Performance analysis of LVQ algorithms

Ghosh, Anarta; Biehl, Michael; Hammer, Barbara

Published in:
Neural Networks

DOI:
10.1016/j.neunet.2006.05.010

Publication date:
2006

Available online at www.sciencedirect.com
SCIENCE DIRECT
Neural Networks 19 (2006) 817–829
Neural Networks
www.elsevier.com/locate/neunet

2006 Special Issue

Performance analysis of LVQ algorithms: A statistical physics approach

Anarta Ghosh[a,*], Michael Biehl[a], Barbara Hammer[b]

[a] *Rijksuniversiteit Groningen, Mathematics and Computing Science, P.O. Box 800, NL-9700 AV Groningen, The Netherlands*
[b] *Clausthal University of Technology, Institute of Computer Science, D-98678 Clausthal-Zellerfeld, Germany*

Abstract

Learning vector quantization (LVQ) constitutes a powerful and intuitive method for adaptive nearest prototype classification. However, original LVQ has been introduced based on heuristics and numerous modifications exist to achieve better convergence and stability. Recently, a mathematical foundation by means of a cost function has been proposed which, as a limiting case, yields a learning rule similar to classical LVQ2.1. It also motivates a modification which shows better stability. However, the exact dynamics as well as the generalization ability of many LVQ algorithms have not been thoroughly investigated so far. Using concepts from statistical physics and the theory of on-line learning, we present a mathematical framework to analyse the performance of different LVQ algorithms in a typical scenario in terms of their dynamics, sensitivity to initial conditions, and generalization ability. Significant differences in the algorithmic stability and generalization ability can be found already for slightly different variants of LVQ. We study five LVQ algorithms in detail: Kohonen's original LVQ1, unsupervised vector quantization (VQ), a mixture of VQ and LVQ, LVQ2.1, and a variant of LVQ which is based on a cost function. Surprisingly, basic LVQ1 shows very good performance in terms of stability, asymptotic generalization ability, and robustness to initializations and model parameters which, in many cases, is superior to recent alternative proposals.

Keywords: Online learning; LVQ1; LVQ2.1; LVQ+; VQ; LFM; Thermodynamic limit; Order parameters

1. Introduction

Due to its simplicity, flexibility, and efficiency, learning vector quantization (LVQ) as introduced by Kohonen has been widely used in a variety of areas, including real time applications like speech recognition (Kohonen, 1995; McDermott & Katagiri, 1994; Neural Networks Research Centre, 0000). Several modifications of basic LVQ have been proposed which aim at a larger flexibility, faster convergence, more flexible metrics, or better adaptation to Bayesian decision boundaries, etc. (Duda, Hart, & Stork, 2000; Hammer & Villmann, 2002; Kohonen, 1995). Most learning schemes including basic LVQ have been proposed on heuristic grounds and their dynamics is not clear. In particular, there exist potentially powerful extensions like LVQ2.1 which require the introduction of additional heuristics, e.g. the so-called window rule for stability, which are not well understood theoretically.

Recently, several approaches relate LVQ-type learning schemes to exact mathematical concepts and thus open the way towards a solid mathematical justification for LVQ-type learning algorithms. The directions are mainly twofold. On one hand, cost functions have been proposed which, possibly as a limiting case, lead to LVQ-type gradient schemes such as the approaches in Hammer and Villmann (2002), Seo, Bode, and Obermayer (2003) and Seo and Obermayer (2003). Thereby, the nature of the cost function has consequences on the stability of the algorithm as pointed out in Sato and Yamada (1995) and Seo and Obermayer (2003); in addition, it allows a principled extension of LVQ-type learning schemes to complex situations introducing e.g. neighborhood cooperation of the prototypes or adaptive kernels into the classifier (Hammer, Strickert, & Villmann, 2005b). On the other hand, generalization bounds of the algorithms have been derived by means of statistical learning theory, as obtained in Crammer, Gilad-Bachrach, Navot, and Tishby (2002) and Hammer, Strickert, and Villmann (2005a), which characterize the generalization capability of LVQ and variants thereof in terms of the hypothesis margin of the classifier. Interestingly, the cost function of some extensions

* Corresponding address: Institute of Mathematics and Computing Science, 9747AC Groningen, The Netherlands.
E-mail address: anarta@cs.rug.nl (A. Ghosh).

0893-6080/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2006.05.010

of LVQ includes a term which measures the structural risk, thus aiming at margin maximization during training similar to support vector machines (Hammer et al., 2005a). However, the exact connection of the classification accuracy and the cost function is not clear for these proposals. In addition, the relation between the learning scheme and the generalization ability has not been investigated in mathematical terms so far. Furthermore, the formulation via a cost function is often limited to approximate scenarios which reach the exact crisp LVQ-type learning scheme only as a limiting case. Thus, there is a need for a systematic investigation of these methods in terms of their generalization ability, dynamics, and sensitivity to initial conditions.

It has been shown in Cottrell, Fort, and Pages (1998) and Fort and Pages (1996) for unsupervised prototype-based methods that a rigorous mathematical analysis of statistical learning algorithms is possible in some cases and might yield quite unexpected results. Recently, first results for supervised learning schemes such as LVQ have been presented in the article (Biehl, Ghosh, & Hammer, 2006); however, they are restricted to methods which adapt only one prototype at a time, excluding methods like LVQ2.1 and variants thereof. In this work we extend this study to a representative collection of popular LVQ-type learning schemes, using concepts from statistical physics. We investigate the learning dynamics, generalization ability, as well as sensitivity to model parameters and initialization.

The dynamics of training is studied along the successful theory of on-line learning (Biehl & Caticha, 2003; Engel & van den Broeck, 2001; Saad, 1998), considering learning from a sequence of uncorrelated, random training data generated according to a model distribution unknown to the training scheme and the limit $N \to \infty$, N being the data dimensionality. In this limit, the system dynamics can be described by coupled ordinary differential equations in terms of characteristic quantities, the solutions of which provide insight into the learning dynamics and interesting quantities such as the generalization error.

Here the investigation and comparison of algorithms are done with respect to the typical behavior of large systems in the framework of a model situation. The approach presented in this paper complements other paradigms which provide rigorously exact results without making explicit assumptions about, for instance, the statistics of the data, see e.g. Cottrell et al. (1998) and Fort and Pages (1996). Our analysis of typical behavior can also be performed for heuristically formulated algorithms which lack, for example, a direct relation to a cost function.

The model of the training data is given in Section 2. In Section 3 we present the LVQ algorithms that are studied in this paper. The method for the dynamical and performance analysis of these LVQ algorithms is described in Section 4. In Section 5 we put forward the results in terms of performance of the LVQ algorithms in the proposed theoretical framework. A brief summary and conclusions are presented in Section 6. At the end we provide some key mathematical results used in the paper as an Appendix A.

2. The data model

We study a simple though relevant model situation with two prototypes and two classes. Note that this situation captures important aspects at the decision boundary between classes. Since LVQ-type learning schemes perform a local adaptation, this simple setting provides insight into the dynamics and generalization ability of interesting areas at the class boundaries when learning a more complex data set. We denote the prototypes as $\vec{w}_s \in \mathbb{R}^N, s \in \{1, -1\}$. An input data vector $\vec{\xi} \in \mathbb{R}^N$ is classified as class s iff $d(\vec{\xi}, \vec{w}_s) < d(\vec{\xi}, \vec{w}_{-s})$, where d is some distance measure (typically Euclidean). At every time step μ, the learning process for the prototype vectors makes use of a labeled training example $(\vec{\xi}^\mu, \sigma^\mu)$ where $\sigma^\mu \in \{1, -1\}$ is the class of the observed training data $\vec{\xi}^\mu$.

We restrict our analysis to random input training data which are independently distributed according to a bimodal distribution $P(\vec{\xi}) = \sum_{\sigma=\pm1} p_\sigma P(\vec{\xi}|\sigma)$. p_σ is the prior probability of the class σ, $p_1 + p_{-1} = 1$. In our study we choose the class conditional distribution $P(\vec{\xi}|\sigma)$ as Gaussians with mean vector $\lambda \vec{B}_\sigma$ and independent components with variance v_σ:

$$p(\vec{\xi}|\sigma) = \frac{1}{(2\pi v_\sigma)^{\frac{N}{2}}} \exp\left[-\frac{1}{2}\frac{(\vec{\xi} - \lambda\vec{B}_\sigma)^2}{v_\sigma}\right]. \tag{1}$$

We consider orthonormal class center vectors, i.e. $\vec{B}_l \cdot \vec{B}_m = \delta_{l,m}$, where $\delta_{\cdot,\cdot}$ is the Kronecker delta. The orthogonality condition merely fixes the position of the class centers with respect to the origin while the parameter λ controls the separation of the class centers. $\langle . \rangle$ denotes the average over $P(\vec{\xi})$ and $\langle . \rangle_\sigma$ denotes the conditional averages over $P(\vec{\xi} \mid \sigma)$, hence $\langle . \rangle = \sum_{\sigma=\pm1} p_\sigma \langle . \rangle_\sigma$. For an input from cluster σ we have, for example, $\langle \xi_j \rangle_\sigma = \lambda(\vec{B}_\sigma)_j$ and $\langle \vec{\xi}^2 \rangle_\sigma = \sum_{j=1}^N \langle \xi_j^2 \rangle_\sigma = \sum_{j=1}^N (v_\sigma + \langle \xi_j \rangle_\sigma^2) = v_\sigma N + \lambda^2 \Rightarrow \langle \vec{\xi}^2 \rangle = (p_1 v_1 + p_{-1} v_{-1})N + \lambda^2$. In the mathematical treatment we will exploit formally the thermodynamic limit $N \to \infty$, which corresponds to very high-dimensional data and prototypes. Among other simplifying consequences this allows us, for instance, to neglect the term λ^2 on the right-hand side of the above expression for $\langle \vec{\xi}^2 \rangle$. Hence we have: $\langle \vec{\xi}^2 \rangle \approx N(p_1 v_1 + p_{-1} v_{-1})$.

In high dimensions the Gaussians overlap significantly. The cluster structure of the data becomes apparent when projected into the plane spanned by $\{\vec{B}_1, \vec{B}_{-1}\}$, while projections in a randomly chosen two-dimensional subspace overlap completely. In an attempt to learn the classification scheme, the relevant directions $\vec{B}_{\pm1} \in \mathbb{R}^N$ have to be identified. Obviously this task becomes highly non-trivial for large N.

