Learning Vector Quantization: The Dynamics of Winner-Takes-All Algorithms

Michael Biehl[1], Anarta Ghosh[1], and Barbara Hammer[2]
1- Rijksuniversiteit Groningen - Mathematics and Computing Science
P.O. Box 800, NL-9700 AV Groningen - The Netherlands
2- Clausthal University of Technology - Institute of Computer Science
D-98678 Clausthal-Zellerfeld - Germany

Abstract

Winner-Takes-All (WTA) prescriptions for Learning Vector Quantization (LVQ) are studied in the framework of a model situation: Two competing prototype vectors are updated according to a sequence of example data drawn from a mixture of Gaussians. The theory of on-line learning allows for an exact mathematical description of the training dynamics, even if an underlying cost function cannot be identified. We compare the typical behavior of several WTA schemes including basic LVQ and unsupervised Vector Quantization. The focus is on the learning curves, i.e. the achievable generalization ability as a function of the number of training examples.

Keywords: Learning Vector Quantization, prototype-based classification, Winner-Takes-All algorithms, on-line learning, competitive learning

1 Introduction

Learning Vector Quantization (LVQ) as originally proposed by Kohonen [10] is a widely used approach to classification. It is applied in a variety of practical problems, including medical image and data analysis, e.g. proteomics, classification of satellite spectral data, fault detection in technical processes, and language recognition, to name only a few. An overview and further references can be obtained from [1].

LVQ procedures are easy to implement and intuitively clear. The classification of data is based on a comparison with a number of so-called prototype vectors. The similarity is frequently measured in terms of Euclidean

distance in feature space. Prototypes are determined in a training phase from labeled examples and can be interpreted in a straightforward way as they directly represent typical data in the same space. This is in contrast with, say, adaptive weights in feedforward neural networks or support vector machines which do not allow for immediate interpretation as easily. Among the most attractive features of LVQ is the natural way in which it can be applied to multi-class problems.

In the simplest so-called *hard* or *crisp* schemes any feature vector is assigned to the closest of all prototypes and the corresponding class. In general, several prototypes will be used to represent each class. Extensions of the deterministic assignment to a probabilistic *soft* classification are conceptually straightforward but will not be considered here.

Plausible training prescriptions exist which employ the concept of on-line competitive learning: Prototypes are updated according to their distance from a given example in a sequence of training data. Schemes in which only the winner, i.e. the currently closest prototype is updated have been termed Winner-Takes-All algorithms and we will concentrate on this class of prescriptions here.

The ultimate goal of the training process is, of course, to find a classifier which labels novel data correctly with high probability, after training. This so-called generalization ability will be in the focus of our analysis in the following.

Several modifications and potential improvements of Kohonen's original LVQ procedure have been suggested. They aim at achieving better approximations of Bayes optimal decision boundaries, faster or more robust convergence, or the incorporation of more flexible metrics, to name a few examples [5, 8–10, 16].

Many of the suggested algorithms are based on plausible but purely heuristic arguments and they often lack a thorough theoretical understanding. Other procedures can be derived from an underlying cost function, such as *Generalized Relevance LVQ* [8, 9] or *LVQ2.1*, the latter being a limit case of a statistical model [14, 15]. However, the connection of the cost functions with the ultimate goal of training, i.e. the generalization ability, is often unclear. Furthermore, several learning rules display instabilities and divergent behavior and require modifications such as the *window rule* for LVQ2.1 [10].

Clearly, a better theoretical understanding of the training algorithms should be helpful in improving their performance and in designing novel, more efficient schemes.

In this work we employ a theoretical framework which makes possible a systematic investigation and comparison of LVQ training procedures. We

consider on-line training from a sequence of uncorrelated, random training data which is generated according to a model distribution. Its purpose is to define a non-trivial structure of data and facilitate our analytical approach. We would like to point out, however, that the training algorithms do not make use of the form of this distribution as, for instance, density estimation schemes.

The dynamics of training is studied by applying the successful theory of online learning [3, 6, 12] which relates to ideas and concepts known from Statistical Physics. The essential ingredients of the approach are (1) the consideration of high-dimensional data and large systems in the so-called thermodynamic limit and (2) the evaluation of averages over the randomness or *disorder* contained in the sequence of examples. The typical properties of large systems are fully described by only a few characteristic quantities. Under simplifying assumptions, the evolution of these so-called *order parameters* is given by deterministic coupled ordinary differential equations (ODE) which describe the dynamics of on-line learning exactly in the thermodynamic limit. For reviews of this very successful approach to the investigation of machine learning processes consult, for instance, [3, 6, 17].

The formalism enables us to compare the dynamics and generalization ability of different WTA schemes including basic LVQ1 and unsupervised Vector Quantization (VQ). The analysis can readily be extended to more general schemes, approaching the ultimate goal of designing novel and efficient LVQ training algorithms with precise mathematical foundations.

The paper is organized as follows: in the next section we introduce the model, i.e. the specific learning scenarios and the assumed statistical properties of the training data. The mathematical description of the training dynamics is briefly summarized in section 3, technical details are given in an appendix. Results concerning the WTA schemes are presented and discussed in section 4 and we conclude with a summary and an outlook on forthcoming projects.

2 The model

2.1 Winner-Takes-All algorithms

We study situations in which input vectors $\boldsymbol{\xi} \in I\!R^N$ belong to one of two possible classes denoted as $\sigma = \pm 1$. Here, we restrict ourselves to the case of two prototype vectors w_S where the label $S = \pm 1$ (or $\pm$ for short) corresponds to the represented class.

In all WTA-schemes, the squared Euclidean distances $d_S(\boldsymbol{\xi}) = (\boldsymbol{\xi} - w_S)^2$ are evaluated for $S = \pm 1$ and the vector $\boldsymbol{\xi}$ is assigned to class σ if $d_{+\sigma} < d_{-\sigma}$.

We investigate incremental learning schemes in which a sequence of single, uncorrelated examples $\{\boldsymbol{\xi}^\mu, \sigma^\mu\}$ is presented to the system. The analysis can be applied to a larger variety of algorithms but here we treat only updates of the form

$$\boldsymbol{w}_S^\mu = \boldsymbol{w}_S^{\mu-1} + \Delta \boldsymbol{w}_S^\mu \quad \text{with} \quad \Delta \boldsymbol{w}_S^\mu = \frac{\eta}{N}\, \Theta_S^\mu \;\; g(S, \sigma^\mu) \;\; \left(\boldsymbol{\xi}^\mu - \boldsymbol{w}_S^{\mu-1}\right). \quad (1)$$

Here, the vector $\boldsymbol{w}_S^\mu$ denotes the prototype after presentation of μ examples and the learning rate η is rescaled with the vector dimension N. The Heaviside term

$$\Theta_S^\mu = \Theta(d_{-S}^\mu - d_{+S}^\mu) \qquad \text{with} \;\; \Theta(x) = \begin{cases} 1 & \text{if } \; x > 0. \\ 0 & \text{else} \end{cases}$$

singles out the current prototype $\boldsymbol{w}_S^{\mu-1}$ which is closest to the new input $\boldsymbol{\xi}^\mu$ in the sense of the measure $d_S^\mu = (\boldsymbol{\xi}^\mu - \boldsymbol{w}_S^{\mu-1})^2$. In this formulation, only the *winner*, say, $\boldsymbol{w}_S$ can be updated whereas the *looser* $\boldsymbol{w}_{-S}$ remains unchanged. The change of the winner is always along the direction $\pm(\boldsymbol{\xi}^\mu - \boldsymbol{w}_S^{\mu-1})$. The function $g(S, \sigma^\mu)$ further specifies the update rule. Here, we focus on three special cases of WTA learning:

I) **LVQ1:** $g(S, \sigma) = S\sigma = +1\,(\text{resp. } -1)$ for $S = \sigma\,(\text{resp. } S \neq \sigma)$. This extension of competitive learning to labeled data corresponds to Kohonen's original LVQ1. The update is towards $\boldsymbol{\xi^\mu}$ if the example belongs to the class represented by the winning prototype, the *correct winner*. On the contrary, a *wrong winner* is moved away from the current input.

