Dynamical analysis of LVQ type learning rules

Anarta Ghosh[1,3], **Michael Biehl**[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
3- **anarta@cs.rug.nl**

Abstract - *Learning vector quantization (LVQ) constitutes a powerful and simple method for adaptive nearest prototype classification which has been introduced based on heuristics. Recently, a mathematical foundation by means of a cost function has been proposed which, as a limit case, yields a learning rule very similar to classical LVQ2.1 and also motivates a modification thereof which shows better stability. However, the exact dynamics as well as the generalization ability of the LVQ algorithms have not been investigated so far in general. Using concepts from statistical physics and the theory of on-line learning, we present a rigorous mathematical investigation of the dynamics of LVQ type classifiers in a prototypical scenario. Interestingly, one can observe significant differences of the algorithmic stability and generalization ability and quite unexpected behavior for these only slightly different variants of LVQ.*

Key words - Online learning, LVQ, 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 [15, 13, 14]. Several modifications of the basic LVQ have been proposed which aim at a larger flexibility, faster convergence, more flexible metrics, or better adaptation to Bayesian decision boundaries, etc. [5, 11, 13]. Thereby, most learning schemes including the basic LVQ have been proposed on heuristic grounds and their dynamics is not clear. In particular, there exist powerful extensions like LVQ2.1 which require the introduction of additional heuristics, the so-called window rule for stability, which are not well understood. A rigorous mathematical analysis of statistical learning algorithms is possible in some cases and might yield quite unexpected results as demonstrated in [3, 7].
Recently, a few approaches relate LVQ-type learning schemes to exact mathematical concepts and thus open the way towards a solid mathematical justification for the success of LVQ as well as exact means to design stable algorithms with good generalization ability. The directions are mainly twofold: on the one hand, the proposal of cost functions which, possibly as a limit case, lead to LVQ-type gradient schemes such as the approaches in [11, 19, 20], thereby, the nature of the cost function has consequences on the stability of the algorithm as pointed out in [18, 20]; on the other hand, the derivation of generalization bounds of the algorithms, as obtained in [4, 10] using statistical learning theory. Interestingly, the cost function of some

extensions of LVQ includes a term which measures the structural risk thus aiming at margin maximization during training similar to support vector machines [10]. However, the analysis is limited to the considered cost function or the usual (crisp) nearest prototype classification is covered only as a limit case if statistical models are used.
In this work we introduce a theoretical framework in which we analyze and compare different LVQ algorithms extending first results presented in [2]. The dynamics of training is studied along the successful theory of on-line learning [1, 6, 17], 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. This formalism is used to investigate learning rules which are similar to the very successful LVQ2.1, including the original LVQ1 as well as a recent modification [20].

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 receptive fields, thus it provides insight into the dynamics and generalization ability of interesting areas 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 normal distribution with mean vector $\lambda\vec{B}_\sigma$ and independent components with variance v_σ. Without loss of generality we consider orthonormal class centre vectors, i.e. $\vec{B}_l \cdot \vec{B}_m = \delta_{l,m}$, where $\delta_{.,.}$ is the Kronecker delta. Hence the parameter λ controls the separation of the class centres. $\langle . \rangle$ denotes the average over $P(\vec{\xi})$ and $\langle . \rangle_\sigma$ denotes the conditional averages over $P(\vec{\xi}|\sigma)$, hence $\langle . \rangle = \sum_{\sigma=\pm1} p_\sigma \langle . \rangle_\sigma$. The mathematical treatment presented in the study is based on the *thermodynamic limit* $N \to \infty$. Note for instance that $\langle \vec{\xi} \cdot \vec{\xi} \rangle \approx N(p_1 v_1 + p_{-1} v_{-1})$ because $\langle \vec{\xi}^2 \rangle_\sigma \approx N v_\sigma$ holds for $N \gg \lambda$.
In high dimensions the Gaussians overlap significantly. The cluster structure of the data becomes apparent when projected into the plane spanned by $\left\{\vec{B}_1, \vec{B}_{-1}\right\}$, and 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. Hence, though the model is a simple one, it is interesting from the practical point of view.