3. LVQ algorithms

We consider the following generic structure of LVQ algorithms:

$$\vec{w}_l^\mu = \vec{w}_l^{\mu-1} + \frac{\eta}{N} f(\{\vec{w}_l^{\mu-1}\}, \vec{\xi}^\mu, \sigma^\mu)(\vec{\xi}^\mu - \vec{w}_l^{\mu-1}),$$
$$l \in \{\pm1\}, \mu = 1, 2 \ldots \tag{2}$$

where η is the so-called learning rate. The specific form of $f_l = f(\{\vec{w}_l^{\mu-1}\}, \vec{\xi}^\mu, \sigma^\mu)$ is determined by the algorithm. In the following $\mathrm{d}_l^\mu = (\vec{\xi}^\mu - \vec{w}_l^{\mu-1})^2$ is the squared Euclidean distance between the prototype and the new training data. We consider the following learning rules determined by different forms of f_l:

(I) *LVQ2.1:*

$f_l = (l\sigma^\mu)$ (Kohonen, 1990). In our model with two prototypes, LVQ2.1 updates both of them at each learning step according to the class of the training data. A prototype is moved closer to (away from) the data-point if the label of the data is the same as (different from) the label of the prototype. As pointed out in Seo and Obermayer (2003), this learning rule can be seen as a limiting case of maximizing the likelihood ratio of the correct and wrong class distribution which are both described by Gaussian mixtures. Because the ratio is not bounded from above, divergences can occur. Adaptation is often restricted to a window around the decision surface to prevent this behavior. We do not consider a window rule in this article, but we will introduce early stopping to prevent divergence.

(II) *LFM:*

$f_l = (l\sigma^\mu)\Theta(d^\mu_{\sigma^\mu} - d^\mu_{-\sigma^\mu})$, where Θ is the Heaviside function. This is the crisp version of robust soft learning vector quantization (RSLVQ) proposed in Seo and Obermayer (2003). In the model considered here, the prototypes are adapted only according to the misclassified data, hence the name *learning from mistakes* (LFM) is used for this prescription. RSLVQ results from an optimization of a cost function which considers the ratio of the class distribution and unlabeled data distribution. Since this ratio is bounded, stability can be expected.

(III) *LVQ1:*

$f_l = (l\sigma^\mu)\Theta(d^\mu_{-l} - d^\mu_l)$. This extension of competitive learning to labeled data corresponds to Kohonen's original LVQ1 (Kohonen, 1990). The update is towards $\vec{\xi}^\mu$ 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.

(IV) *LVQ+:*

$f_l = \frac{1}{2}[1 + l\sigma^\mu]\Theta(d^\mu_{-l} - d^\mu_l)$. In this scheme the update is non-zero only for a correct winner and, then, always positive. Hence, a prototype $\vec{w}_S$ can only accumulate updates from its own class $\sigma = S$. We will use the abbreviation LVQ+ for this prescription.

(V) *VQ:*

$f_l = \Theta(d^\mu_{-l} - d^\mu_l)$. This update rule disregards the actual data label and always moves the winner towards the example input. It corresponds to unsupervised Vector Quantization (VQ) and aims at finding prototypes which yield a good representation of the data in the sense of Euclidean distances. The choice $f_l = \Theta(d^\mu_{-l} - d^\mu_l)$ can also be interpreted as describing two prototypes which represent the same class and compete for updates from examples from this very class only.

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

$$e(\vec{\xi}^\mu) = \sum_{S=\pm 1} \frac{1}{2}(\vec{\xi}^\mu - \vec{w}_S^{\mu-1})^2 \Theta(d^\mu_{-S} - d^\mu_{+S}); \tag{3}$$

see e.g. Freking, Reents, and Biehl (1996) for details. While intuitively clear and well motivated, the other algorithms mentioned above lack such a straightforward interpretation as stochastic gradient descent with respect to a cost function.

4. Dynamics and performance analysis

The key steps for the dynamical analysis of LVQ-type learning rules can be sketched as follows: 1. The original system with many degrees of freedom is characterized in terms of only few quantities, the so-called *macroscopic order parameters*. For these quantities, recursion relations can be derived from the learning algorithm. 2. Application of the central limit theorem enables us to perform an average over the random sequence of example data by means of Gaussian integrations. 3. Self-averaging properties of the order parameters allow us to restrict the description to their mean values. Fluctuations of the stochastic dynamics can be neglected in the limit $N \to \infty$. 4. A *continuous time limit* leads to the description of the dynamics using coupled, deterministic ordinary differential equations (ODEs) in terms of the above-mentioned order parameters. 5. The (numerical) integration of the ODEs for a given modulation function and initial conditions yields the evolution of order parameters in the course of learning. From the latter one can directly obtain the learning curve, i.e. the generalization ability of the LVQ classifier as a function of the number of example data.

ODEs for the learning dynamics: We assume that learning is driven by statistically independent training examples such that the process is Markovian. For an underlying data distribution which has the same second order statistics as the mixture of Gaussians introduced above, the system dynamics can be analysed using only few order parameters which depend on the relevant characteristics of the prototypes. In our setting, the system will be described in terms of the projections $\{R_{lm} = \vec{w}_l \cdot \vec{B}_m, Q_{lm} = \vec{w}_l \cdot \vec{w}_m\}, l, m \in \{\pm 1\}$. In the thermodynamic limit, these order parameters become *self-averaging* (Reents & Urbanczik, 1998), i.e. the fluctuations about their mean-values can be neglected as $N \to \infty$. In Reents and Urbanczik (1998) a detailed mathematical foundation of this self-averaging property is given for on-line learning. This property facilitates an analysis of the stochastic evolution of the prototype vectors in terms of a deterministic system of differential equations and helps to analyse the system in an exact theoretical way. One can get the following recurrence relations from the generic learning scheme 2 (Ghosh, Biehl, Freking, & Reents, 2004):

$$R^\mu_{lm} - R^{\mu-1}_{lm} = \frac{\eta}{N}(b^\mu_m - R^{\mu-1}_{lm})\, f_l \tag{4}$$

$$Q^\mu_{lm} - Q^{\mu-1}_{lm} = \frac{\eta}{N}((h^\mu_l - Q^{\mu-1}_{lm})\, f_m + (h^\mu_m - Q^{\mu-1}_{lm})\, f_l + \eta f_l \times f_m) \tag{5}$$

where $h_l^\mu = \vec{w}_l^{\mu-1} \cdot \vec{\xi}^\mu$, $b_m^\mu = \vec{B}_m \cdot \vec{\xi}^\mu$, $R_{lm}^\mu = \vec{w}_l^\mu \cdot \vec{B}_m$, $Q_{lm}^\mu = \vec{w}_l^\mu \cdot \vec{w}_m^\mu$. As the analysis is done for very large N, terms of $\mathcal{O}(1/N^2)$ are neglected in (5).

Defining $t \equiv \frac{\mu}{N}$, for $N \to \infty$, t can be conceived as a continuous time variable and the order parameters R_{lm} and Q_{lm} as functions of t become self-averaging with respect to the random sequence of input training data. An average is performed over the disorder introduced by the randomness in the training data and (4) and (5) become a coupled system of differential equations (Ghosh et al., 2004):

$$\frac{\mathrm{d}R_{lm}}{\mathrm{d}t} = \eta(\langle b_m f_l\rangle - \langle f_l\rangle R_{lm}) \tag{6}$$

$$\frac{\mathrm{d}Q_{lm}}{\mathrm{d}t} = \eta(\langle h_l f_m\rangle - \langle f_m\rangle Q_{lm} + \langle h_m f_l\rangle - \langle f_l\rangle Q_{lm}) + \eta^2 \left(\sum_{\sigma=\pm1} v_\sigma p_\sigma \langle f_l \times f_m\rangle_\sigma \right). \tag{7}$$

After plugging in the exact forms of f_l, the averages in (6) and (7) can be computed in terms of an integration over the density $p(\vec{x} = (h_1, h_{-1}, b_1, b_{-1}))$ (Ghosh et al., 2004), see Appendix A. The central limit theorem yields that, in the limit $N \to \infty$, $\vec{x} \sim N(C_\sigma, \mu_\sigma)$ (for class σ) where μ_σ and C_σ are the mean vector and covariance matrix of $\vec{x}$ respectively, cf. Ghosh et al. (2004). The first order and second order statistics of $\vec{x}$, viz. μ_σ and C_σ respectively, can be computed using the following conditional averages (Ghosh et al., 2004):

$$\begin{aligned}
&\langle h_S^\mu\rangle_\sigma = \lambda R_{S\sigma}^{\mu-1}, \qquad \langle b_\tau^\mu\rangle_\sigma = \lambda\delta_{\sigma,\tau},\\
&\langle h_S^\mu h_T^\mu\rangle_\sigma - \langle h_S^\mu\rangle_\sigma\langle h_T^\mu\rangle_\sigma = v_\sigma Q_{ST}^{\mu-1}\\
&\langle h_S^\mu b_\tau^\mu\rangle_\sigma - \langle h_S^\mu\rangle_\sigma\langle b_\tau^\mu\rangle_\sigma = v_\sigma R_{S\tau}^{\mu-1},\\
&\langle b_\rho^\mu b_\tau^\mu\rangle_\sigma - \langle b_\rho^\mu\rangle_\sigma\langle b_\tau^\mu\rangle_\sigma = v_\sigma\delta_{\rho,\tau}
\end{aligned}$$

where $S, T, \sigma, \rho, \tau \in \{1, -1\}$ and $\delta_{.,.}$ is the Kronecker delta. Hence, the density of $h_{\pm1}^\mu$ and $b_{\pm1}^\mu$ is given in terms of the model parameters λ, $p_{\pm1}$, $v_{\pm1}$ and the set of order parameters as defined above. Note that this holds for any distribution $p(\vec{\xi})$ with the same second order statistics as characterized above, thus it is not necessary for the analysis that the conditional densities in Eq. (1) of $\vec{\xi}$ are Gaussians.

Inserting the closed form expressions of these averages (cf. (12), (16) and (17)) we get the final form of the system of ODEs for a given learning rule. In Eq. (20) we present the final form of ODEs for the LFM algorithm. For the other algorithms, we refer to Ghosh et al. (2004).