II) **LVQ+:** $g(S, \sigma) = \Theta(S\sigma) = +1\,(\text{resp. } 0)$ for $S = \sigma\,(\text{resp. } S \neq \sigma)$. In this scheme the update is non-zero only for a correct winner and, then, always positive. Hence, a prototype $\boldsymbol{w}_S$ can only accumulate updates from its own class $\sigma = S$. We will use the abbreviation LVQ+ for this prescription.

III) **VQ:** $g(S, \sigma) = 1$. This update rule disregards the actual data label and always moves the winner towards the example input. It corresponds to unsupervised Vector Quantization and aims at finding prototypes which yield a good representation of the data in the sense of Euclidean distances. The choice $g(S, \sigma) = 1$ can also be interpreted as describing two prototypes which represent the same class and compete for updates from examples of this very class, only.

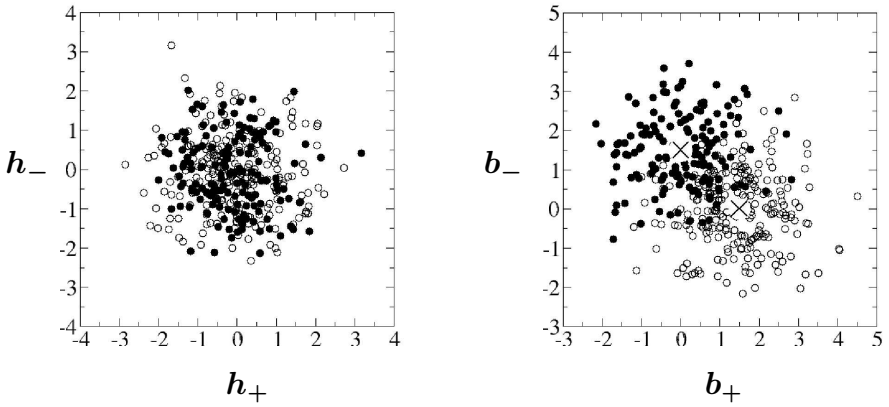

Figure 1: Data as generated according to the model density (3) in $N = 200$ dimension with $p_- = 0.6, p_+ = 0.4$. Open (filled) circles represent 240 (160) vectors $\boldsymbol{\xi}$ from clusters centered about orthonormal vectors $\ell \boldsymbol{B}_+$ ($\ell \boldsymbol{B}_-$) with $\ell = 1.5$. The left panel shows the projections $h_\pm = \boldsymbol{w}_\pm \cdot \boldsymbol{\xi}$ of the data on a randomly chosen pair of orthogonal unit vectors $\boldsymbol{w}_\pm$, whereas the right panel displays $b_\pm = \boldsymbol{B} \cdot \boldsymbol{\xi}$, i.e. the plane spanned by $\boldsymbol{B}_+, \boldsymbol{B}_-$; crosses mark the position of the cluster centers.

Note that the VQ procedure (III) can be readily formulated as a stochastic gradient descent with respect to the quantization error

$$e(\boldsymbol{\xi}^\mu) = \sum_{S=\pm 1} \frac{1}{2} (\boldsymbol{\xi}^\mu - \boldsymbol{w}_S^{\mu-1})^2 \, \Theta(d_{-S}^\mu - d_{+S}^\mu), \tag{2}$$

see e.g. [7] for details. While intuitively clear and well motivated, LVQ1 (I) and LVQ+ (II) lack such a straightforward interpretation and relation to a cost function.

2.2 The model data

In order to analyse the behavior of the above algorithms we assume that randomized data is generated according to a model distribution $P(\boldsymbol{\xi})$. As a simple yet non-trivial situation we consider input data drawn from a binary mixture of Gaussian clusters

$$P(\boldsymbol{\xi}) = \sum_{\sigma=\pm 1} p_\sigma P(\boldsymbol{\xi} \,|\, \sigma) \quad \text{with} \quad P(\boldsymbol{\xi} \,|\, \sigma) = \frac{1}{\sqrt{2\pi}^N} \exp\left[-\frac{1}{2} \left(\boldsymbol{\xi} - \lambda \boldsymbol{B}_\sigma \right)^2 \right] \tag{3}$$

where the weights p_σ correspond to the prior class membership probabilities and $p_+ + p_- = 1$. Clusters are centered about $\lambda \boldsymbol{B}_+$ and $\lambda \boldsymbol{B}_-$, respectively. We assume that $\boldsymbol{B}_\sigma \cdot \boldsymbol{B}_\tau = \Theta(\sigma\tau)$, i.e. $\boldsymbol{B}_\sigma^2 = 1$ and $\boldsymbol{B}_+ \cdot \boldsymbol{B}_- = 0$. The latter condition fixes the location of the cluster centers with respect to the origin while the parameter λ controls their separation.

We consider the case where the cluster membership σ coincides with the class label of the data. The corresponding classification scheme is not linearly separable because the Gaussian contributions $P(\boldsymbol{\xi}\,|\sigma)$ overlap. According to Eq. (3) a vector $\boldsymbol{\xi}$ consists of statistically independent components with unit variance. Denoting the average over $P(\boldsymbol{\xi}\,|\,\sigma)$ by $\langle\cdots\rangle_\sigma$ we have, for instance, $\langle\xi_j\rangle_\sigma = \lambda(\boldsymbol{B}_\sigma)_j$ for a component and correspondingly

$$\left\langle \boldsymbol{\xi}^2 \right\rangle_\sigma = \sum_{j=1}^{N} \left\langle \xi_j^2 \right\rangle_\sigma = \sum_{j=1}^{N} \left(1 + \langle \xi_j \rangle_\sigma^2\right) = N + \lambda^2.$$

Averages over the full $P(\boldsymbol{\xi})$ will be written as $\langle\cdots\rangle = \sum_{\sigma=\pm1} \langle\cdots\rangle_\sigma$.

Note that in high dimensions, i.e. for large N, the Gaussians overlap significantly. The cluster structure of the data becomes only apparent when projected into the plane spanned by $\{\boldsymbol{B}_+, \boldsymbol{B}_-\}$. However projections in a randomly chosen two-dimensional subspace would overlap completely. Figure 1 illustrates the situation for Monte Carlo data in $N = 200$ dimensions. In an attempt to learn the classification scheme, the relevant directions $\boldsymbol{B}_\pm \in I\!R^N$ have to be identified to a certain extent. Obviously this task becomes highly non-trivial for large N.

3 The dynamics of learning

The following analysis is along the lines of on-line learning, see e.g. [3,6,12]. for a comprehensive overview and example applications. In this section we briefly outline the treatment of WTA algorithms in LVQ and refer to appendix A for technical details.

The actual configuration of prototypes is characterized by the projections

$$R_{S\sigma}^\mu = \boldsymbol{w}_S^\mu \cdot \boldsymbol{B}_\sigma \quad \text{and} \quad Q_{ST}^\mu = \boldsymbol{w}_S^\mu \cdot \boldsymbol{w}_T^\mu, \quad \text{for} \;\; S,T,\sigma = \pm 1 \qquad (4)$$

The self-overlaps Q_{++} and Q_{--} specify the lengths of vectors $\boldsymbol{w}_\pm$, whereas the remaining five overlaps correspond to projections, i.e. angles between $\boldsymbol{w}_+$ and $\boldsymbol{w}_-$ and between the prototypes and the center vectors $\boldsymbol{B}_\pm$.