3 LVQ algorithms

We consider the following generic structure of LVQ algorithms:

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

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 $d_l^\mu = (\vec{\xi}^\mu - \vec{w}_l^{\mu-1})^2$ is the squared Euclidean distance between the prototype and the new training data. We consider the following forms of f_l:

LVQ2.1: $f_l = (l\sigma^\mu)$ [12]. 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 [20], this learning rule can be seen as a limit case of a maximization of the log-likelihood ratio of the correct and wrong class distribution which are both described by Gaussian mixtures. Because the ratio is not bounded from above, divergence can occur. Adaptation is often restricted to a window around the decision border to prevent this behavior.

LFM: $f_l = (l\sigma^\mu)\Theta\big(d^\mu_{-\sigma^\mu} - d^\mu_{\sigma^\mu}\big)$, where Θ is the Heaviside function. This is the crisp version of robust soft learning vector quantization (RSLVQ) proposed in [20]. In the model considered here, the prototypes are adapted only according to 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.

LVQ1: $f_l = l\sigma^\mu\Theta\big(d^\mu_{-l} - d^\mu_l\big)$. This is Kohonen's original LVQ1 algorithm. At each learning step the prototype which is closest to the data-point, i.e the winner is updated [12].

4 Dynamics of learning

We assume that learning is driven by statistically independent training examples such that the process is Markovian. For not too complicated underlying data distribution, the system dynamics can be analyzed using only few order parameters, $\{R_{lm} = \vec{w}_l \cdot \vec{B}_m, Q_{lm} = \vec{w}_l \cdot \vec{w}_m\}$. In the thermodynamic limit, these order parameters become *self-averaging* [16], i.e. the fluctuations about their mean-values can be neglected as $N \to \infty$. This property facilitates an analysis of the stochastic evolution of the prototype vectors in terms of a deterministic system of differential equations, which greatly helps to build a theoretical understanding of such systems. One can get the following recurrence relations from (1) [9]:

$$R^\mu_{l,m} - R^{\mu-1}_{l,m} = \frac{\eta}{N}\big(b^\mu_m - R^{\mu-1}_{lm}\big) f_l \tag{2}$$

$$Q^\mu_{l,m} - Q^{\mu-1}_{l,m} = \frac{\eta}{N}\Big((h^\mu_l - Q^{\mu-1}_{lm}) f_m + (h^\mu_m - Q^{\mu-1}_{lm}) f_l + \eta f_l \times f_m\Big) \tag{3}$$

where $h^\mu_l = \vec{w}^{\mu-1}_l \cdot \vec{\xi}^\mu, b^\mu_m = \vec{B}_m \cdot \vec{\xi}^\mu, R^\mu_{l,m} = \vec{w}^\mu_l \cdot \vec{B}_m, Q^\mu_{l,m} = \vec{w}^\mu_l \cdot \vec{w}^\mu_m$. As the analysis is done for very large N, terms of $\mathcal{O}(1/N^2)$ are neglected in (3). Define $t \equiv \frac{\mu}{N}$. For $N \to \infty$, t can be conceived as a continuous time variable and the order parameters $R_{l,m}$ and $Q_{l,m}$ 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 (2) and (3) become a coupled system of differential equations [9]:

$$\frac{dR_{l,m}}{dt} = \eta\big(\langle b_m f_l\rangle - \langle f_l\rangle R_{lm}\big) \tag{4}$$

$$\frac{dQ_{l,m}}{dt} = \eta\Big(\langle h_l f_m\rangle - \langle f_m\rangle Q_{lm} + \langle h_m f_l\rangle - \langle f_l\rangle Q_{lm} + \eta \sum_{\sigma=\pm 1} v_\sigma p_\sigma \langle f_l \times f_m\rangle_\sigma\Big) \tag{5}$$