The system of ODEs for a specific modulation function, f_l, can explicitly be solved by (possibly numeric) integration. In the case of LVQ2.1 an exact analytic integration is possible. For given initial conditions $\{R_{ST}(0), Q_{ST}(0)\}$, learning rate η, and model parameters $\{p_{\pm1}, \lambda, v_{\pm1}\}$ one obtains the typical evolution of the characteristic quantities $\{R_{ST}(t), Q_{ST}(t)\}$. This forms the basis for an analysis of the performance and the convergence properties of LVQ-type algorithms in this study.

We will consider training of prototypes that are initialized as independent random vectors of squared length $\hat{Q}_{\pm1}$ with no prior knowledge about the cluster positions. In terms of order parameters this implies $Q_{11}(0) = \hat{Q}_1$, $Q_{-1,-1}(0) = \hat{Q}_{-1}$, $Q_{1-1}(0) = R_{ST}(0) = 0$ for all $S, T = \pm1$, $\hat{Q} \approx \hat{Q}_{-1}$.

Generalization ability: After training, the success of learning can be quantified in terms of the generalization error, i.e. the probability for misclassifying novel, random data which did not appear in the training sequence. The generalization error can be decomposed into the two contributions stemming from misclassified data from cluster $\sigma = 1$ and cluster $\sigma = -1$:

$$\varepsilon_g = p_1\varepsilon_1 + p_{-1}\varepsilon_{-1} \quad \text{with } \varepsilon_\sigma = \langle\Theta(d_{-\sigma} - d_{+\sigma})\rangle_\sigma. \tag{8}$$

Exploiting the central limit theorem in the same fashion as above, one can formulate the generalization error (ε_g) as an explicit function of the order parameters and data statistics (see Appendix A and Ghosh et al. (2004)):

$$\varepsilon_\sigma = \phi\left[\frac{Q_{\sigma\sigma} - Q_{-\sigma-\sigma} - 2\lambda(R_{\sigma\sigma} - R_{-\sigma\sigma})}{2\sqrt{v_\sigma}\sqrt{Q_{11} - 2Q_{1-1} + Q_{-1-1}}}\right] \tag{9}$$

where $\phi(z) = \int_{-\infty}^{z} \mathrm{d}x \frac{\mathrm{e}^{-x^2/2}}{\sqrt{2\pi}}$.

By inserting $\{R_{ST}(t), Q_{ST}(t)\}$ we obtain the learning curve $\varepsilon_g(t)$, i.e. the typical generalization error after on-line training with tN random examples. Here, once more, we exploit the fact that the order parameters and, thus, also ε_g are self-averaging non-fluctuating quantities in the thermodynamic limit $N \to \infty$.

In order to verify the correctness of the aforementioned theoretical framework, we compare the solutions of the system of differential equations with the Monte Carlo simulation results and find an excellent agreement already for $N \geq 100$ in the simulations. Fig. 1(a) and (b) show how the average result in simulations approaches the theoretical prediction and how the corresponding variance vanishes with increasing N.

For a 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 many examples, $\tilde{t} = \eta t \to \infty$. In the absence of a cost function we can still consider the above limit, in which the system of ODEs simplifies and can be expressed in terms of the rescaled $\tilde{t}$ after neglecting terms $\propto \eta^2$. A fixed point analysis then yields a well defined asymptotic configuration, cf. Freking et al. (1996). The dependence of the asymptotic ε_g on the choice of learning rate is illustrated for LVQ1 in Fig. 1(c).

5. Results—performance of the LVQ algorithms

In Fig. 2 we illustrate the evolution of the generalization error in the course of training. Qualitatively all the algorithms show a similar evolution of the generalization error along the training process. The performance of the algorithms is evaluated in terms of stability and generalization error. To quantify the generalization ability of the algorithms we define the following performance measure:

$$PM = \frac{\sqrt{\int_{p_1=0}^{1} (\varepsilon_{g,p_1,lvq} - \varepsilon_{g,p_1,bld})^2 \mathrm{d}p_1}}{\sqrt{\int_{p_1=0}^{1} \varepsilon_{g,p_1,bld}^2 \mathrm{d}p_1}} \tag{10}$$

where $\varepsilon_{g,p_1,lvq}$ and $\varepsilon_{g,p_1,bld}$ are the generalization errors that can be achieved by a given LVQ algorithm (except for LVQ2.1

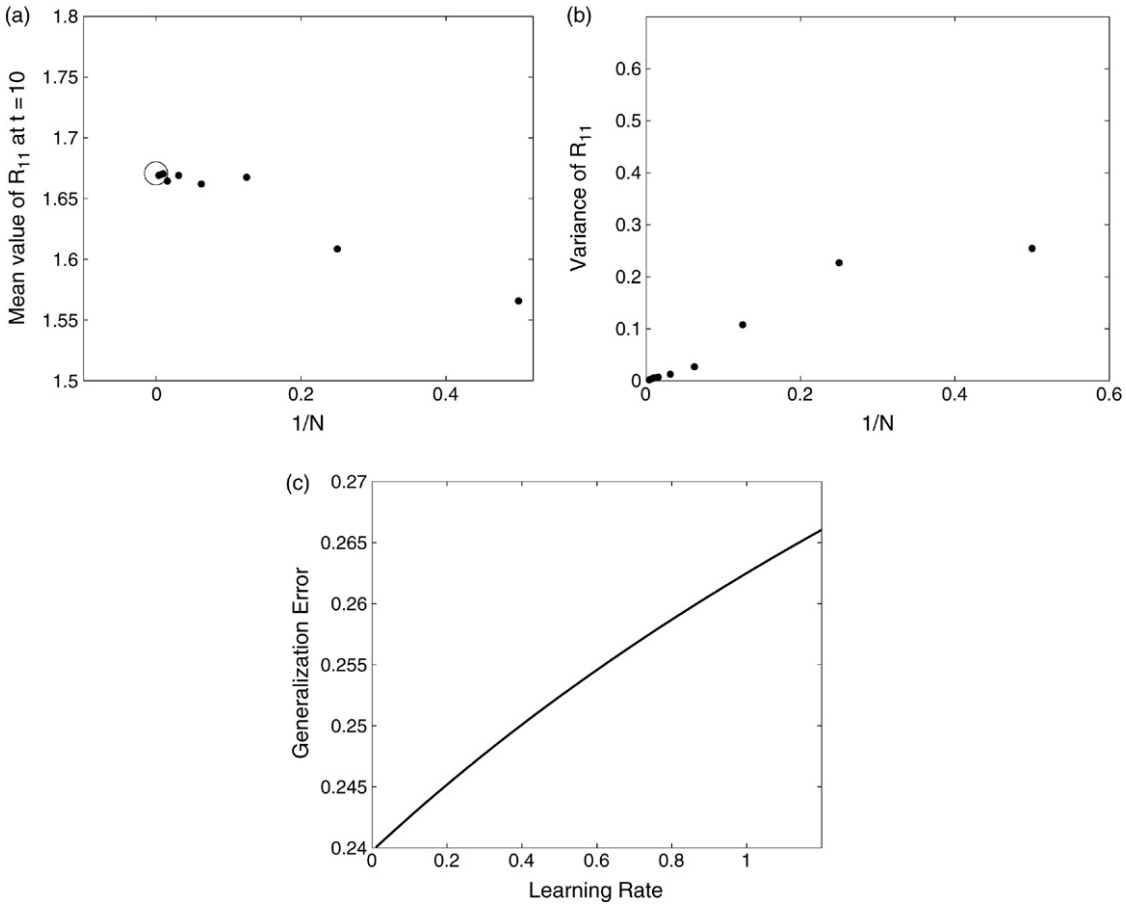

Fig. 1. (a: top left frame) Convergence of Monte Carlo results to the theoretical prediction for $N \to \infty$. The open ring at $\frac{1}{N} = 0$ marks the theoretical result for R_{11} at $t = 10$; dots correspond to Monte Carlo results on average over 100 independent runs. (b: top right frame) Self-averaging property: the variance of R_{11} at $t = 10$ vanishes with increasing N in Monte Carlo simulations. In both (a) and (b) following parameter values are used as one example setting: $v_1 = 9$, $v_{-1} = 16$, $\lambda = 3$, $p_1 = 0.8$, $\eta = 0.1$. (c: bottom frame) Dependence of the generalization error for $t \to \infty$ for LVQ1 (with parameter values: $\lambda = v_1 = v_{-1} = 1$, $p_1 = 0.5$) on the learning rate η.

this corresponds to the $t \to \infty$ asymptotic ε_g) and the best linear decision rule, respectively, for a given class prior probability p_1. Unless otherwise specified, the generalization error of an optimal linear decision rule is depicted as a dotted line in the figures and the solid line represents the performance of the corresponding LVQ algorithm. We evaluate the performance of the algorithms in terms of the asymptotic generalization error for two example parameter settings: (i) Equal class variances ($v_1 = v_{-1}$): $v_1 = v_{-1} = 1, \lambda = 1$. (ii) Unequal class variances ($v_1 \neq v_{-1}$): $v_1 = 0.25$, $v_{-1} = 0.81$, $\lambda = 1$. Hereinafter, unless otherwise mentioned the results illustrated in all figures representing the performances of the algorithms in terms of ε_g are obtained choosing these parameter values and the following initialization of $\vec{w}_{\pm 1}$: $R_{11}(0) = R_{-11}(0) = R_{1-1}(0) = R_{-1-1}(0) = 0$, $Q_{11}(0) = 0.001$, $Q_{1-1}(0) = 0$, $Q_{-1-1}(0) = 0.002$.

LVQ2.1: In Fig. 3(a) we illustrate the divergent behavior of LVQ2.1. If the prior probabilities are skewed the prototype corresponding to the class with lower probability diverges during the learning process and this results in a trivial classification with $\varepsilon_{g,p_1,lvq2.1} = \min(p_1, p_{-1})$. Note that in the singular case when $p_1 = p_{-1}$ the behavior of the differential equations differs from the generic case and LVQ2.1 yields prototypes which are symmetric about $\frac{\lambda(B_1+B_{-1})}{2}$. Hence the performance is optimal in the case of equal priors (Fig. 4(a)).