The algorithm (1) directly implies recursions for the above defined quantities upon presentation of a novel example:

$$\begin{aligned} N(R^{\mu}_{S\sigma} - R^{\mu-1}_{S\sigma}) &= \eta\, \Theta^{\mu}_{S}\, g(S,\sigma^{\mu})\, \left(b^{\mu}_{S} - R^{\mu-1}_{S\sigma}\right) \\ N(Q^{\mu}_{ST} - Q^{\mu-1}_{ST}) &= \eta\, \Theta^{\mu}_{S}\, g(S,\sigma^{\mu})\, \left(h^{\mu}_{T} - Q^{\mu-1}_{ST}\right) \\ &\quad + \eta\, \Theta^{\mu}_{T}\, g(T,\sigma^{\mu})\, \left(h^{\mu}_{S} - Q^{\mu-1}_{ST}\right) \\ &\quad + \eta^2\, \Theta^{\mu}_{S}\, \Theta^{\mu}_{T}\, g(S,\sigma^{\mu})\, g(T,\sigma^{\mu}) + \mathcal{O}(1/N) \end{aligned} \tag{5}$$

Here, the actual input $\boldsymbol{\xi}^{\mu}$ enters only through the projections

$$h^{\mu}_{S} = \boldsymbol{w}^{\mu-1}_{S} \cdot \boldsymbol{\xi}^{\mu} \quad \text{and} \quad b^{\mu}_{\sigma} = \boldsymbol{B}_{\sigma} \cdot \boldsymbol{\xi}^{\mu}. \tag{6}$$

Note in this context that $\Theta^{\mu}_{S} = \Theta(Q^{\mu-1}_{-S-S} - 2h^{\mu}_{-S} - Q^{\mu-1}_{SS} + 2h^{\mu}_{S})$ also does not depend on $\boldsymbol{\xi}^{\mu}$ explicitly.

An important assumption is that all examples in the training sequence are independently drawn from the model distribution and, hence, are uncorrelated with previous data and with $\boldsymbol{w}^{\mu-1}_{\pm}$. As a consequence, the statistics of the projections (6) are well known for large N. By means of the Central Limit Theorem their joint density becomes a mixture of Gaussians, which is fully specified by the corresponding conditional first and second moments:

$$\begin{aligned} &\left\langle h^{\mu}_{S} \right\rangle_{\sigma} = \lambda R^{\mu-1}_{S\sigma}, \quad \left\langle b^{\mu}_{\tau} \right\rangle_{\sigma} = \lambda \Theta(\tau\sigma), \quad \left\langle h^{\mu}_{S} h^{\mu}_{T} \right\rangle_{\sigma} - \left\langle h^{\mu}_{S} \right\rangle_{\sigma} \left\langle h^{\mu}_{T} \right\rangle_{\sigma} = Q^{\mu-1}_{ST} \\ &\left\langle h^{\mu}_{S} b^{\mu}_{\tau} \right\rangle_{\sigma} - \left\langle h^{\mu}_{S} \right\rangle_{\sigma} \left\langle b^{\mu}_{\tau} \right\rangle_{\sigma} = R^{\mu-1}_{S\tau}, \quad \left\langle b^{\mu}_{\rho} b^{\mu}_{\tau} \right\rangle_{\sigma} - \left\langle b^{\mu}_{\rho} \right\rangle_{\sigma} \left\langle b^{\mu}_{\tau} \right\rangle_{\sigma} = \Theta(\rho\tau), \end{aligned} \tag{7}$$

see the appendix for derivations. This observation enables us to perform an average of the recursions w.r.t. the latest example data in terms of Gaussian integrations. Details of the calculation are presented in appendix A, see also [4]. On the right hand sides of (5) terms of order $(1/N)$ have been neglected using, for instance, $\left\langle \boldsymbol{\xi}^2 \right\rangle / N = 1 + \lambda^2/N \approx 1$ for large N.

The limit $N \to \infty$ has further simplifying consequences. First, the averaged recursions can be interpreted as ordinary differential equations (ODE) in *continuous training time* $\alpha = \mu/N$. Second, the overlaps $\{R_{S\sigma}, Q_{ST}\}$ as functions of α become *self–averaging* with respect to the random sequence of examples. Fluctuations of these quantities, as for instance observed in computer simulations of the learning process, vanish with increasing N and the description in terms of mean values is sufficient. For a detailed mathematical discussion of this property see [11].

Given initial conditions $\{R_{S\sigma}(0), Q_{ST}(0)\}$, the resulting system of coupled ODE can be integrated numerically. This yields the evolution of overlaps (4) with increasing α in the course of training. The behavior of the

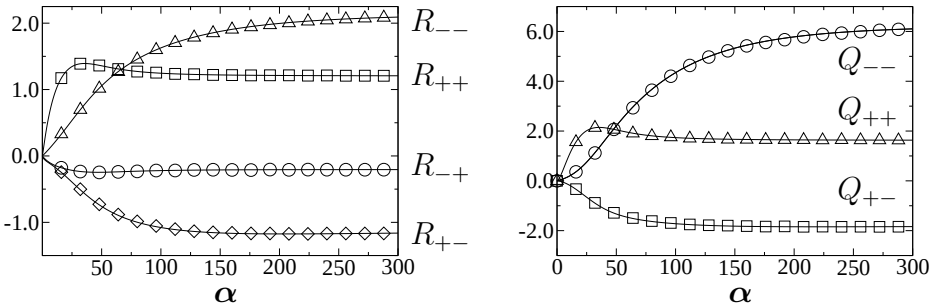

Figure 2: The characteristic overlaps $R_{S\sigma}$ (left panel) and Q_{ST} (right panel) vs. the rescaled number of examples $\alpha = \mu/N$ for LVQ1 with $p_+ = 0.8$, $\lambda = 1$, and $\eta = 0.2$. The solid lines display the result of integrating the system of ODE, symbols correspond to Monte Carlo simulations for $N = 200$ on average over 100 independent runs. Error bars are smaller than the symbol size. Prototypes were initialized close to the origin with $R_{S\sigma} = Q_{+-} = 0$ and $Q_{++} = Q_{--} = 10^{-4}$.

system will depend on the characteristics of the data, i.e. the separation λ, the learning rate η, and the actual algorithm as specified by the choice of $g(S, \sigma)$ in Eq. (1). Monte Carlo simulations of the learning process are in excellent agreement with the $N \to \infty$ theory for dimensions as low as $N = 200$ already, see Figure 2 for a comparison in the case of LVQ1. Our simulations furthermore confirm that the characteristic overlaps $\{R_{S\sigma}(\alpha), Q_{ST}(\alpha)\}$ are self-averaging quantities: their standard deviation determined from independent runs vanishes as $N \to \infty$, details will be published elsewhere.

The success of learning can be quantified as the probability of misclassifying novel random data, the *generalization error* $\epsilon_g = \sum_{\sigma=\pm 1} p_\sigma \, \langle \Theta_{-\sigma} \rangle_\sigma$. Performing the averages is done along the lines discussed above, see the appendix and [4]. One obtains ϵ_g as a function of the overlaps $\{Q_{ST}, R_{S\sigma}\}$:

$$\epsilon_g \;=\; \sum_{S=\pm 1} p_\sigma \, \Phi \left(\frac{Q_{SS} - Q_{-S-S} - 2\lambda \left[R_{SS} - R_{-SS} \right]}{\sqrt{Q_{++} + Q_{--} - 2Q_{+-}}} \right), \tag{8}$$

where $\phi(x) = \int_{-\infty}^{x} \frac{dt}{\sqrt{2\pi}} e^{-t^2/2}$. Hence, by inserting $\{Q_{ST}(\alpha), R_{S\sigma}(\alpha)\}$ we can evaluate the learning curve $\epsilon_g(\alpha)$, i.e. the typical generalization error as achieved from a sequence of αN examples.

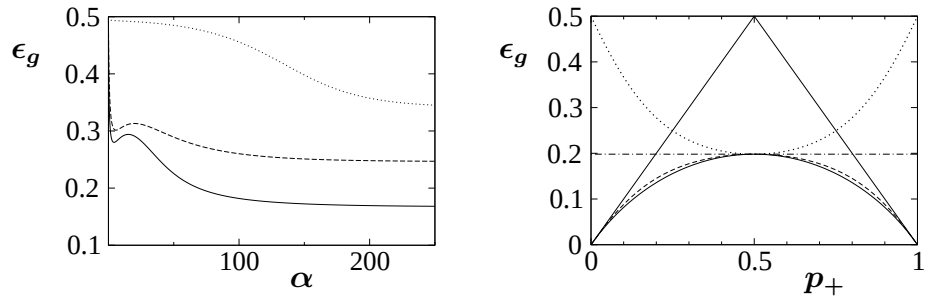

Figure 3: Left panel: Typical learning curves $\epsilon_g(\alpha)$ of unsupervised VQ (dotted), LVQ+ (dashed) and LVQ1 (solid line) for $\lambda = 1.0$ and learning rate $\eta = 0.2$.
Right panel: asymptotic ϵ_g for $\eta \to 0, \eta\alpha \to \infty$ for $\lambda = 1.2$ as a function of the prior weight p_+. The lowest, solid curve corresponds to the optimum ϵ_g^{min} whereas the dashed line represents the typical outcome of LVQ1. The horizontal line is the $p_\pm$–independent result for LVQ+, it can even exceed $\epsilon_g = \min\{p_+, p_-\}$ (thin solid line). Unsupervised VQ yields an asymptotic ϵ_g as marked by the chain line.

