\sqrt{2\pi}} \sqrt{\frac{\det \tilde{C}_k}{\det C_{[n,n]}}} \int_R \exp \left( \frac{1}{2} (C^{-1})_{nn} (x'_n)^2 \right) + \frac{1}{2} (C^{-1})_{[n,n]} \tilde{C}_k(C^{-1})_{[n,n]} (x'_n)^2 \Phi \left( \frac{\tilde{\beta}_{ab,k}}{\tilde{\alpha}_{ab,k}} \right) dx'_n \quad (30)$$
We use the substitution
\[ \int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}ax^2 + bx\right) \Phi(cx + d) = \frac{1}{\sqrt{a}} \exp\left(\frac{b^2}{2a}\right) \Phi\left(\frac{bc + ad}{\sqrt{a^2 + ac^2}}\right) \]
and get the final form,
\[ \langle \Theta_{ab} \exp (\beta x_n) \rangle_k = \exp (\beta (\mu_k)_n) \sqrt{\frac{\det \tilde{C}_k}{\det C_\sigma}} \left\{ \frac{1}{\sqrt{p}} \exp\left(\frac{q^2}{2p}\right) \Phi\left(\frac{qr + ps}{\sqrt{p^2 + ps}}\right) \right\} \]
with
\[ p = (C_\sigma^{-1})_{nn} - (C_\sigma^{-1})_{[n,n]}^T \tilde{C}_k (C_\sigma^{-1})_{[n,n]}, \quad q = \beta, \]
\[ r = (\alpha_{ab})_n - (\alpha_{ab})_n \tilde{C}_k (C_\sigma^{-1})_{[n,n]}, \quad s = \frac{\tilde{\alpha}_{ab,k}}{\alpha_{ab,k}} \]
Publications

Journal Papers


Conference Proceedings


- Aree Witoelar, Anarta Ghosh and Michael Biehl - *Phase transitions in Vector Quantization*, in M. Verleysen (ed.), Proceedings of European Symposium on
Artificial Neural Networks (ESANN) 2008, d-side, Evere, Belgium, pp. 221-226, 2008.


**Technical Report**


**Posters**

- Aree Witoeelar, Michael Biehl and Anarta Ghosh - *Dynamics of Multiple Prototype LVQ*. Scientific ICT Research Event Netherlands (SIREN), Utrecht, the Netherlands, 2006.

- Michael Biehl, Anarta Ghosh and Aree Witoeelar - *Learning from examples in Neural Gas and Vector Quantization*, DPG Frhjahrstagung, Dresden, Germany, 2006.

Samenvatting

In dit proefschrift presenteren we een theoretisch raamwerk voor het onderzoeken van op prototypen-gebaseerde trainvoorschriften. De analyse is uitgevoerd gebruikmakend van concepten uit de statistische fysica, welke een exact wiskundig voor- schrift van het systeem mogelijk maken. Hiermee kan het systeem uitgedrukt worden in zijn karakteristieke eigenschappen, de zogenaamde ordeparameters. De traindynamica en equilibrische leertoestanden worden volledig beschreven door de set van ordeparameters. Het proefscenario dat we gebruiken om de leerende- vectorkwantisatie (learning vector quantisation, LVQ)-algoritmen is beschreven in Hoofdstuk 2.

In Hoofdstuk 3 en Hoofdstuk 4 bekijken we het online-trainen en beschouwen we de leerdynamica van (niet-gesuperviseerde) vectorkwantisatie en (gesuperviseerde) lerende-vectorkwantisatie. Het leergedrag wordt bestudeerd in termen van de evolutie van de ordeparameters, voorgeschreven door gewone differentiaalvergelijkingen. In de niet-gesuperviseerde dataclusteranalyse tonen we de voordelen van neurale-gasalgoritmen (en.: “Neural Gas” of “NG”-algoritmen) op algoritmen waarin de winnaar alles krijgt (en.: “Winner-Takes-All” of “WTA”-algoritmen). Ten eerste kan NG, ten opzichte van WTA, de convergentiesnelheid verbeteren wanneer de prototypen slecht zijn geinitialiseerd. Verder bereikt NG robuustheid ten opzichte van de begincondities. Desondanks zou NG alsnog kunnen convergeren naar lokale minima en bereikt het niet altijd de best mogelijke kwantisatienauwkeurigheid.