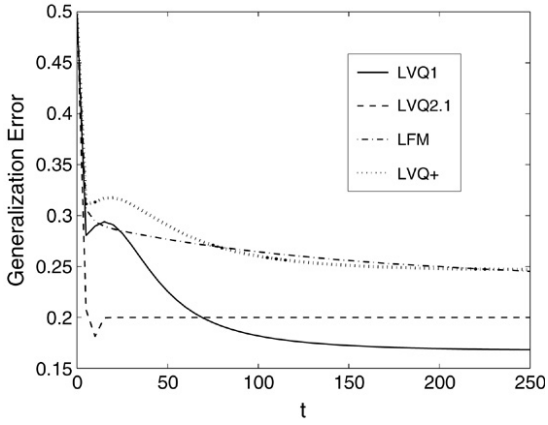

Fig. 2. Evolution of the generalization error for different LVQ algorithms. Parameters: $v_1 = v_{-1} = \lambda = 1$, $\eta = 0.2$, $p_1 = 0.8$. As the objective of VQ is to minimize the quantization error and not to achieve good generalization ability, evolution of ε_g for VQ is not shown in this figure.

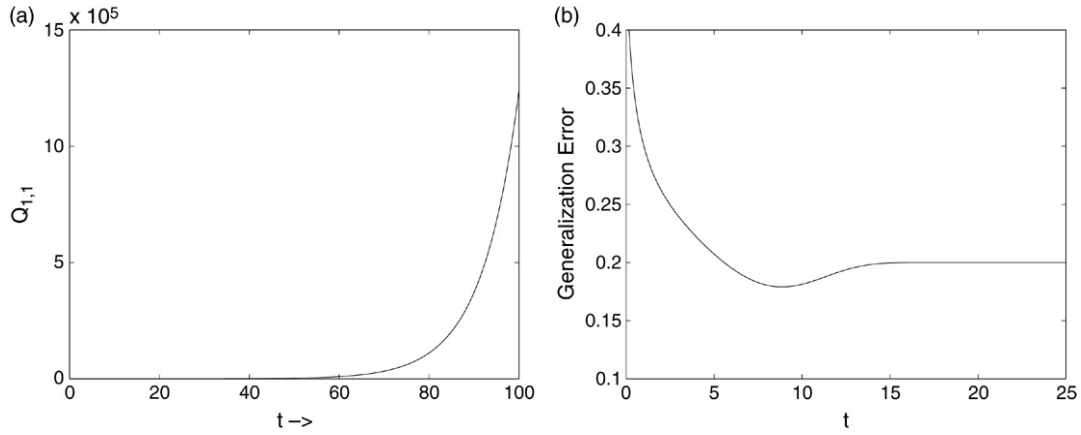

Fig. 3. (a: left frame) Diverging order parameter in LVQ2.1. (b: right frame) Modality of generalization error with respect to t in the case of LVQ2.1. In both figures the results are for parameter values: $v_1 = v_{-1} = \lambda = 1$, $\eta = 0.2$, $p_1 = 0.2$.

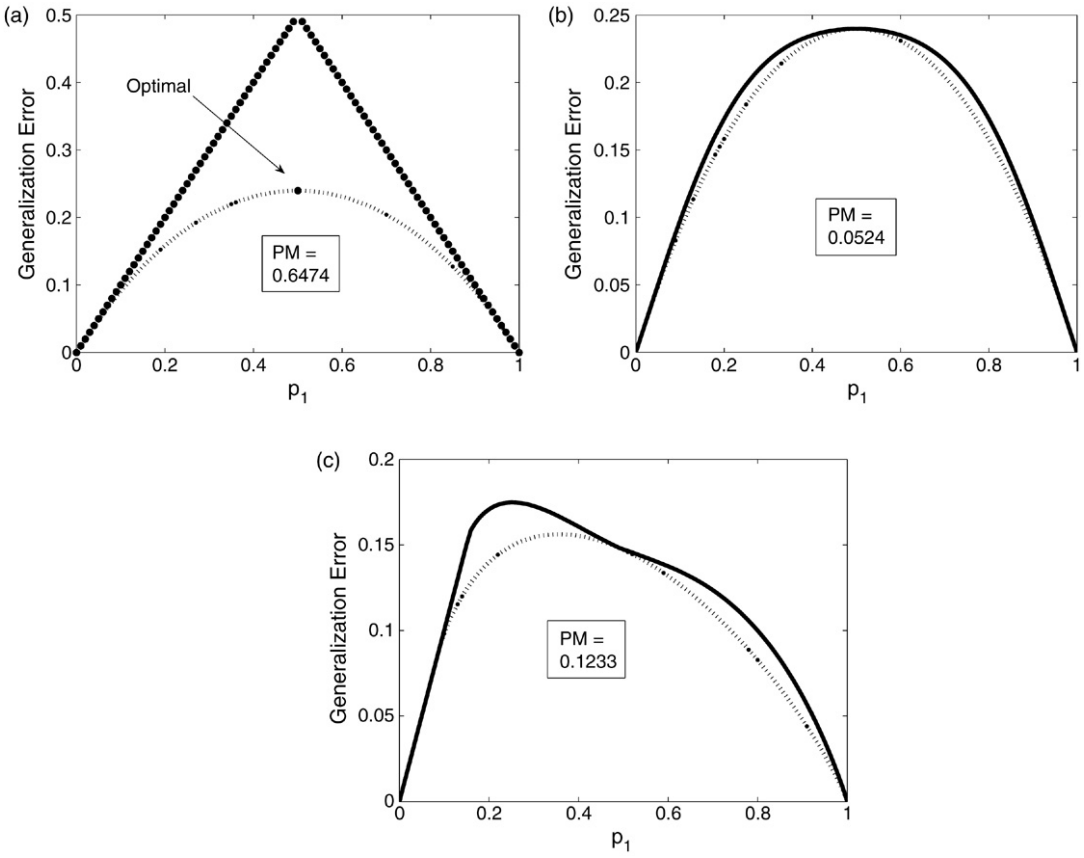

Fig. 4. Performance of LVQ2.1: (a: top left frame) Asymptotic behavior for $v_1 = v_{-1}$; note that for $p_1 = 0.5$ the performance is optimal, in all other cases $\varepsilon_g = \min(p_1, p_{-1})$. To highlight the different characteristics of LVQ2.1 compared to the other algorithms studied here we have shown the asymptotic generalization error through a dot plot instead of a solid line used for the other algorithms. (b: top right frame) LVQ2.1 with the stopping criterion for $v_1 = v_{-1}$. The performance is near optimal. (c: bottom frame) LVQ2.1 with the stopping criterion when $v_1 \neq v_{-1}$. The performance measure *PM* as given here and in the following figures is defined in (10).

In high dimensions, this divergent behavior can also be observed if a window rule of the original formulation (Seo & Obermayer, 2003) is used (Ghosh et al., 2004), thus this heuristic does not prevent the instability of the algorithm. Alternative modifications will be the subject of further work. As the most important objective of a classification algorithm is to achieve minimal generalization error, one way to deal with this divergent behavior of LVQ2.1 is to *stop* at a point when the generalization error is minimal, e.g. as measured on a validation set. In Fig. 3(b) we see that typically the generalization error has a modality with respect to t, hence an optimal stopping point exists. In Fig. 4 we illustrate the performance of LVQ2.1.

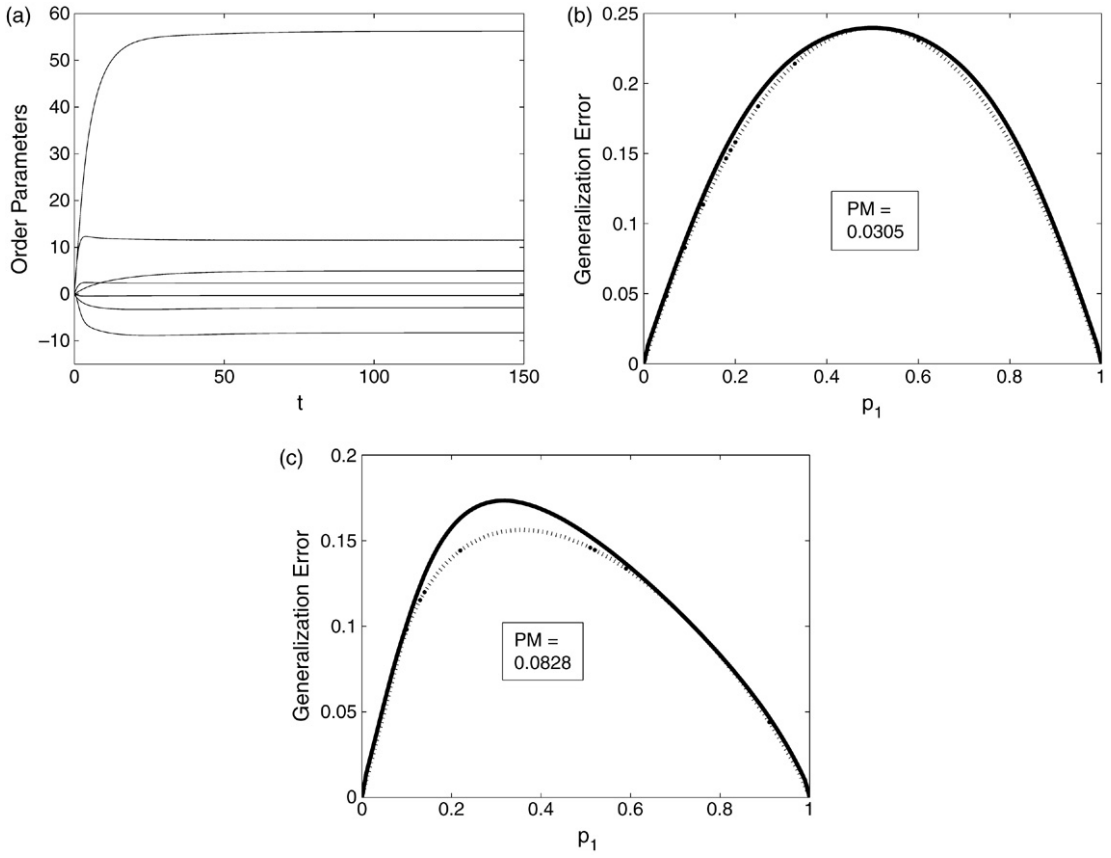

Fig. 5. Performance of LVQ1: (a: top left frame) Dynamics of the order parameters for model parameters: $v_1 = 4$, $v_{-1} = 9$, $\lambda = 2$, $p_1 = 0.8$ and $\eta = 1.8$. (b: top right frame) Generalization with $v_1 = v_{-1}$. The performance is near optimal. (c: bottom frame) Generalization for $v_1 \neq v_{-1}$. The performance is worse than that in the case when $v_1 = v_{-1}$.