4 Results

4.1 The dynamics

The dynamics of unsupervised VQ, setting III in section 2.1, has been studied for $p_+ = p_-$ in an earlier publication [7]. Because data labels are disregarded or unavailable, the prototypes could be exchanged with no effect on the achieved quantization error (2). This permutation symmetry causes the initial learning dynamics to take place mainly in the subspace orthogonal to $\lambda(\boldsymbol{B}_+ - \boldsymbol{B}_-)$. It is reflected in a weakly repulsive fixed point of the ODE in which all $R_{S\sigma}$ are equal. Generically, the prototypes remain *unspecialized* up to rather large values of α, depending on the precise initial conditions. Without prior knowledge, of course, $R_{S\sigma}(0) \approx 0$ holds. This key feature, as discussed in [7], persists for $p_+ \neq p_-$.

While VQ does not aim at good generalization, we can still obtain $\epsilon_g(\alpha)$ from the prototype configuration, see Figure 3 (left panel) for an example. The very slow initial decrease relates to the above mentioned effect.

Figure 2 shows the evolution of the order parameters $\{Q_{ST}, R_{S\sigma}\}$ for an example set of model parameters in LVQ1 training, scenario I in sec. 2.1. Monte Carlo simulations of the system for $N = 200$ already display excellent agreement with the $N \to \infty$ theory.

In LVQ1, data and prototypes are labeled and, hence, specialization is enforced as soon as $\alpha > 0$. The corresponding ϵ_g displays a very fast initial decrease, cf. Figure 3. The non-monotonic intermediate behavior of $\epsilon_g(\alpha)$ is particularly pronounced for very different prior weights, e.g. $p_+ > p_-$, and for strongly overlapping clusters (small λ).

Qualitatively, the typical behavior of LVQ+ is similar to that of LVQ1. However, unless $p_+ = p_-$, the achieved $\epsilon_g(\alpha)$ is significantly larger as shown in Figure 3. This effect becomes clearer from the discussion of asymptotic configurations in the next section. The typical trajectories of prototype vectors in the space of order parameters will be discussed in greater detail for a variety of LVQ algorithms in forthcoming publications.

4.2 Asymptotic generalization

Apart from the behavior for small and intermediate values of α, we are also interested in the generalization ability that can be achieved, in principle, from an unlimited number of examples. This provides important means for judging the potential performance of training algorithms.

For stochastic gradient descent procedures like VQ, the expectation value of the associated cost function is minimized in the simultaneous limits of $\eta \to 0$ and $\alpha \to \infty$ such that $\tilde{\alpha} = \eta\alpha \to \infty$. In the absence of a cost function we can still consider this limit in which the system of ODE simplifies and can be expressed in the rescaled $\tilde{\alpha}$ after neglecting terms of order $\mathcal{O}\left(\eta^2\right)$. A fixed point analysis then yields well defined asymptotics, see [7] for a treatment of VQ. Figure 3 (right panel) summarizes our findings for the asymptotic ϵ_g^∞ as a function of p_+.

It is straightforward to work out the decision boundary with minimal generalization error ϵ_g^{min} in our model. For symmetry reasons it is a plane orthogonal to $(\boldsymbol{B}_+ - \boldsymbol{B}_-)$ and contains all ξ with $p_+ P(\xi|+1) = p_- P(\xi|-1)$ [5]. The lowest, solid line in Figure 3 (right panel) represents $\epsilon_g^{\min}$ for $\lambda = 1.2$ as a function of p_+. For comparison, the trivial classification according to the priors $p_\pm$ yields $\epsilon_g^{triv} = \min\{p_-, p_+\}$ and is also included in Figure 3.

In unsupervised VQ a strong prevalence, e.g. $p_+ \approx 1$, will be accounted for by placing both vectors inside the stronger cluster, thus achieving a low quantization error (2). Obviously this yields a poor classification as indicated by $\epsilon_g^\infty = 1/2$ in the limiting cases $p_+ = 0$ or 1. In the special case $p_+ = 1/2$ the aim of representation happens to coincide with good generalization and ϵ_g becomes optimal, indeed.

LVQ1, setting I in section 2.1, yields a classification scheme which is very close to being optimal for all values of p_+, cf. Figure 3 (right panel). On the

contrary, LVQ+ (algorithm II) updates each $\boldsymbol{w}_S$ only with data from class S. As a consequence, the asymptotic positions of the $\boldsymbol{w}_\pm$ is always symmetric about the geometrical center $\lambda(\boldsymbol{B}_+ + \boldsymbol{B}_-)$ and ϵ^∞ is independent of the priors $p_\pm$. Thus, LVQ+ is robust w.r.t. a variation of $p_\pm$ after training, i.e. it is optimal in the sense of the minmax-criterion $\sup_{p_\pm} \epsilon_g(\alpha)$ [5].

5 Conclusions

LVQ type learning models constitute popular learning algorithms due to their simple learning rule, their intuitive formulation of a classifier by means of prototypical locations in the data space, and their efficient applicability to any given number of classes. However, only very few theoretical results have been achieved so far which explain the behavior of such algorithms in mathematical terms, including large margin bounds as derived in [2, 8] and variations of LVQ type algorithms which can be derived from a cost function [9, 13–15]. In general, the often excellent generalization ability of LVQ type algorithms is not guaranteed by theoretical arguments, and the stability and convergence of various LVQ learning schemes is only poorly understood. Often, further heuristics such as the window rule for LVQ2.1 are introduced to overcome the problems of the original, heuristically motivated algorithms [10]. This has led to a large variety of (often only slightly) different LVQ type learning rules the drawbacks and benefits of which are hardly understood. Apart from an experimental benchmarking of these proposals, there is clearly a need for a thorough theoretical comparison of the behavior of the models to judge their efficiency and to select the most powerful learning schemes among these proposals.

We have investigated different Winner-Takes-All algorithms for Learning Vector Quantization in the framework of an analytically treatable model. The theory of on-line learning enables us to describe the dynamics of training in terms of differential equations for a few characteristic quantities. The formalism becomes exact in the limit $N \to \infty$ of very high dimensional data and many degrees of freedom.

This framework opens the way towards an exact investigation of situations where, so far, only heuristic evaluations have been possible, and it can serve as a uniform approach to investigate and compare LVQ type learning schemes based on a solid theoretical ground: The generalization ability can be evaluated also for heuristic training prescriptions which lack a corresponding cost function, including Kohonen's basic LVQ algorithm. Already in our simple model setting, this formal analysis shows pronounced

characteristics and differences of the typical learning behavior. On the one hand, the learning curves display common features such as an initial non-monotonic behavior. This behavior can be attributed to the necessity of symmetry breaking for the fully unsupervised case, and an overshooting effect due to different prior probabilities of the two classes, for supervised settings. On the other hand, slightly different intuitive learning algorithms yield fundamentally different asymptotic configurations as demonstrated for VQ, LVQ1, and LVQ+. These differences can be attributed to the different inherent goals of the learning schemes. In consequence, these differences are particularly pronounced for unbalanced class distributions where the goals of minimizing the quantization error, the minmax error, and the generalization error do not coincide. It is quite remarkable that the conceptually simple, original learning rule LVQ1 shows close to optimal generalization for the entire range of the prior weights $p_{\pm}$.

Despite the relative simplicity of our model situation, the training of only two prototypes from two class data captures many non-trivial features of more realistic settings: In terms of the itemization in section 2.1, situation I (LVQ1) describes the generic competition at class borders in basic LVQ schemes. Scenario II (LVQ+) corresponds to an intuitive variant which can be seen as a mixture of VQ and LVQ, i.e. Vector Quantization within the different classes. Finally, setting III (VQ) models the representation of one class by several competing prototypes.