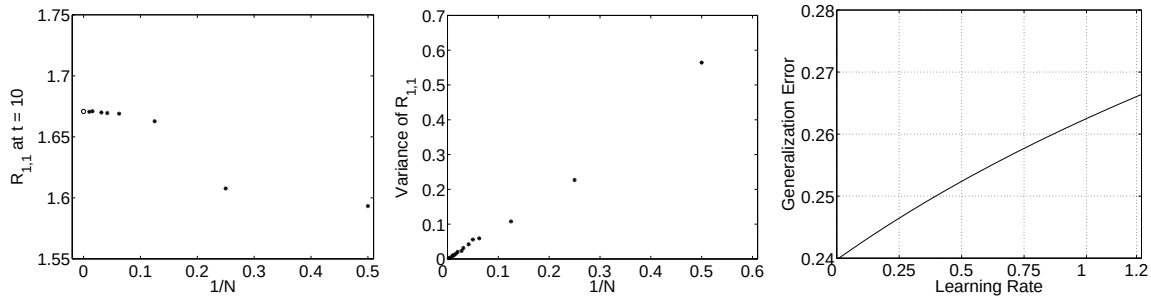

Figure 1: (a: left frame) Convergence of Monte Carlo results to the theoretical prediction for $N \to \infty$. The open ring at $\frac{1}{N} = 0$ marks the theoretical result for $R_{1,1}$ at $t = 10$; stars correspond to Monte Carlo results on average over 100 independent runs. (b: middle frame) Self averaging property: the variance of $R_{1,1}$ at $t = 10$ vanishes with increasing N in Monte Carlo simulations. (c: right 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.

In [9] we show in detail how the averages in (4) and (5) can be computed in terms of an integration over the density $p\big(\vec{x} = (h_1, h_{-1}, b_1, b_{-1})\big)$. 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. [9]. Since this holds for any well behaved $p(\vec{\xi})$, the above mentioned explicit Gaussian assumption of the conditional densities of $\vec{\xi}$ is not necessary for the analysis.

Given the initial conditions $\{R_{l,m}(0), Q_{l,m}(0)\}$, the above mentioned system of coupled ordinary differential equations can be integrated either numerically or analytically. This yields the evolution of the order parameters with increasing t in the course of training. The properties of these *learning curves* depend on the characteristics of the data $\{\lambda, \vec{B}_l \cdot \vec{B}_m\}$, the learning rate η, and the choice of the specific algorithm i.e. the form of f_l. The detailed derivation of the system of differential equations for each of the above mentioned LVQ algorithms is presented in [9]. In our analysis we use $R_{l,m}(0) = Q_{1,-1}(0) = 0, Q_{1,1}(0) = 0.01, Q_{-1,-1}(0) = 0.02$ as the initial conditions for the system of differential equations. For large N, the Central Limit theorem can also be exploited to obtain the generalization error ε_g of a given configuration as a function of the order parameters as follows: $\varepsilon_g = \sum_{\sigma=\pm 1} p_\sigma \Phi[\frac{Q_{\sigma,\sigma} - Q_{-\sigma,-\sigma} - 2\lambda(R_{\sigma,\sigma} - R_{-\sigma,\sigma})}{2\sqrt{v_\sigma}\sqrt{Q_{\sigma,\sigma} - 2Q_{\sigma,-\sigma} + Q_{-\sigma,-\sigma}}}]$, where $\Phi(x) = \int_{-\infty}^{x} e^{-\frac{z^2}{2}}$, c.f. [9]. Hence the evolution of $R_{l,m}$ and $Q_{l,m}$ with the rescaled number of examples t provides us with the learning curve $\varepsilon_g(t)$ as well. 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 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 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 ODE simplifies and can be expressed in the rescaled $\tilde{t}$ after neglecting terms $\propto \eta^2$. A fixed point analysis then yields a well defined asymptotic configuration, c.f. [8]. The dependence of the asymptotic ε_g on the choice of learning rate is illustrated for LVQ1 in Fig. 1(c).