Fig. 4(a) shows the poor asymptotic behavior. Only for equal priors it achieves optimal performance. However, as depicted in Fig. 4(b), an idealized early stopping method as described above can indeed give near optimal behavior for the equal class variance case. The performance is worse when we deal with unequal class variances (Fig. 4(c)). It is important to note that the existence of the minimum in $\varepsilon_g(t)$ and its location and depth depend on the precise initialization of the vectors $\vec{w}_{\pm 1}(0)$ see Fig. 11. This initialization issue is discussed in detail later.

LVQ1: Fig. 5(a) shows the convergent behavior of LVQ1. The asymptotic generalization error as achieved by LVQ1 is typically quite close to the potential optimum $\varepsilon_{g,p_1,bld}$. Fig. 5(b) and (c) display the asymptotic generalization error as a function of the prior p_1 in two different settings of the model. In the completely symmetric situation with equal variances and balanced priors, $p_1 = p_{-1}$, the LVQ1 result coincides with the best linear decision boundary which is through $\lambda(\vec{B}_1 + \vec{B}_{-1})/2$ for this setting. Whenever the cluster-variances are different, the symmetry about $p_1 = 1/2$ is lost but the performance is optimal for one particular (v_1, v_{-1})-dependent value of $p_1 \in (0, 1)$. Unlike LVQ2.1 the good performance of LVQ1 is invariant to initialization of prototype vectors $w_{\pm 1}$, see Fig. 11.

LFM: The dynamics of the LFM algorithm is shown in Fig. 6(a). We see that its performance is far from optimal in both equal (Fig. 6(b)) and unequal class variance (Fig. 6(c)) cases. Hence, though the LFM algorithm converges to a stable configuration of the prototypes, it fails to give a near optimal performance in terms of the asymptotic generalization error.

Further interesting properties which can be detected within this theoretical analysis of the LFM algorithm are as follows: (i) The asymptotic generalization error is independent of learning rate η. It merely controls the magnitude of the fluctuations orthogonal to $\{\vec{B}_1, \vec{B}_{-1}\}$ and the asymptotic distance of the prototypes from the decision boundary. (ii) $p_1\varepsilon_1 = p_{-1}\varepsilon_{-1}$ cf. Eq. (8). That means, the two contributions to the total ε_g, Eqs. (8) and (9), become equal for $t \to \infty$. As a consequence, LFM updates are based on balanced data, asymptotically, as they are restricted to misclassified examples.

Note that we consider only the crisp LFM procedure here. It is very well possible that *soft* realizations of RSLVQ as discussed in Seo et al. (2003) and Seo and Obermayer (2003) yield significantly better performance.

VQ: In Fig. 7(a) we see the evolution of the order parameters in the course of training for VQ. The dynamics of VQ has been studied in detail in Freking et al. (1996) for the case of equal class variances ($v_1 = v_{-1}$) and equal priors ($p_1 = p_{-1}$). Though the objective of VQ is to minimize the quantization error and not to achieve a good generalization ability yet we can compute the asymptotic ε_g from the prototype configuration. In Fig. 7 we illustrate the asymptotic generalization error for

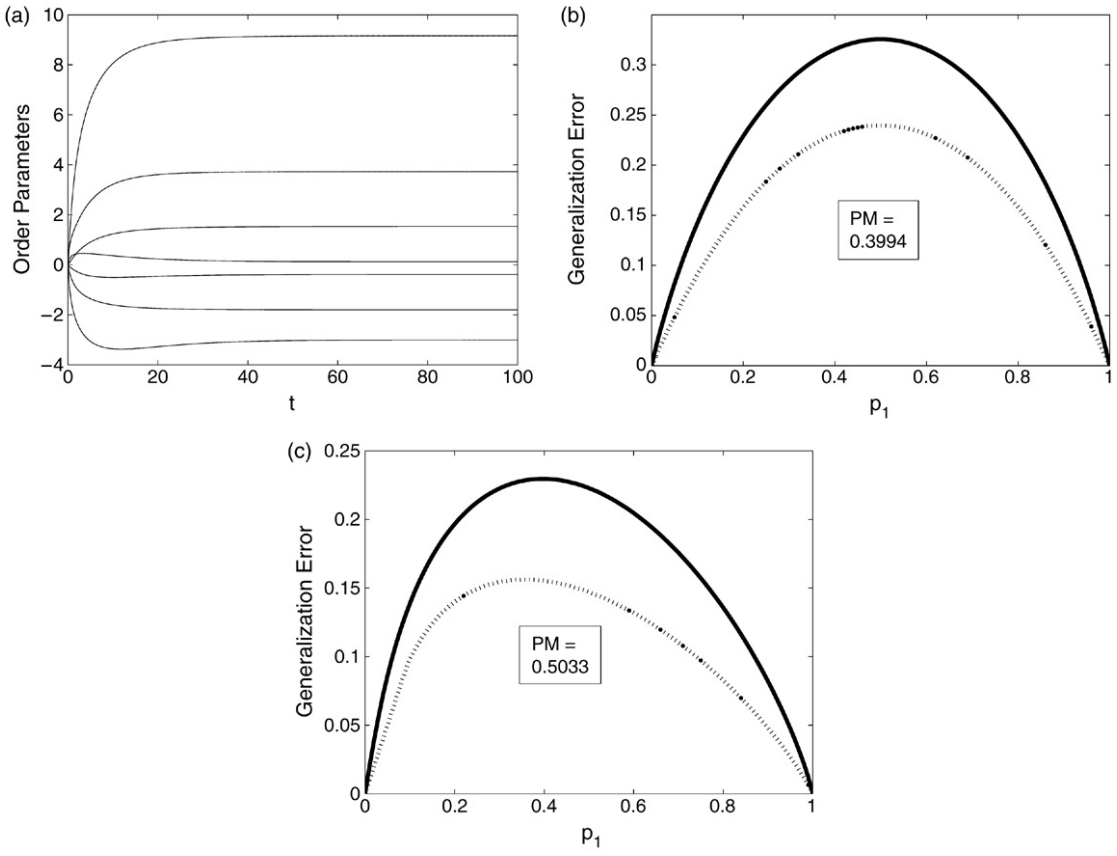

Fig. 6. Performance of LFM: (a: top left frame) Convergence for the following parameter values: $v_1 = v_{-1} = \lambda = 1$, $p_1 = 0.8$ and $\eta = 3$. (b: top right frame) Generalization with $v_1 = v_{-1}$. (c: bottom frame) Generalization with $v_1 \neq v_{-1}$. In both cases the performance of LFM is far from optimal.

the VQ algorithm and note the following interesting facts. In unsupervised VQ a strong prevalence, e.g. $p_1 \approx 1$, will be accounted for by placing both vectors inside the stronger cluster, thus achieving a low quantization error. Obviously this yields a poor classification as indicated by the asymptotic value $\varepsilon_g = 1/2$ in the limiting cases $p_1 = 0$ or 1. In the equal class variance case for $p_1 = 1/2$ the aim of representation happens to coincide with good generalization and ε_g becomes optimal, Fig. 7(b). In Fig. 7(c) we see that in the case of unequal class variances there exists a prior probability p_1 for which the generalization error of VQ is identical to that of the best linear decision surface.

LVQ+: In Fig. 8(a) we illustrate the convergence of the LVQ+ algorithm. LVQ+ updates each $\vec{w}_S$ only with data from class S. As a consequence, the asymptotic positions of the $\vec{w}_{\pm 1}$ are always symmetric about the geometrical center $\lambda(\vec{B}_1 + \vec{B}_{-1})$ and the asymptotic ε_g is independent of the priors $p_{\pm 1}$. Thus, in the equal class variance case (Fig. 8(b)) LVQ+ is robust with respect to the variations of $p_{\pm 1}$ after training, i.e. it is optimal in the sense of the minmax-criterion $\sup_{p \pm 1} \varepsilon_g(t)$ (Duda et al., 2000). However, in the unequal class variance case this minmax property is not observed (Fig. 8(c)). Nevertheless the resulting asymptotic ε_g depends linearly on p_1 and is tangent to ε_g^{bld}.

Comparison of the performance of the algorithms: To facilitate a better understanding, we compare the performances of the algorithms in Fig. 9, where we present the asymptotic performance of the three relevant LVQ algorithms: LVQ1, LVQ2.1 with the idealized stopping criterion, and LFM. As the performances of LVQ+ and VQ are qualitatively entirely different (Figs. 8 and 7) from the other algorithms, these two algorithms are not discussed in this comparison.

In Fig. 9(a), we see that LVQ1 outperforms the other algorithms for equal class variances, and LVQ2.1 with the early stopping criterion yields results which are only slightly worse. However, the superiority of LVQ1 is partly lost in the case of unequal class variances (see Fig. 9(b)) where an interval for p_1 exists for which the performance of LVQ2.1 with the stopping criterion is better than LVQ1. However, if we compare the overall performance of LVQ1 for $v_1 \neq v_{-1}$ through the performance measure *PM* defined in (10) with other algorithms then LVQ1 is found to be the best algorithm among these LVQ variants.