Here, we have only considered WTA schemes for training. It is possible to extend the same analysis to more complex update rules such as LVQ2.1 or recent proposals [9, 14, 15] which update more than one prototype at a time. The treatment of LVQ type learning in the online scenario can easily be extended to algorithms which obey a more general learning rule than equation 1, e.g. learning rules which, by substituting the Heaviside term, adapt more than one prototype at a time. Our preliminary studies along this line show remarkable differences in the generalization ability and convergence properties of such variations of LVQ-type algorithms.

Clearly, our model cannot describe all features of more complex real world problems. Forthcoming investigations will concern, for instance, the extension to situations where the model complexity and the structure of the data do not match perfectly. This requires the treatment of scenarios with more than two prototypes and cluster centers. We also intend to study the influence of different variances within the classes and non-spherical clusters, aiming at a more realistic modeling. We are convinced that this line of research will lead to a deeper understanding of LVQ type training and will facilitate the development of novel efficient algorithms.

References

[1] *Bibliography on the Self-Organizing Map (SOM) and Learning Vector Quantization (LVQ)*. Neural Networks Research Centre, Helsinki University of Technology, 2002.

[2] K. Crammer, R. Gilad-Bachrach, A. Navot, and A. Tishby. Margin analysis of the LVQ algorithm. In: *Advances in Neural Information Processing Systems*, 2002.

[3] M. Biehl and N. Caticha, *The statistical physics of learning and generalization*. In: M. Arbib (ed.), *Handbook of brain theory and neural networks*, second edition, MIT Press, 2003.

[4] M. Biehl, A. Freking, A. Ghosh, and G. Reents, *A theoretical framework for analysing the dynamics of Learning Vector Quantization*. Technical Report 2004-9-02, Institute of Mathematics and Computing Science, University Groningen, available from `www.cs.rug.nl/ biehl`, 2004.

[5] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley, 2001.

[6] A. Engel and C. van den Broeck, *The Statistical Mechanics of Learning*, Cambridge University Press, 2001.

[7] A. Freking, G. Reents, and M. Biehl, *The dynamics of competitive learning*, Europhysics Letters 38 (1996) 73-78.

[8] B. Hammer, M. Strickert and T. Villmann. *On the generalization capability of GRLVQ networks*, Neural Processing Letters 21 (2005) 109-120.

[9] B. Hammer and T. Villmann. *Generalized relevance learning vector quantization*, Neural Networks 15 (2002) 1059-1068.

[10] T. Kohonen. *Self-organizing maps*, Springer, Berlin, 1995.

[11] G. Reents and R. Urbanczik. *Self-averaging and on-line learning*, Physical Review Letters 80 (1998) 5445-5448.

[12] D. Saad, editor. *Online learning in neural networks*, Cambridge University Press, 1998.

[13] S. Sato and K. Yamada. Generalized learning vector quantization. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural*

Information Processing Systems, volume 7, pages 423–429, MIT Press, 1995.

[14] S. Seo, M. Bode, and K. Obermayer. *Soft nearest prototype classification*, IEEE Transactions on Neural Networks 14 (2003) 390-398.

[15] S. Seo and K. Obermayer. *Soft Learning Vector Quantization*. Neural Computation 15 (2003) 1589-1604.

[16] P. Somervuo and T. Kohonen. *Self-organizing maps and learning vector quantization for feature sequences*, Neural Processing Letters 10 (1999) 151-159.

[17] T.L.H. Watkin, A. Rau, and M. Biehl, *The statistical mechanics of learning a rule*, Reviews of Modern Physics 65 (1993) 499-556.

A The theoretical framework

In this appendix we outline key steps of the calculations referred to in the text. The formalism was first used in the context of unsupervised Vector Quantization [7] and the calculations were recently detailed in a Technical Report [4].

Throughout this appendix indices $l, m, k, s, \sigma \in \{\pm 1\}$ (or $\pm$ for short) represent the class labels and cluster memberships.

Note that the following corresponds to a slightly more general input distribution with variances v_σ of the Gaussian clusters $\sigma = \pm 1$

$$P(\boldsymbol{\xi}) = \sum_{\sigma=\pm 1} p_\sigma P(\boldsymbol{\xi}\,|\,\sigma) \quad \text{with} \quad P(\boldsymbol{\xi}\,|\,\sigma) = \frac{1}{\sqrt{2\pi v_\sigma}^N} \exp\left[-\frac{1}{2\,v_\sigma}\,(\boldsymbol{\xi} - \lambda \boldsymbol{B}_\sigma)^2\right]. \tag{9}$$

All results in the main text refer to the special choice $v_+ = v_- = 1$ which corresponds to Eq. (3). The influence of choosing different variances on the behavior of LVQ type training will be studied in forthcoming publications.

A.1 Statistics of the projections

To a large extent, our analysis is based on the observation that the projections $h_\pm = \boldsymbol{w}_\pm \cdot \boldsymbol{\xi}$ and $b_\pm = \boldsymbol{B}_\pm \cdot \boldsymbol{\xi}$ are correlated Gaussian random quantities for a vector $\boldsymbol{\xi}$ drawn from one of the clusters contributing to the density (9). Where convenient, we will combine the projections into a four-dimensional vector denoted as $\underline{x} = (h_+, h_-, b_+, b_-)^T$.

We will assume implicitly that $\boldsymbol{\xi}$ is statistically independent from the considered weight vectors $\boldsymbol{w}_\pm$. This is obviously the case in our on-line prescription where the novel example $\boldsymbol{\xi}^\mu$ is uncorrelated with all previous data and hence with $\boldsymbol{w}_\pm^{\mu-1}$.

The first and second conditional moments given in Eq. (7) are obtained from the following elementary considerations.

First moments
Exploiting the above mentioned statistical independence we can show immediately that

$$\langle h_l \rangle_k = \langle \boldsymbol{w}_l \cdot \boldsymbol{\xi} \rangle_k = \boldsymbol{w}_l \cdot \langle \boldsymbol{\xi} \rangle_k = \boldsymbol{w}_l \cdot (\lambda\, \boldsymbol{B}_k) = \lambda\, R_{lk}. \tag{10}$$

Similarly we get for b_l:

$$\langle b_l \rangle_k = \langle \boldsymbol{B}_l \cdot \boldsymbol{\xi} \rangle_k = \boldsymbol{B}_l \cdot \langle \boldsymbol{\xi} \rangle_k = \boldsymbol{B}_l \cdot (\lambda\, \boldsymbol{B}_k) = \lambda\, \delta_{lk}, \tag{11}$$

where δ_{lk} is the Kronecker delta and we exploit that $\boldsymbol{B}_+$ and $\boldsymbol{B}_-$ are orthonormal. Now the conditional means $\underline{\mu}_k = \langle \underline{x} \rangle_k$ can be written as

$$\underline{\mu}_{k=+1} = \begin{pmatrix} \lambda R_{++} \\ \lambda R_{-+} \\ \lambda \\ 0 \end{pmatrix} \qquad \text{and} \qquad \underline{\mu}_{k=-1} = \begin{pmatrix} \lambda R_{+-} \\ \lambda R_{--} \\ 0 \\ \lambda \end{pmatrix} \tag{12}$$