Onder de gesuperviseerde LVQ-schema’s vergelijken we de voordelen en nadelen van verscheidene venstergebaseerde voorbeeldselectieschema’s in detail, waaronder LVQ 2.1, het “leren van fouten” (textiten.: learning from mistakes, LFM), gegeneneraliseerde LVQ (GLVQ) en robuuste zachte LVQ (textiten.: robust soft LVQ, RSLVQ). De gevoeligheid van de leercurven wordt bestudeerd voor ieder van deze algoritmen. Verrassend genoeg produceert LVQ 2.1 de optimale lineaire besliss-
In Hoofdstuk 5 en Hoofdstuk 6 gebruiken we de statistische fysica van het offline leren voor het analyseren van de LVQ-algoritmen. Het trainen wordt geïnterpreteerd als een stochastische minimalisatie van de kostenfunctie op de dataset \( D \), waarin de formele temperatuur \( T \) het aantal vrijheidsgraden bepaalt. We onderzoeken de equilibrische eigenschappen van WTA en rang-gebaseerde vectorkwantisatiesystemen, gebruikmakend van de hoge temperatuurslimiet en de zogenaamde afkoelbenadering. We ondertekken dat, gelet op de initialisatie, neurale gas meer robuust is dan WTA. Dit komt goed overeen met de resultaten die verkregen zijn bij de analyse van online-leren. We vinden faseovergangen in het leerstadium: een kritisch aantal voorbeelden moet aan het systeem worden gepresenteerd voordat de onderliggende structuur van de data kan geïdentificeerd worden. De aard van de overgang is continu in systemen met twee prototypen en discreet in systemen met drie prototypen. Dit is zeer relevant vanuit een praktisch perspectief: optimalisatiestrategieën zullen mislukken wanneer te weinig voorbeelddata beschikbaar zijn en metastabiele toestanden voor systemen met drie prototypen zouden een grote vertraging tijdens het leren kunnen veroorzaken.

Ten slotte analyseren we de kostenfunctie van LFM, LVQ 2.1 en RSVQ in de hoge-temperatuurslimiet. LFM werkt onverwacht slecht op dit leerprobleem, in vergelijking met de optimaal bereikbare fout. Met een goed gekozen gewichtsverval toont LVQ 2.1 een beter generalisatievermogen dan LFM en RSLVQ voor systemen met zowel twee als drie prototypen. In systemen met drie prototypen vinden we continue faseovergangen tussen prototypeconfiguraties. We observeren een kritisch benodigde grootte van de trainset om effectief gebruik te maken van alle beschikbare prototypen en om onderliggende structuren binnen de data te vinden.

**De relatie tussen online- en offline-analyses**

Terwijl de generieke aanpak van online- en offline-analyses technisch verschillend zijn, complemeneren beide technieken elkaar sterk. Online analyse geeft de mogelijkheid tot het onderzoeken van de convergentiesnelheid en haar afhankelijkheid van leerpparameters en begincondities, welke essentieel afwezig zijn in offline analyse. Ook geeft het de mogelijkheid tot het studeren van de heuristieke praktische LVQ-voorschriften die niet gebaseerd zijn op kostenfuncties. Bovendien kunnen ongelimiteerde kostenfuncties zoals in LVQ 2.1 leiden tot sterk divergerend gedrag, wat niet behandelde kan worden met offline analyse. Met behulp van online-analyse kunnen we het niet-triviale asymptotische leergedrag schatten. Aan de andere kant
worden in offline analyse de karakteristieken van het landschap van een kostenfunctie rigoureu se geëvalueerd, om alle mogelijke equilibrische toestanden te vinden, zonder de convergentietijd expliciet te beschouwen. De stabiele en metastabiele toestanden onthullen de presentie van vaste punten die online-analyse zou kunnen aantreffen. De kwalitatieve overeenkomst tussen de vondsten van de twee methodes is reeds zichtbaar. Onze online-analyse van niet-gesuperviseerd leren in Hoofdstuk 3 tentoonstelt significante vertraging van het leerproces voor WTA met een slechte beginstoestand en het voordeel van rang-gebaseerde NG-algoritmen.