The good performance of the LVQ1 algorithm can be further investigated through a geometrical analysis of relevant quantities. Fig. 10 displays the trajectories of prototypes projected onto the plane spanned by $\vec{B}_1$ and $\vec{B}_{-1}$. Note that, as can be expected from symmetry arguments, the $(t \to \infty)$-asymptotic projections of prototypes into the $\vec{B}_{\pm 1}$-plane are along the axis connecting the cluster centers. Moreover, in the limit $\eta \to 0$, their asymptotic positions lie exactly on the plane and fluctuations orthogonal to $\vec{B}_{\pm 1}$ vanish. This is signaled by the fact that the order parameters for $\tilde{t} \to \infty$ satisfy

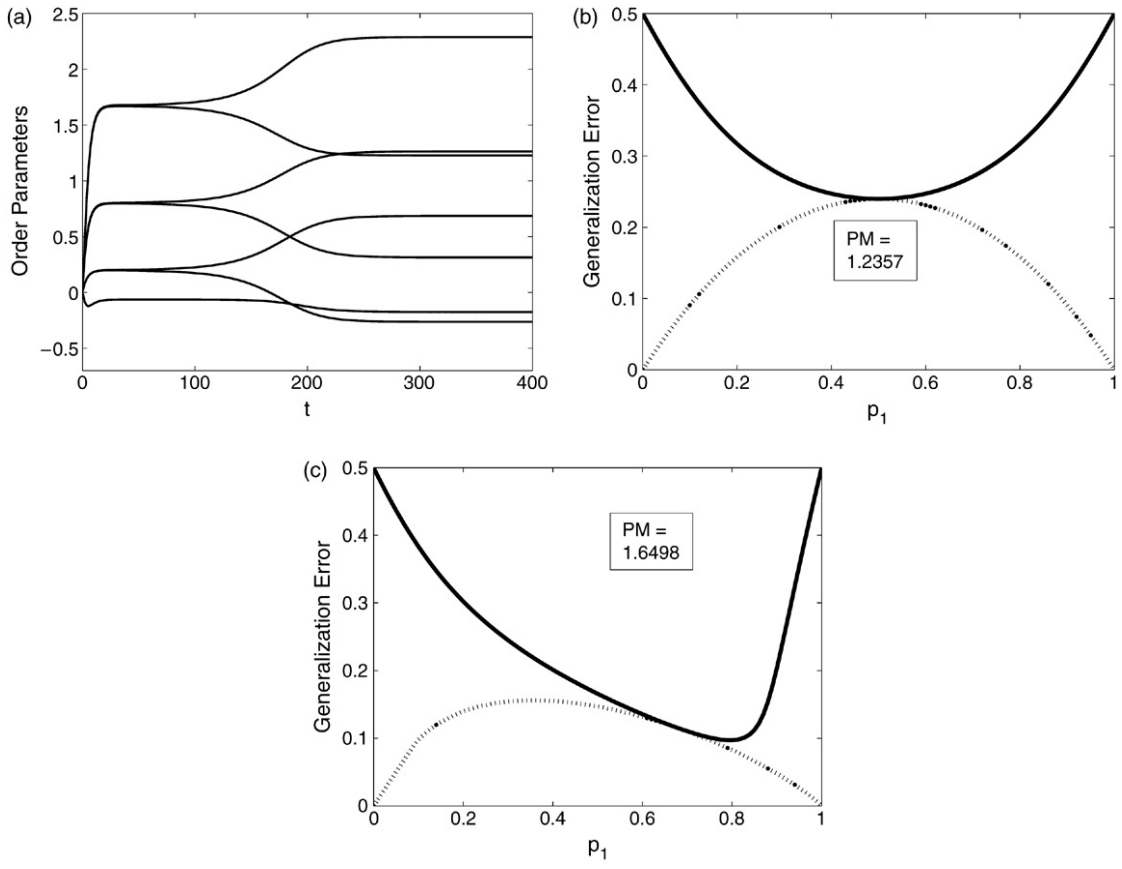

Fig. 7. Performance of VQ: (a: top left frame) Evolution of order parameters, parameters: $v_1 = v_{-1} = \lambda = 1$, $p_1 = 0.8$ and $\eta = 0.5$. (b: top right frame) Generalization error when $v_1 = v_{-1}$ (c: bottom frame) Generalization for $v_1 \neq v_{-1}$.

$Q_{SS} = R_{S1}^2 + R_{S-1}^2$, and $Q_{1-1} = R_{11}R_{-11} + R_{1-1}R_{-1-1}$ which implies:

$$\vec{w}_S(\tilde{t} \to \infty) = R_{S1}\vec{B}_1 + R_{S-1}\vec{B}_{-1} \quad \text{for } S = \pm 1. \tag{11}$$

Hence we can conclude that the actual prototype vectors asymptotically approach the above unique configuration. Note that, in general, stationarity of the order parameters does not necessarily imply that $\vec{w}_{\pm 1}$ converge to points in the N-dimensional space. For LVQ1 with $\eta > 0$, for instance, fluctuations in the space orthogonal to $\{\vec{B}_1, \vec{B}_{-1}\}$ persist even for constant $\{R_{ST}, Q_{ST}\}$.

Fig. 10 reveals further information about the learning process. When learning from unbalanced data, e.g. $p_1 > p_{-1}$ in this case, the prototype representing the stronger cluster will be updated more frequently and in fact *overshoots*, resulting in a non-monotonic behavior of ε_g (Fig. 2).

In Fig. 9 we see that the performances of LVQ1 and LVQ2.1 with the stopping criterion are comparable. As the distribution of the training data is unknown to the algorithms the performance of the algorithms should also be judged in terms of robustness to the initialization of the prototype vectors $\vec{w}_{\pm 1}$. In Fig. 11 we illustrate this robustness of the algorithms LVQ1 and LVQ2.1 with stopping by considering an initialization difference from the usual $R_{11}(0) = R_{-11}(0) = R_{1-1}(0) = R_{-1-1}(0) = 0$, $Q_{11}(0) = 0.001$, $Q_{1-1}(0) = 0$, $Q_{-1-1}(0) = 0.002$, as specified in the caption. We find that though there are variations in the learning curves (ε_g versus t) the asymptotic performance (ε_g for $t \to \infty$) of LVQ1 is robust to initialization. However, the performance of LVQ2.1 with the stopping criterion is extremely sensitive to initialization; for a good performance it is required that the initial decision boundary is already close to the optimal one. Since no density estimation is performed prior to the training procedure such an ideal initialization of $\vec{w}_{\pm 1}$ cannot be assured in general.

Another interesting aspect of the LVQ1, VQ and LVQ+ algorithms is highlighted in Fig. 12. For unequal class variances, these three algorithms give optimal performance for the same value of p_1 viz. in the neighborhood of 0.65 for the model parameters used here.

6. Summary and conclusions

We have investigated different variants of LVQ-type algorithms in an exact mathematical way by means of the theory of on-line learning. For $N \to \infty$, using concepts from statistical physics, the system dynamics can be described by few characteristic quantities, and the learning curves can be evaluated exactly also for heuristically motivated learning algorithms where a global cost function is lacking, like for standard LVQ1, or where a cost function has only been proposed for a soft version like for LVQ2.1 and LFM.

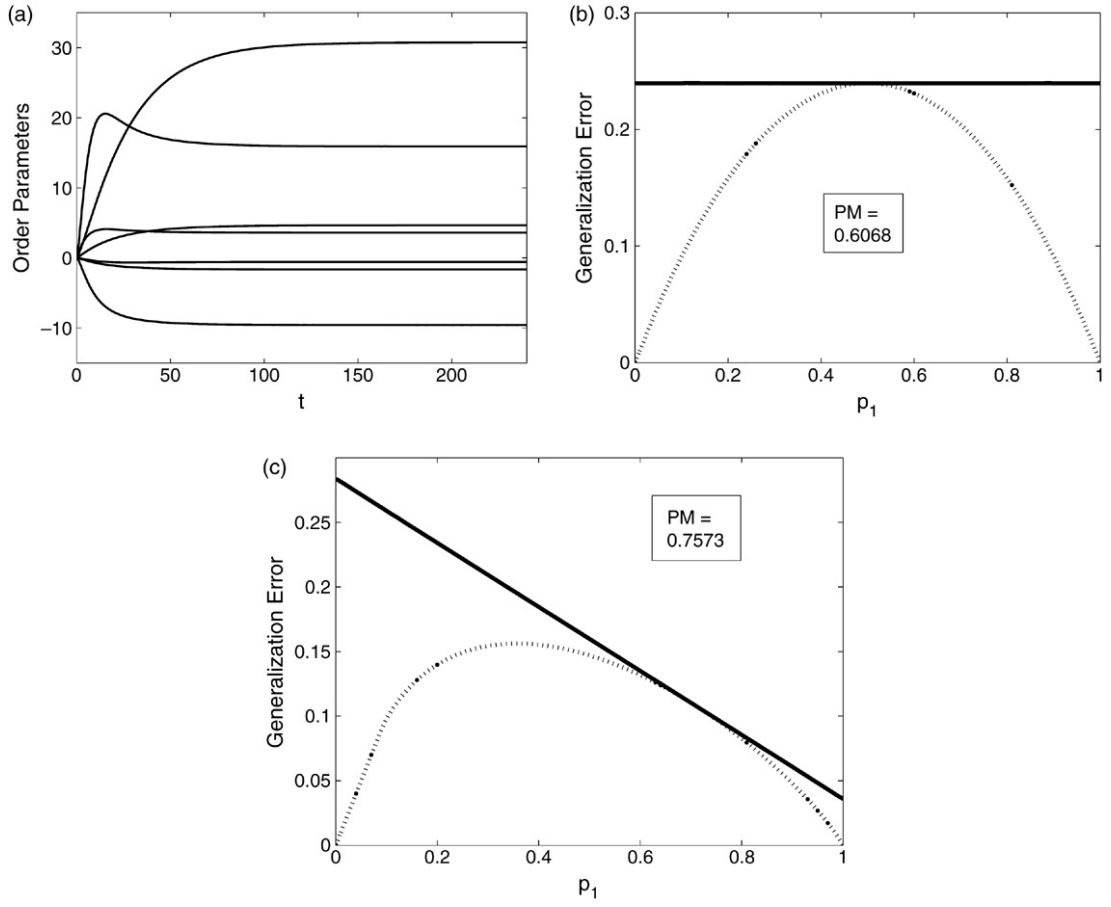

Fig. 8. Performance of LVQ+: (a: top left frame) Convergence with parameter values: $v_1 = 9$, $v_{-1} = 16$, $\lambda = 3$, $p_1 = 0.8$ and $\eta = 0.5$. (b: top right frame) Generalization with $v_1 = v_{-1}$. The performance is independent of the class prior probabilities (minmax). (c: bottom frame) Generalization with unequal variances $v_1 \neq v_{-1}$.