Second moments

In order to compute the conditional variance or covariance $\langle h_l h_m \rangle_k - \langle h_l \rangle_k \langle h_m \rangle_k$ we first consider the average

$$\begin{aligned}
\langle h_l \, h_m \rangle_k &= \langle (\boldsymbol{w}_l \cdot \boldsymbol{\xi})(\boldsymbol{w}_m \cdot \boldsymbol{\xi}) \rangle_k = \left\langle \left(\sum_{i=1}^N (\boldsymbol{w}_l)_i (\boldsymbol{\xi})_i \right) \left(\sum_{j=1}^N (\boldsymbol{w}_m)_j (\boldsymbol{\xi})_j \right) \right\rangle_k \\
&= \left\langle \sum_{i=1}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_i (\boldsymbol{\xi})_i (\boldsymbol{\xi})_i + \sum_{i=1}^N \sum_{j=1, j\neq i}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_j (\boldsymbol{\xi})_i (\boldsymbol{\xi})_j \right\rangle_k \\
&= \sum_{i=1}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_i \langle (\boldsymbol{\xi})_i (\boldsymbol{\xi})_i \rangle_k + \sum_{i=1}^N \sum_{j=1, j\neq i}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_j \langle (\boldsymbol{\xi})_i (\boldsymbol{\xi})_j \rangle_k \\
&= \sum_{i=1}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_i \big[v_k + \lambda^2 (\boldsymbol{B}_k)_i (\boldsymbol{B}_k)_i \big] \\
&\quad + \sum_{i=1}^N \sum_{j=1, j\neq i}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_j \lambda^2 (\boldsymbol{B}_k)_i (\boldsymbol{B}_k)_j \\
&= v_k \sum_{i=1}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_i + \lambda^2 \sum_{i=1}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_i (\boldsymbol{B}_k)_i (\boldsymbol{B}_k)_i \\
&\quad + \lambda^2 \sum_{i=1}^N \sum_{j=1, j\neq i}^N (\boldsymbol{w}_l)_i (\boldsymbol{w}_m)_j (\boldsymbol{B}_k)_i (\boldsymbol{B}_k)_j \\
&= v_k \boldsymbol{w}_l \cdot \boldsymbol{w}_m + \lambda^2 (\boldsymbol{w}_l \cdot \boldsymbol{B}_k)(\boldsymbol{w}_m \cdot \boldsymbol{B}_k) \; = \; v_k \, Q_{lm} + \lambda^2 \, R_{lk} \, R_{mk}
\end{aligned}$$

Here we have used once more that components of $\boldsymbol{\xi}$ from cluster k have variance v_k and are independent. This implies for all $i, j \in \{1, \ldots, N\}$

$$\langle (\boldsymbol{\xi})_i \, (\boldsymbol{\xi})_i \rangle_k - \langle (\boldsymbol{\xi})_i \rangle_k \, \langle (\boldsymbol{\xi})_i \rangle_k \; = \; v_k \quad \Rightarrow \quad \langle (\boldsymbol{\xi})_i \, (\boldsymbol{\xi})_i \rangle_k = v_k + \langle (\boldsymbol{\xi})_i \rangle_k \, \langle (\boldsymbol{\xi})_i \rangle_k \, ,$$

and $\langle (\boldsymbol{\xi})_i \, (\boldsymbol{\xi})_j \rangle_k \; = \; \langle (\boldsymbol{\xi})_i \rangle_k \, \langle (\boldsymbol{\xi})_j \rangle_k \, . \quad \text{for } i \neq j.$

Finally, we obtain the conditional second moment

$$\langle h_l\, h_m\rangle_k - \langle h_l\rangle_k\langle h_m\rangle_k \;=\; v_k\, Q_{lm} + \lambda^2 R_{lk}R_{mk} - \lambda^2 R_{lk}R_{mk} \;=\; v_k\, Q_{lm} \quad (13)$$

Similarly, we find for the quantities b_+, b_-:

$$\langle b_l b_m\rangle_k - \langle b_l\rangle_k\langle b_m\rangle_k = v_k\delta_{lm} + \lambda^2\delta_{lk}\delta_{mk} - \lambda^2\delta_{lk}\delta_{mk} = v_k\,\delta_{lm} \quad (14)$$

Eventually, we evaluate the covariance $\langle h_l b_m\rangle_k - \langle h_l\rangle_k\langle b_m\rangle_k$ by considering the average

$$\begin{aligned}\langle h_l b_m\rangle_k &= \langle(\boldsymbol{w}_l\cdot\boldsymbol{\xi})(\boldsymbol{B}_m\cdot\boldsymbol{\xi})\rangle_k \;=\; v_k\,\boldsymbol{w}_l\cdot\boldsymbol{B}_m + \lambda^2(\boldsymbol{w}_l\cdot\boldsymbol{B}_k)(\boldsymbol{B}_m\cdot\boldsymbol{B}_k)\\ &= v_k\,R_{lm} + \lambda^2\,R_{lk}\,\delta_{mk}\end{aligned}$$

and obtain

$$\langle h_l, b_m\rangle_k - \langle h_l\rangle_k\langle b_m\rangle_k \;=\; v_k\,R_{lm} + \lambda^2 R_{lk}\,\delta_{mk} - \lambda^2\,R_{lk}\,\delta_{mk} \;=\; v_k\,R_{lm}. \quad (15)$$

The above results are summarized in Eq. (7). The conditional covariance matrix of $\underline{x}$ can be expressed explicitly in terms of the order parameters as follows:

$$C_k = v_k \begin{pmatrix} Q_{++} & Q_{+-} & R_{++} & R_{+-} \\ Q_{+-} & Q_{--} & R_{-+} & R_{--} \\ R_{++} & R_{-+} & 1 & 0 \\ R_{+-} & R_{--} & 0 & 1 \end{pmatrix} \quad (16)$$

The conditional density of $\underline{x}$ for data from class k is a Gaussian $N(\underline{\mu}_k, C_k)$ where $\underline{\mu}_k$ is the conditional mean vector, Eq. (12), and C_k is given above.

A.2 Differential Equations

For the training prescriptions considered here it is possible to use a particularly compact notation. The function $g(l,\sigma^\mu)$ as introduced in Eq. (1) can be written as

$$g(l,\sigma) \;=\; a \;+\; b\; l\; \sigma$$

where $a, b \in \mathbb{R}$ and $l, \sigma \in \{+1, -1\}$. The WTA schemes listed in section 2.1 are recovered as follows

I) $a = 0, b = 1$: LVQ1 with $g(l,\sigma) = l\,\sigma = \pm 1$

II) $a = b = 1/2$: LVQ+ with $g(l,\sigma) = \Theta(l\,\sigma) = \begin{cases} 1 & \text{if } l = \sigma \\ 0 & \text{else} \end{cases}$

III) $a = 1, b = 0$: VQ with $g(l,\sigma) = 1$.

The recursion relations, Eq. (5), are to be averaged over the density of a new, independent input $\boldsymbol{\xi}$ drawn from the density (9). In the limit $N \to \infty$ this yields a system of coupled differential equations in continuous learning time $\alpha = P/N$ as argued in the text. Exploiting the notations defined above, it can be formally expressed in the following way:

$$\begin{aligned}
\frac{dR_{lm}}{d\alpha} &= \eta\Big[a\Big(\langle b_m\Theta_l\rangle - \langle\Theta_l\rangle R_{lm}\Big) + bl\Big(\langle\sigma b_m\Theta_l\rangle - \langle\sigma\Theta_l\rangle R_{lm}\Big)\Big] \\
\frac{dQ_{lm}}{d\alpha} &= \eta\Big[b\Big(l\langle\sigma h_m\Theta_l\rangle - l\langle\sigma\Theta_l\rangle Q_{lm} + m\langle\sigma h_l\Theta_m\rangle \\
&\quad -m\langle\sigma\Theta_m\rangle Q_{lm}\Big) + a\Big(\langle h_m\Theta_l\rangle - \langle\Theta_l\rangle Q_{lm} \\
&\quad +\langle h_l\Theta_m\rangle - \langle\Theta_m\rangle Q_{lm}\Big) + \delta_{lm}\eta\big[a^2 + b^2 lm\big] \\
&\quad \left(\sum_{\sigma=\pm1} v_\sigma p_\sigma\langle\Theta_l\rangle_\sigma\right) + \delta_{lm}ab(l+m)\eta\left(\sum_{\sigma=\pm1} v_\sigma p_\sigma\langle\Theta_l\rangle_\sigma\right)\Big]
\end{aligned} \tag{17}$$

where $\langle\cdots\rangle = \sum_{k=\pm1}\langle\cdots\rangle_k$. Note that the equations for Q_{lm} contain terms of order η^2 whereas the ODE for R_{lm} are linear in the learning rate.