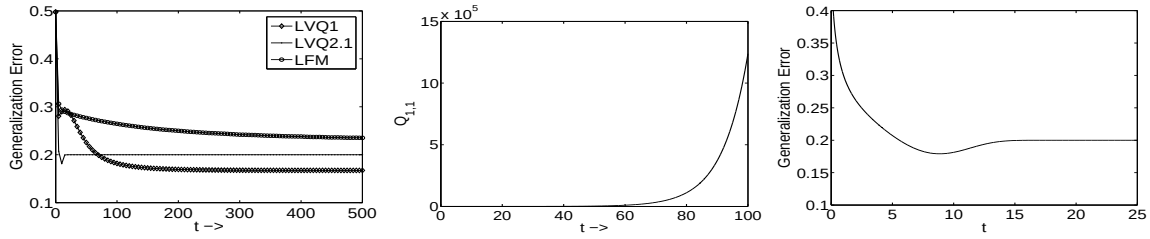

Figure 2: (a: left frame) Evolution of the generalization error for different LVQ algorithms. (b: middle frame) Diverging order parameter in LVQ2.1. (c: right frame) Modality of generalization error with respect to t in the case of LVQ2.1.

5 Results - Performance of the LVQ algorithms

In Fig. 2(a) we illustrate the evolution of the generalization error in the course of training. Qualitatively all the algorithms show a similar evolution of the 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: $PM = \sqrt{\int_{p_1=0}^{1}(\varepsilon_{g,p_1,lvq} - \varepsilon_{g,p_1,bld})^2\, dp_1} \Big/ \sqrt{\int_{p_1=0}^{1} \varepsilon^2_{g,p_1,bld}\, dp_1}$ where $\varepsilon_{g,p_1,lvq}$ and $\varepsilon_{g,p_1,bld}$ are the generalization errors achieved by a given LVQ algorithm 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.

In Fig. 2(b) we illustrate the divergent nature of **LVQ2.1**. If the prior probabilities are skewed the prototype corresponding to the class with lower probability diverges during the learning process and yields 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 differ 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 equal prior case. In high dimensions, this divergent behavior can also be observed if a window rule of the original formulation [20] is used [9], 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. 2(c) we see that the generalization error has a modality with respect to t, hence an optimal stopping point exists. In Fig. 3 we illustrate the performance of LVQ2.1. Fig. 3(a) shows the poor asymptotic behavior. Only for equal priors it achieves optimal performance. However, as depicted in Fig. 3(b), an idealized early stopping method as described above indeed gives near optimal behavior for the equal class variance case. However, the performance is worse when we deal with unequal class variances (Fig. 3(c)).

Fig. 4(a) shows the convergent behavior of **LVQ1**. As depicted in Fig. 4(b), LVQ1 gives a near optimal performance for equal class variances. For unequal class variance, the performance degrades, but it is still comparable with the performance of the best linear decision surface.

The dynamics of the **LFM** algorithm is shown in Fig. 5(a). We see that its performance is far from optimal in both equal (Fig. 5(b)) and unequal class variance (5(c)) cases. Hence, though LFM converges to a stable configuration of the prototypes, it fails to give a near optimal performance in terms of the generalization error. Note that we consider only the crisp