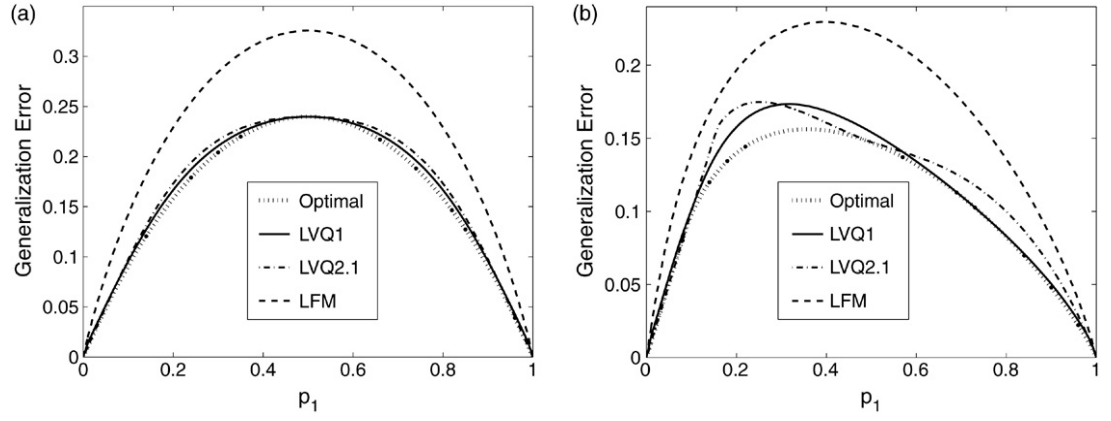

Fig. 9. Comparison of asymptotic performances of the algorithms. For the LVQ2.1 algorithm the performance with the stopping criterion is shown. (a: left frame) For $v_1 = v_{-1}$ LVQ1 outperforms all other algorithms. (b: right frame) When $v_1 \neq v_{-1}$ the absolute supremacy of LVQ1 is lost. There exists a range of values of p_1 for which LVQ2.1 with the stopping criterion outperforms LVQ1. However, this performance of the LVQ2.1 with stopping is extremely sensitive to the initialization of prototypes.

Surprisingly, fundamentally different limiting solutions are observed for the algorithms LVQ1, LVQ2.1, LFM, LVQ+ although their learning rules are quite similar. The behavior of LVQ2.1 is unstable and modifications such as a stopping rule become necessary. The generalization ability of the algorithms differs in particular for unbalanced class distributions. Even more involved properties are revealed when the class variances differ. It is remarkable that the basic LVQ1 algorithm shows near optimal generalization error for all choices of the prior distribution in the equal class variance case. LVQ2.1 with the

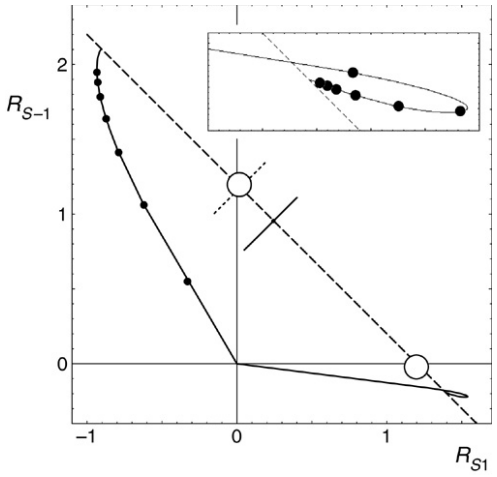

Fig. 10. LVQ1 for $\lambda = 1.2$, $v_1 = v_{-1} = 1$, and $p_1 = 0.8$. Trajectories of prototypes in the limit $\eta \to 0$, $t \to \infty$. Solid lines correspond to the projections of prototypes into the plane spanned by $\lambda\vec{B}_1$ and $\lambda\vec{B}_{-1}$ (marked by open circles). The dots correspond to the pairs of values $\{R_{S1}, R_{S-1}\}$ observed at $\tilde{t} = \eta t = 2, 4, 6, 8, 10, 12, 14$ in Monte Carlo simulations with $\eta = 0.01$ and $N = 200$, averaged over 100 runs. Note that, because $p_1 > p_{-1}$, $\vec{w}_1$ approaches its final position much faster and in fact overshoots. The inset displays a close-up of the region around its stationary location. The short solid line marks the asymptotic decision boundary as parameterized by the prototypes, the short dashed line marks the best linear decision boundary. The latter is very close to $\lambda\vec{B}_{-1}$ for the pronounced dominance of the $\sigma = 1$ cluster with $p_1 = 0.8$.

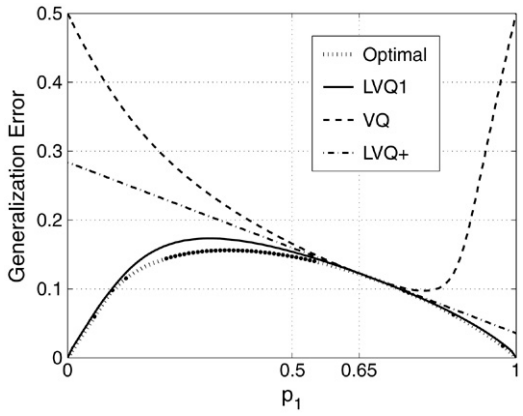

Fig. 12. The ε_g versus p_1 curves for LVQ1, VQ and LVQ+ touch that of the best linear decision surface at the same prior probability p_1 (in the neighborhood of 0.65 for the parameter values used) for the unequal class variance case, $v_1 \neq v_{-1}$.

stopping criterion also performs close to optimal for equal class variances. In the unequal class variance case, LVQ2.1 with stopping outperforms the other algorithms for a range of p_1 when appropriate initial conditions are used.

However, the good performance of LVQ2.1 with the idealized stopping criterion is highly sensitive to initialization of prototype vectors. The performance degrades heavily if the prototypes are not initialized in such a way that the initial decision surface is close to the optimal one. Due to an unknown density prior to training the positioning of the cluster centers is unknown in a practical scenario and hence the aforementioned ideal initialization cannot be assured, whereas the asymptotic performance of LVQ1 does not depend on initialization, though it yields learning curves (ε_g versus t) which are different for different initializations. This partially mirrors the well-known effects of LVQ1 for given settings where a data cluster from a different class can act as a barrier, slowing down the convergence significantly.

Another interesting finding from this theoretical analysis is that in the equal class variance case LVQ+ achieves a minmax

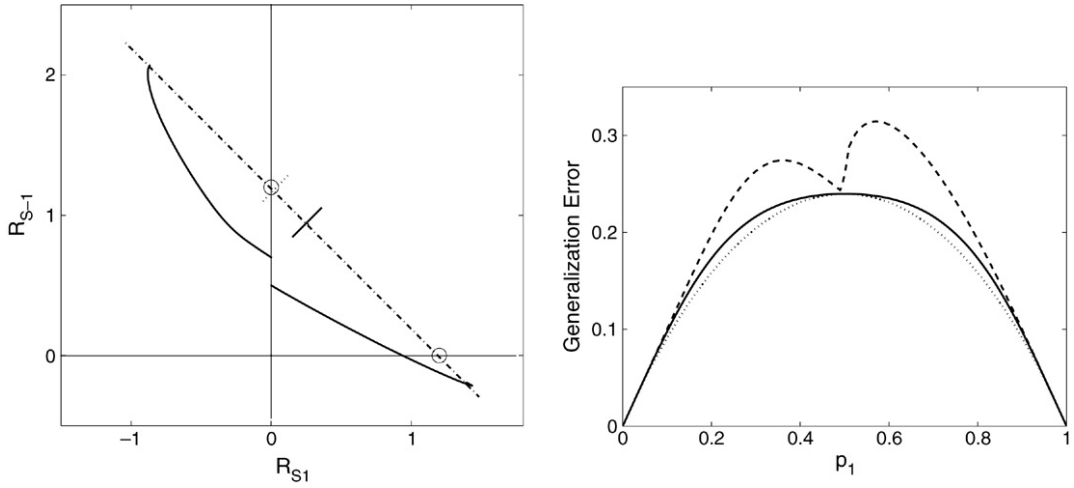

Fig. 11. Left panel: The trajectories of the prototypes in the plane spanned by $\{\vec{B}_{\pm 1}\}$ corresponding to the following initialization: $R_{11}(0) = R_{-11}(0) = 0$, $R_{1-1}(0) = 0.5$, $R_{-1-1}(0) = 0.7$, $Q_{11}(0) = 1.8$, $Q_{1-1}(0) = 0$, $Q_{-1-1}(0) = 2.9$. Comparing with Fig. 10 we see that though the learning curves are different due to different trajectories yet the asymptotic generalization errors are the same since the final configuration is invariant to the initialization of $\vec{w}_{\pm 1}$. The other parameter values are the same as in Fig. 10. Right panel: The performance of LVQ2.1 with the stopping criterion heavily depends on the initialization; the solid line corresponds to the initialization: $R_{11}(0) = R_{-11}(0) = R_{1-1}(0) = R_{-1-1}(0) = 0$, $Q_{11}(0) = 0.001$, $Q_{1-1}(0) = 0$, $Q_{-1-1}(0) = 0.002$, whereas the dashed line is for: $R_{11}(0) = R_{-11}(0) = 0$, $R_{1-1}(0) = 0.5$, $R_{-1-1}(0) = 0.7$, $Q_{11}(0) = 1.8$, $Q_{1-1}(0) = 0$, $Q_{-1-1}(0) = 2.9$. The dotted line corresponds to $\varepsilon_{g,p_1,bld}$. The results are for $v_1 = v_{-1}$.

characteristics; however, this special property is lost in the unequal class variance case.

The theoretical framework proposed in this article will be used to study further characteristics of the dynamics such as fixed points, asymptotic positioning of the prototypes, etc. The main goal of the research presented in this article is to provide a deterministic description of the stochastic evolution of the learning process in an exact mathematical way for interesting learning rules in relevant (though simple) situations, which will be helpful in constructing efficient (in the Bayesian sense) LVQ algorithms.

Appendix A. The mathematical treatment

Here we outline some key mathematical results used in the analysis. The formalism was first used in the context of unsupervised vector quantization (Freking et al., 1996) and the calculations were recently detailed in a Technical Report (Ghosh et al., 2004).

Throughout this appendix indices $l, m, k, s, \sigma \in \{\pm 1\}$ represent the class labels or cluster memberships. We furthermore use the following shorthands: (i) $\Theta_S = \Theta(d_{-S} - d_{+S})$ for LVQ1, LVQ+ and VQ and (ii) $\Theta_\sigma = \Theta(d_\sigma - d_{-\sigma})$ for LFM. For convenience the three winner takes all algorithms, LVQ1, LVQ+, VQ, are collectively called the WTA algorithms.