A.2.1 Averages

We introduce the vectors and auxiliary quantities

$$\underline{\alpha}_l = (+2l, -2l, 0, 0) \quad \text{and} \quad \beta_l = l(Q_{+l+l} - Q_{-l-l}) \tag{18}$$

which allow us to rewrite the Heaviside terms as

$$\Theta_l = \Theta\left(Q_{-l-l} - 2h_{-l} - Q_{+l+l} + 2h_{+l}\right) = \Theta\big(\underline{\alpha}_l \cdot \underline{x} - \beta_l\big) \tag{19}$$

Performing the averages in (17) involves conditional means of the form

$$\langle(\underline{x})_n\Theta_s\rangle_k \quad \text{and} \quad \langle\Theta_s\rangle_k$$

where $(\underline{x})_n$ is the n^{th} component of vector $\underline{x} = (h_{+1}, h_{-1}, b_{+1}, b_{-1})$. We first address the term

$$\begin{aligned}
\langle(\underline{x})_n\Theta_s\rangle_k &= \frac{1}{(2\pi)^{4/2}\left(det(C_k)\right)^{\frac{1}{2}}}\int_{\mathbb{R}^4}(x)_n\Theta\left(\underline{\alpha}_s\cdot\underline{x} - \beta_s\right) \\
&\quad \exp\left(-\frac{1}{2}\left(\underline{x} - \underline{\mu}_k\right)^T C_k^{-1}\left(\underline{x} - \underline{\mu}_k\right)\right)d\underline{x}
\end{aligned}$$

$$= \frac{1}{(2\pi)^{4/2}\,(det(C_k))^{\frac{1}{2}}} \int_{\mathbb{R}^4} \left(x^{'} + \underline{\mu}_k\right)_n \Theta\left(\underline{\alpha}_s \cdot \underline{x}^{'} + \underline{\alpha}_s \cdot \underline{\mu}_k - \beta_s\right)$$
$$\times \exp\left(-\frac{1}{2}\underline{x}^{'T} C_k^{-1} \underline{x}^{'}\right) d\underline{x}^{'} \qquad \text{with the substitution } \underline{x}^{'} = \underline{x} - \underline{\mu}_k$$

Let $\underline{x}^{'} = C_k{}^{\frac{1}{2}}\underline{y}$, where $C_k{}^{\frac{1}{2}}$ is defined in the following way: $C_k = C_k{}^{\frac{1}{2}} C_k{}^{\frac{1}{2}}$. Since C_k is a covariance matrix, it is positive semidefinite and $C_k{}^{\frac{1}{2}}$ exists. Hence we have $d\underline{x}^{'} = det(C_k^{\frac{1}{2}}) d\underline{y} = (det(C_k))^{\frac{1}{2}} d\underline{y}$ and

$$\begin{aligned}
\langle (\underline{x})_n\, \Theta_s \rangle_k &= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^4} (C_k^{\frac{1}{2}}\underline{y})_n \Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}} \underline{y} + \underline{\alpha}_s \cdot \underline{\mu}_k - \beta_s\right) \\
&\quad \exp\left(-\frac{1}{2}\underline{y}^2\right) d\underline{y} + (\underline{\mu}_k)_n \langle \Theta_s \rangle_k \\
&= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^4} \sum_{j=1}^{4} \left((C_k^{\frac{1}{2}})_{nj} (\underline{y})_j\right) \Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}} \underline{y} + \underline{\alpha}_s \cdot \underline{\mu}_k - \beta_s\right) \\
&\quad \exp\left(-\frac{1}{2}\underline{y}^2\right) d\underline{y} + (\underline{\mu}_k)_n \langle \Theta_s \rangle_k \\
&= I + (\underline{\mu}_k)_n \langle \Theta_s \rangle_k \qquad \text{(with the abbreviation } I\text{)} \qquad (20)
\end{aligned}$$

Consider the integrals contributing to I:

$$I_j = \int_{\mathbb{R}} (C_k^{\frac{1}{2}})_{nj} (\underline{y})_j \Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}} \underline{y} + \underline{\alpha}_s \cdot \underline{\mu}_k - \beta_s\right) \exp\left(-\frac{1}{2}(\underline{y})_j^2\right) d(\underline{y})_j.$$

We can perform an integration by parts, $\int u dv = uv - \int v du$, with

$u = \Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}} \underline{y} + \underline{\alpha}_s \cdot \underline{\mu}_k - \beta_s\right), \qquad v = (C_k^{\frac{1}{2}})_{nj} \exp\left(-\frac{1}{2}(\underline{y})_j^2\right)$

$du = \dfrac{\partial}{\partial(\underline{y})_j} \Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}} \underline{y} + \underline{\alpha}_s \cdot \underline{\mu}_k - \beta_s\right) d(\underline{y})_j$

$dv = (-)(C_k^{\frac{1}{2}})_{nj} (\underline{y})_j \exp\left(-\dfrac{1}{2}(\underline{y})_j^2\right) d(\underline{y})_j$, and obtain

$$I_j = \underbrace{-\left[\Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}} \underline{y} + \underline{\alpha}_s \cdot \underline{\mu}_k - \beta_s\right) (C_k^{\frac{1}{2}})_{nj} \exp\left(-\frac{1}{2}(\underline{y})_j^2\right)\right]_{-\infty}^{\infty}}_{0}$$

$$
\begin{aligned}
&+\left[\int_{\mathbb{R}}(C_k^{\frac{1}{2}})_{nj}\frac{\partial}{\partial(\underline{y})_j}\left(\Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}}\underline{y}+\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s\right)\right)\right.\\
&\left.\exp\left(-\frac{1}{2}(\underline{y})_j^2\right)d(\underline{y})_j\right]\\
=\;&\int_{\mathbb{R}}(C_k^{\frac{1}{2}})_{nj}\frac{\partial}{\partial(\underline{y})_j}\left(\Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}}\underline{y}+\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s\right)\right)\\
&\exp\left(-\frac{1}{2}(\underline{y})_j^2\right)d(\underline{y})_j \qquad (21)
\end{aligned}
$$

The sum over j gives

$$
\begin{aligned}
I \;=\;&\frac{1}{(2\pi)^2}\sum_{j=1}^{4}(C_k^{\frac{1}{2}})_{nj}\int_{\mathbb{R}^4}\frac{\partial}{\partial(\underline{y})_j}\left(\Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}}\underline{y}+\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s\right)\right)\\
&\exp\left(-\frac{1}{2}\underline{y}^2\right)d\underline{y}\\
=\;&\frac{1}{(2\pi)^2}\sum_{j=1}^{4}\left((C_k^{\frac{1}{2}})_{nj}\sum_{i=1}^{4}(\underline{\alpha}_s)_i(C_k^{\frac{1}{2}})_{i,j}\right)\\
&\int_{\mathbb{R}^4}\left(\delta\left(\underline{\alpha}_s C_k^{\frac{1}{2}}\underline{y}+\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s\right)\right)\exp\left(-\frac{1}{2}\underline{y}^2\right)d\underline{y}.\\
=\;&\frac{1}{(2\pi)^2}(C_k\underline{\alpha}_s)_n\int_{\mathbb{R}^4}\left(\delta\left(\underline{\alpha}_s C_k^{\frac{1}{2}}\underline{y}+\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s\right)\right)\\
&\exp\left(-\frac{1}{2}\underline{y}^2\right)d\underline{y}. \qquad (22)
\end{aligned}
$$

In the last step we have used

$$
\frac{\partial}{\partial(\underline{y})_j}\Theta\left(\underline{\alpha}_s C_k^{\frac{1}{2}}\underline{y}+\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s\right)=\sum_{i=1}^{4}(\underline{\alpha}_s)_i(C_k^{\frac{1}{2}})_{i,j}\delta(\underline{\alpha}_s C_k^{\frac{1}{2}}\underline{y}+\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s)
$$

where $\delta(.)$ is the Dirac-delta function.