In het offline leren correspondeert dit met suboptimaal equilibrische toestanden, bediscussieerd in Sectie 5.4.1 waarbij triviale minima bestaan bij grote prototype-lengten. Bij offline NG demonstreren we ook hoe de rangparameter \( \lambda \) het energielandschap verandert, wat de typische ontsnappingstijd van de suboptimale configuratie beïnvloedt. In beide analyses demonstreren we dat NG meer robuust is dan WTA. In online-leren zijn de asymptotische configuratie van NG onafhankelijk van de begincondities. Dit is weerspiegeld in offline leren door het verdwijnen van de metastabiele toestanden in het vloeiende energielandschap, gegeven een voldoende grote \( \lambda \).

In het algemeen worden de resultaten van de twee analyses identiek bij lange leertijden \( \tilde{\alpha} \to \infty \) voor online- en bij grote trainingsets \( \hat{\alpha} \to \infty \) voor offline-trainen. We vergelijken gesuperviseerde problemen in Hoofdstuk 4 en Hoofdstuk 5 en bevestigen identieke resultaten voor RSLVQ algoritmen. De afhankelijkheid van asymptotische configuraties op de begincondities wordt uitgelegd door de degeneratie van de kostenfunctieminima. De vergelijking van LFM is meer beperkt, doordat de prototypen samenvallen in de eerder genoemde limieten. We observeren dat het gedrag bij een grote \( \tilde{\alpha} \) dat van een grote \( \hat{\alpha} \) benadert.

Ten einde beide aanpakken volledig te vergelijken, moeten verschillende geformeerde limieten opgelost worden. We nemen bijvoorbeeld kleine leernelheden aan in de online-analyse. Dit zou echter corresponderen met een lage-temperatuur, offline analyse, wat een volledige behandeling van de replicaemethode vereist. In dit werk kan dit alleen benaderd worden met de afkoelbenadering. Omgekeerd staat offline analyse het toe om het leren van beperkte datasets te onderzoeken, en we observeren een verschillende train- en generalisatiefout bij eindige temperaturen: zie de resultaten van de afkoelvoorbeelden in Sectie 5.4.3. Dit correspondeert in de online-analyse met het hergebruiken van voorbeelden uit een eindige dataset, wat correlaties veroorzaakt tussen trainvoorbeelden en het systeem. Het onderwerp van online-leren met eindige trainsets in neurale netwerken wordt onderzocht in bijvoorbeeld (Barber and Sollich 1998, Rae et al. 1999).

De aanwezigheid van symmetrie produceert een bepaald leergedrag dat zich uitzet in zowel online- als offline-analyse. Dit uit zich gedurende de specialisatiefase
van prototypen in niet-gesuperviseerd leren. Permutatiesymmetrie tussen twee of meer prototypen, dat is, gelijke configuratie door uitwisseling van prototypen, beperkt het leerproces binnen de thermodynamische limiet sterk.

In online-leren vereisen prototypen lange leertijden om te ontsnappen van de door-symmetrie veroorzaakte vaste punten, welke plateaus veroorzaken in de leercurve van VQ. Symmetrie in offline-leren veroorzaakt competitie in toestanden met een variërende entropie, waarbij de niet-gespecialiseerde toestand de voorkeur heeft bij kleine trainsets. Dit resulteert op zijn beurt in vertraagd leren, waarbij specialisatie alleen voor komt bij trainsets groter dan een kritieke omvang. In gesuperviseerde leerproblemen worden gepermuteerde symmetrieën gebroken door verschillende toewijzing van klassen aan de prototypen, maar zij blijven bestaan tussen prototypen van dezelfde klasse. Derhalve worden leerplateaus en vertraagd leren alleen geobserveerd in LVQ systemen met meerdere prototypen.