A.1. Averages

In order to obtain the final form the ODEs for a given modulation function, averages over the joint density $P(h_1, h_{-1}, b_1, b_{-1})$ are performed.

LVQ2.1: The elementary averages involved in the system of ODEs for the LVQ2.1 can be expressed in a closed form as follows:

$$\langle \sigma b_m \rangle = \sum_{\sigma=\pm 1} \sigma p_\sigma \lambda \delta_{m,\sigma}, \langle \sigma h_m \rangle = \sum_{\sigma=\pm 1} \sigma p_\sigma \lambda R_{m\sigma},$$
$$\langle \sigma \rangle = \sum_{\sigma=\pm 1} \sigma p_\sigma. \tag{12}$$

Other algorithms: After inserting the corresponding modulation function f_l of LVQ1, LVQ+, VQ and LFM in the system of ODEs presented in Eqs. (6) and (7) we encounter Heaviside functions of the following generic form:

$$\Theta_S = \Theta(\vec{\alpha}_S.\vec{x} - \beta_S). \tag{13}$$

In the case of WTA algorithms (LVQ1, LVQ+, VQ): $\Theta_S = \Theta(d_{-S} - d_{+S}) = \Theta(\vec{\alpha}_S.\vec{x} - \beta_S)$ with

$$\vec{\alpha}_S = (+2S, -2S, 0, 0) \text{ and } \beta_S = S(Q_{+S+S} - Q_{-S-S}), \tag{14}$$

and for LFM: $\Theta_\sigma = \Theta(d_\sigma - d_{-\sigma}) = \Theta(\vec{\alpha}_\sigma.\vec{x} - \beta_\sigma)$ with

$$\vec{\alpha}_\sigma = (-2\sigma, +2\sigma, 0, 0) \quad \text{and} \quad \beta_\sigma = -(Q_{\sigma\sigma} - Q_{-\sigma-\sigma}). \tag{15}$$

After plugging in the modulation function f_l performing the averages in Eqs. (6) and (7) for the LFM, LVQ1, VQ and LVQ+ algorithms involves conditional means of the form

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

where $(\vec{x})_n$ is the nth component of $\vec{x} = (h_1, h_{-1}, b_1, b_{-1})$.

The above-mentioned averages can be expressed in a closed form in the following way (Ghosh et al., 2004):

$$\langle (\vec{x})_n \Theta_s \rangle_k = \frac{(C_k \vec{\alpha}_s)n}{\sqrt{2\pi}\tilde{\alpha}_{sk}} \exp\left[-\frac{1}{2}\left(\frac{\tilde{\beta}_{sk}}{\tilde{\alpha}_{sk}}\right)^2\right] + (\vec{\mu}_k)_n \Phi\left(\frac{\tilde{\beta}_{sk}}{\tilde{\alpha}_{sk}}\right). \tag{16}$$

$$\langle \Theta_s \rangle_k = \Phi\left(\frac{\tilde{\beta}_{sk}}{\tilde{\alpha}_{sk}}\right) \tag{17}$$

where

$$\tilde{\alpha}_{sk} = \|C_k^{\frac{1}{2}} \vec{\alpha}_s\| = \sqrt{\vec{\alpha}_s C_k \vec{\alpha}_s} \quad \text{and} \quad \tilde{\beta}_{sk} = \vec{\alpha}_s \cdot \vec{\mu}_k - \beta_s \tag{18}$$

$$\Phi(x) = \int_{-\infty}^{x} \mathrm{d}z \frac{\mathrm{e}^{-\frac{z^2}{2}}}{\sqrt{2\pi}}. \tag{19}$$

Using these averages the final form of the system of differential equations corresponding to different algorithms are obtained (Ghosh et al., 2004).

For brevity we give an example of such final form for the LFM algorithm only and refer to Ghosh et al. (2004) for other algorithms:

$$\begin{aligned}
\frac{\mathrm{d}R_{lm}}{\mathrm{d}t} &= \eta l \left(\sum_{\sigma=\pm 1} \sigma p_\sigma \left[\frac{(C_\sigma \vec{\alpha}_\sigma) n_{bm}}{\sqrt{2\pi}\tilde{\alpha}_{\sigma\sigma}} \exp\left[-\frac{1}{2}\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right)^2\right] + (\vec{\mu}_\sigma)_{n_{bm}} \Phi\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right) \right] \right. \\
&\quad \left. - \sum_{\sigma=\pm 1} \sigma p_\sigma \left[\Phi\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right) \right] R_{lm} \right) \\
\frac{\mathrm{d}Q_{lm}}{\mathrm{d}t} &= \eta \left(l \sum_{\sigma=\pm 1} \sigma p_\sigma \left[\frac{(C_\sigma \vec{\alpha}_\sigma) n_{hm}}{\sqrt{2\pi}\tilde{\alpha}_{\sigma\sigma}} \exp\left[-\frac{1}{2}\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right)^2\right] \right. \right. \\
&\quad \left. + (\vec{\mu}_\sigma)_{n_{hm}} \Phi\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right) \right] - l \sum_{\sigma=\pm 1} \sigma p_\sigma \left[\Phi\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right) \right] Q_{lm} \\
&\quad + m \sum_{\sigma=\pm 1} \sigma p_\sigma \left[\frac{(C_\sigma \vec{\alpha}_\sigma)_{n_{hl}}}{\sqrt{2\pi}\tilde{\alpha}_{\sigma\sigma}} \exp\left[-\frac{1}{2}\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right)^2\right] \right. \\
&\quad \left. + (\vec{\mu}_\sigma)_{n_{hl}} \Phi\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right) \right] - m \sum_{\sigma=\pm 1} \sigma p_\sigma \left[\Phi\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right) \right] \\
&\quad \left. \times Q_{lm} + lm\eta \sum_{\sigma=\pm 1} v_\sigma p_\sigma \Phi\left(\frac{\tilde{\beta}_{\sigma\sigma}}{\tilde{\alpha}_{\sigma\sigma}}\right) \right).
\end{aligned} \tag{20}$$

Here, again, we have to insert $\tilde{\alpha}_{\sigma\sigma}$ and $\tilde{\beta}_{\sigma\sigma}$, as defined in (18) with $\vec{\alpha}_\sigma = (-2\sigma, +2\sigma, 0, 0)$ and $\beta_\sigma = -(Q_{\sigma\sigma} - Q_{-\sigma-\sigma})$. Also,

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

A.2. The generalization error

Using (17) we can directly compute the generalization error as follows: $\varepsilon_g = \sum_{k=\pm 1} p_{-k} \langle \Theta_k \rangle_{-k} = \sum_{k=\pm 1} p_{-k} \Phi\left(\frac{\tilde{\beta}_{k-k}}{\tilde{\alpha}_{k-k}}\right)$ which yields Eq. (9) in the text after inserting $\tilde{\alpha}_{sk}$ and $\tilde{\beta}_{sk}$ as given in (18) with $\vec{\alpha}_S = (+2S, -2S, 0, 0)$ and $\beta_S = S(Q_{+S+S} - Q_{-S-S})$.

References

Biehl, M., & Caticha, N. (2003). The statistical mechanics of online learning and generalization. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks*. MIT Press.

Biehl, M., Ghosh, A., & Hammer, B. (2006). Learning vector quantization: the dynamics of winner-takes-all algorithms. *Neurocomputing*, *69*, 660–670.

Cottrell, M., Fort, J. C., & Pages, G. (1998). Theoretical aspects of the S.O.M algorithm, survey. *Neuro-computing*, *21*, 119–138.

Crammer, K., Gilad-Bachrach, R., Navot, A., & Tishby, A. Margin analysis of the LVQ algorithm. In *NIPS*.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification. 2e*. New York: Wiley.

Engel, A., & van den Broeck, C. (Eds.) (2001). *The statistical mechanics of learning*. Cambridge University Press.

Fort, J. C., & Pages, G. (1996). Convergence of stochastic algorithms: From the Kushner and Clark theorem to the Lyapounov functional. *Advances in applied probability*, *28*, 1072–1094.

Freking, A., Reents, G., & Biehl, M. (1996). The dynamics of competitive learning. *Europhysics Letters*, *38*, 73–78.

Ghosh, A., Biehl, M., Freking, A., & Reents, G. (2004). A theoretical framework for analysing the dynamics of LVQ: A statistical physics approach. Technical Report 2004-9-02, Mathematics and Computing Science, University Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands, December 2004, available from http://www.cs.rug.nl/~biehl.

Hammer, B., Strickert, M., & Villmann, T. (2005a). On the generalization ability of GRLVQ networks. *Neural Processing Letters*, *21*(2), 109–120.

Hammer, B., Strickert, M., & Villmann, T. (2005b). Supervised neural gas with general similarity measure. *Neural Processing Letters*, *21*(1), 21–44.

Hammer, B., & Villmann, T. (2002). Generalized relevance learning vector quantization. *Neural Networks*, *15*, 1059–1068.

Kohonen, T. (1990). Improved versions of learning vector quantization. *IJCNN, International Joint conference on Neural Networks*, *1*, 545–550.

Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer.

McDermott, E., & Katagiri, S. (1994). Prototype-based minimum classification error/generalized probabilistic descent training for various speech units. *Computer Speech and Language*, *8*(4), 351–368.

Neural Networks Research Centre, Bibliography on the self-organizing maps (som) and learning vector quantization (lvq). Otaniemi: Helsinki University of Technology. Available on-line: http://liinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html.

Reents, G., & Urbanczik, R. (1998). Self-averaging and on-line learning. *Physical Review Letters*, *80*(24), 5445–5448.

Saad, D. (Ed.), (1998). *Online learning in neural networks*. Cambridge University Press.

Sato, A. S., & Yamada, K. (1995). Generalized learning vector quantization. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Advances in neural information processing systems*: *vol. 7* (pp. 423–429).

Seo, S., Bode, M., & Obermayer, K. (2003). Soft nearest prototype classification. *IEEE Transactions on Neural Networks*, *14*(2), 390–398.

Seo, S., & Obermayer, K. (2003). Soft learning vector quantization. *Neural Computation*, *15*, 1589–1604.