Now, note that $\exp\left[-\frac{1}{2}\underline{y}^2\right]d\underline{y}$ is a measure which is invariant under rotation of the coordinate axes. We rotate the system in such a way that one of the axes, say $\tilde{y}$, is aligned with the vector $C_k^{\frac{1}{2}}\underline{\alpha}_s$. The remaining three coordinates can be integrated over and we get

$$
I=\frac{1}{\sqrt{2\pi}}(C_k\underline{\alpha}_s)_n\int_{\mathbb{R}}\delta\left(\|C_k^{\frac{1}{2}}\underline{\alpha}_s\|\tilde{y}+\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s\right)\exp\left[-\frac{1}{2}\tilde{y}^2\right]d\tilde{y} \qquad (23)
$$

We define

$$
\tilde{\alpha}_{sk}\;=\;\|C_k^{\frac{1}{2}}\underline{\alpha}_s\|=\sqrt{\underline{\alpha}_s C_k\underline{\alpha}_s}\quad\text{and}\quad\tilde{\beta}_{sk}\;=\;\underline{\alpha}_s\cdot\underline{\mu}_k-\beta_s \qquad (24)
$$

and obtain

$$\begin{aligned} I &= \frac{1}{\sqrt{2\pi}}(C_k\underline{\alpha}_s)_n \int_{\mathbb{R}} \delta\left(\tilde{\alpha}_{sk}\tilde{y}+\tilde{\beta}_{sk}\right)\exp\left[-\frac{1}{2}\tilde{y}^2\right]d\tilde{y} \\ &= \frac{(C_k\underline{\alpha}_s)_n}{\sqrt{2\pi}\tilde{\alpha}_{sk}} \int_{\mathbb{R}} \delta\left(z+\tilde{\beta}_{sk}\right)\exp\left[-\frac{1}{2}(\frac{z}{\tilde{\alpha}_{sk}})^2\right]dz \quad (\text{with} \quad z=\alpha_{sk}\tilde{y}) \\ &= \frac{(C_k\underline{\alpha}_s)_n}{\sqrt{2\pi}\tilde{\alpha}_{sk}}\exp\left[-\frac{1}{2}(\frac{\tilde{\beta}_{sk}}{\tilde{\alpha}_{sk}})^2\right] \end{aligned} \tag{25}$$

Now we compute the remaining average in (20) in an analogous way and get

$$\begin{aligned} \langle\Theta_s\rangle_k &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\Theta\left(\tilde{\alpha}_{sk}\tilde{y}+\tilde{\beta}_{sk}\right)\exp\left[-\frac{1}{2}\tilde{y}^2\right]d\tilde{y} \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\frac{\tilde{\beta}_{sk}}{\tilde{\alpha}_{sk}}}\exp\left[-\frac{1}{2}\tilde{y}^2\right]d\tilde{y} = \Phi\left(\frac{\tilde{\beta}_{sk}}{\tilde{\alpha}_{sk}}\right), \end{aligned} \tag{26}$$

where $\Phi(x)=\frac{1}{2\pi}\int_{-\infty}^{x}e^{-\frac{z^2}{2}}dz$.

Finally we obtain the required average using (25) and (26) as follows:

$$\langle(\underline{x})_n\Theta_s\rangle_k = \frac{(C_k\underline{\alpha}_s)_n}{\sqrt{2\pi}\tilde{\alpha}_{sk}}\exp\left[-\frac{1}{2}(\frac{\tilde{\beta}_{sk}}{\tilde{\alpha}_{sk}})^2\right]+(\underline{\mu}_k)_n\Phi\left(\frac{\tilde{\beta}_{sk}}{\tilde{\alpha}_{sk}}\right). \tag{27}$$

The quantities $\tilde{\alpha}_{sk}$ and $\tilde{\beta}_{sk}$ are defined through Eqs. (24) and (18).

A.2.2 Final form of the differential equations

Using (26) and (27) the system differential equations reads

$$\begin{aligned} \frac{dR_{lm}}{d\alpha} = \eta\Bigg[(bl)\Big(\sum_{\sigma=\pm1}\sigma p_\sigma\Big[\frac{(C\underline{\alpha}_l)_{n_{bm}}}{\sqrt{2\pi}\tilde{\alpha}_{l\sigma}}\exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}})^2\big] \\ +(\underline{\mu}_\sigma)_{n_{bm}}\Phi\big(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}\big)\Big]-\sum_{\sigma=\pm1}\sigma p_\sigma\Phi\big(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}\big)R_{lm}\Big) \end{aligned}$$

$$+ a \Big(\sum_{\sigma=\pm 1} p_\sigma \Big[\frac{(C\underline{\alpha}_l)_{n_{bm}}}{\sqrt{2\pi}\tilde{\alpha}_{l\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}})^2 \big] + (\underline{\mu}_\sigma)_{n_{bm}} \Phi(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}) \Big]$$

$$- \sum_{\sigma=\pm 1} p_\sigma \Phi(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}) R_{lm} \Big) \Big] \tag{28}$$

$$\begin{aligned}
\frac{dQ_{lm}}{d\alpha} \;=\; & \eta \Bigg(bl \sum_{\sigma=\pm 1} \sigma p_\sigma \Big[\frac{(C\underline{\alpha}_l)_{n_{hm}}}{\sqrt{2\pi}\tilde{\alpha}_{l\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}})^2 \big] + (\underline{\mu}_\sigma)_{n_{hm}} \Phi(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}) \Big] \\
& -bl \sum_{\sigma=\pm 1} \sigma p_\sigma \big[\Phi(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}) Q_{lm} + bm \sum_{\sigma=\pm 1} \sigma p_\sigma \Big[\frac{(C\underline{\alpha}_l)_{n_{hl}}}{\sqrt{2\pi}\tilde{\alpha}_{m\sigma}} \\
& \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{m\sigma}}{\tilde{\alpha}_{m\sigma}})^2 \big] + (\underline{\mu}_\sigma)_{n_{hl}} \Phi(\frac{\tilde{\beta}_{m\sigma}}{\tilde{\alpha}_{m\sigma}}) \Big] - bm \sum_{\sigma=\pm 1} \sigma p_\sigma \Phi(\frac{\tilde{\beta}_{m\sigma}}{\tilde{\alpha}_{m\sigma}}) \\
& Q_{lm} \Bigg) + \delta_{lm}(a^2 + b^2 lm)\eta^2 \sum_{\sigma=\pm 1} \sigma v_\sigma p_\sigma \Phi(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}) + \delta_{lm}\eta^2 \big(ab(l+m) \big) \\
& \sum_{\sigma=\pm 1} \sigma v_\sigma p_\sigma \Phi(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}) + \eta \Bigg(a \sum_{\sigma=\pm 1} p_\sigma \Big[\frac{(C\underline{\alpha}_l)_{n_{hm}}}{\sqrt{2\pi}\tilde{\alpha}_{l\sigma}} \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}})^2 \big] \\
& + (\underline{\mu}_\sigma)_{n_{hm}} \Phi(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}) \Big] - a \sum_{\sigma=\pm 1} p_\sigma \Phi(\frac{\tilde{\beta}_{l\sigma}}{\tilde{\alpha}_{l\sigma}}) Q_{lm} + a \sum_{\sigma=\pm 1} p_\sigma \Big[\frac{(C\underline{\alpha}_l)_{n_{hl}}}{\sqrt{2\pi}\tilde{\alpha}_{m\sigma}} \\
& \exp\big[-\frac{1}{2}(\frac{\tilde{\beta}_{m\sigma}}{\tilde{\alpha}_{m\sigma}})^2 \big] + (\underline{\mu}_\sigma)_{n_{hl}} \Phi(\frac{\tilde{\beta}_{m\sigma}}{\tilde{\alpha}_{m\sigma}}) \Big] - a \sum_{\sigma=\pm 1} p_\sigma \Phi(\frac{\tilde{\beta}_{m\sigma}}{\tilde{\alpha}_{m\sigma}}) Q_{lm} \Bigg).
\end{aligned} \tag{29}$$

Here $n_{bm} = \begin{array}{ll} 3 & \text{if } m = 1 \\ 4 & \text{if } m = -1 \end{array}$ and $n_{hm} = \begin{array}{ll} 1 & \text{if } m = 1 \\ 2 & \text{if } m = -1. \end{array}$

A.3 The generalization error

Using (26) we can easily compute the generalization error as follows:

$$\epsilon_g \;=\; \sum_{k=\pm 1} p_{-k} \langle \Theta_k \rangle_{-k} \;=\; \sum_{k=\pm 1} p_{-k} \Phi\Big(\frac{\tilde{\beta}_{k,-k}}{\tilde{\alpha}_{k,-k}} \Big) \tag{30}$$

which amounts to Eq. 8 in the text after inserting $\tilde{\beta}_{k,-k}$ and $\tilde{\alpha}_{k,-k}$ as given above with $v_+ = v_- = 1$.