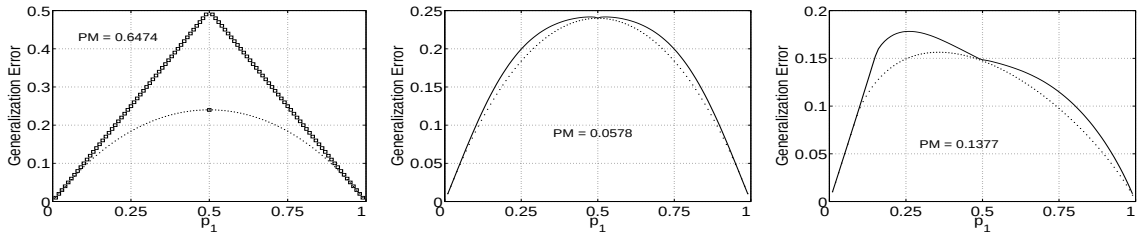

Figure 3: Performance of LVQ2.1: (a: left frame) Asymptotic behavior for $v_1 = v_{-1} = \lambda = 1$, note that for $p_1 = 0.5$ the performance is optimal, in all other cases $\varepsilon_g = \min(p_1, p_{-1})$. (b: middle frame) LVQ2.1 with stopping criterion when $v_1 = v_{-1} = \lambda = 1$, (c: right frame) LVQ2.1 with stopping criterion when $\lambda = 1, v_1 = 0.25, v_{-1} = 0.81$. The performance measure PM as given here and in the following figures is defined in Section 5.

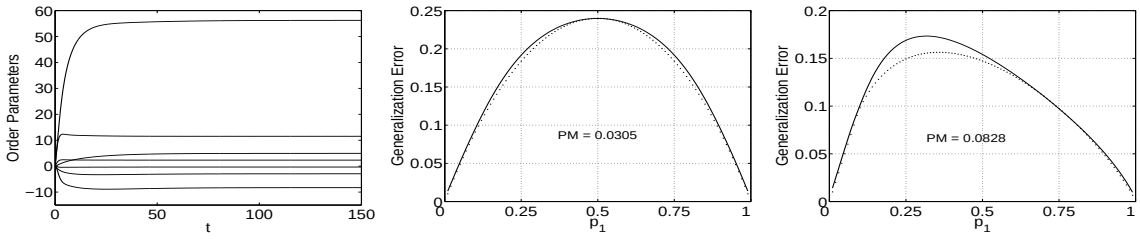

Figure 4: Performance of LVQ1: (a: left frame) Dynamics of the order parameters. (b: middle frame) Generalization with equal variance ($v_1 = v_{-1} = \lambda = 1$). (c: right frame) Generalization with unequal variance ($\lambda = 1, v_1 = 0.25, v_{-1} = 0.81$).

LFM procedure here. It is very well possible that *soft* realizations of RSLVQ as discussed in [19, 20] yield significantly better performance.

To facilitate a better understanding, we compare the performance of the algorithms in Fig. 6. In the first part, we see that LVQ1 outperforms the other algorithms for equal class variance, and it is closely followed by LVQ2.1 with early stopping. However the supremacy of LVQ1 is partly lost in the case of unequal class variance (see Fig. 6(b)) where an interval for p_1 exists for which the performance of LVQ2.1 with stopping criterion is better than LVQ1.

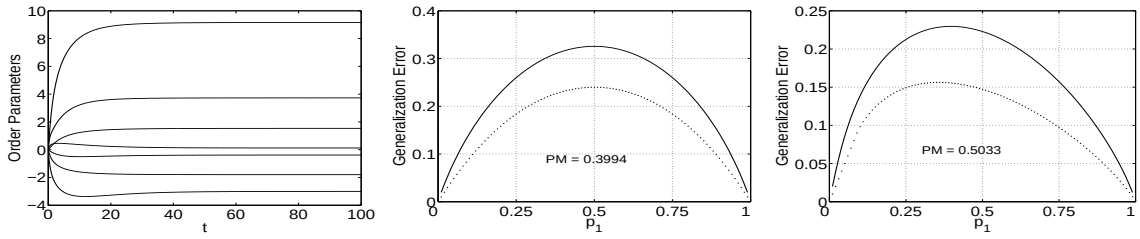

Figure 5: Performance of LFM: (a: left frame) Convergence. (b: middle frame) Generalization with equal variance ($v_1 = v_{-1} = \lambda = 1$) (c: right frame) Generalization with unequal variance ($\lambda = 1, v_1 = 0.25, v_{-1} = 0.81$).

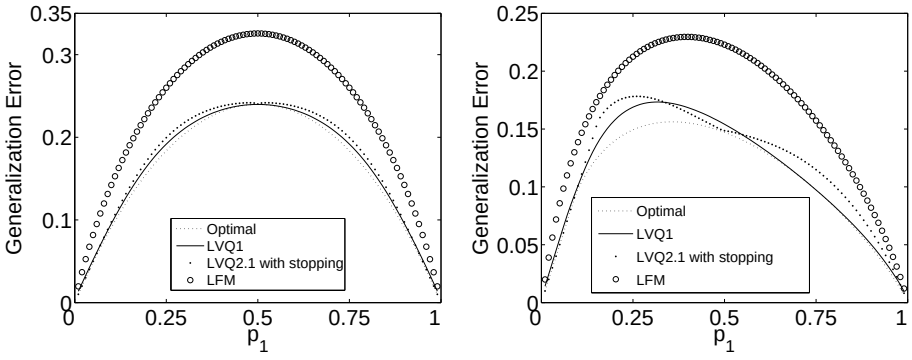

Figure 6: Comparison of performances of the algorithms: (a: left frame) equal class variance ($v_1 = v_{-1} = \lambda = 1$), (b: right frame) unequal class variance ($\lambda = 1, v_1 = 0.25, v_{-1} = 0.81$).

6 Conclusions

We have investigated different variants of LVQ algorithms in an exact mathematical way by means of the theory of on-line learning. For $N \to \infty$, the system dynamics can be described by a few characteristic quantities, and the generalization ability can be evaluated also for heuristic settings 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.
The behavior of LVQ2.1 is unstable and modifications such as a stopping rule become necessary. Surprisingly, fundamentally different limiting solutions are observed for the algorithms LVQ1, LVQ2.1, LFM, although their learning rules are quite similar. The generalization ability of the algorithms differs in particular for unbalanced class distributions. Even more convolved properties are revealed when the class variances differ. It is remarkable that the basic LVQ1 shows near optimal generalization error for all choices of the prior distribution in the equal class variance case. The LVQ2.1 algorithm with stopping criterion also performs close to optimal in the equal class variance. In the unequal class variance case LVQ2.1 with stopping criterion even outperforms the other algorithms for a range of p_1.
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 and in relevant (though simple) situations, which will be helpful in constructing efficient (in Bayesian sense) LVQ algorithms.

References

[1] M. Biehl and N. Caticha. The statistical mechanics of on-line learning and generalization. *In M.A. Arbib, The Handbook of Brain Theory and Neural Networks, MIT Press*, 2003.

[2] M. Biehl, A. Ghosh, and B. Hammer. The dynamics of learning vector quantization. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks, d-side*, pages 13–18.

[3] M. Cottrell, J. C. Fort, and G. Pages. Theoretical aspects of the S.O.M algorithm , survey. *Neuro-computing*, 21:119–138, 1998.

[4] K. Crammer, R. Gilad-Bachrach, A. Navot, and A. Tishby. Margin analysis of the LVQ algorithm. In *NIPS*. 2002.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern classification. *2e, New York: Wiley*, 2000.

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

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

[8] A. Freking, G. Reents, and M. Biehl. The dynamics of competitive learning. *Europhysics Letters 38*, pages 73–78, 1996.

[9] A. Ghosh, M. Biehl, A. Freking, and G. Reents. 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* `www.cs.rug.nl/~biehl`.

[10] B. Hammer, M. Strickert, and T. Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters*, 21(2):109–120, 2005.

[11] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks 15*, pages 1059–1068, 2002.

[12] T. Kohonen. Improved versions of learning vector quantization. *IJCNN, International Joint conference on Neural Networks, Vol. 1*, pages 545–550, 1990.

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

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

[15] 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*.

[16] G. Reents and R. Urbanczik. Self-averaging and on-line learning. *Physical review letters. Vol. 80. No. 24*, pages 5445–5448, 1998.

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

[18] A. S. Sato and K. Yamada. Generalized learning vector quantization. *In G. Tesauro, D. Touretzky and T. Leen, editors, Advances in Neural Information Processing Systems, Vol. 7*, pages 423–429, 1995.

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

[20] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation, 15*, pages 1589–1604, 2003.