LETTER Communicated by Youshen Xia

Window-Based Example Selection in Learning Vector Quantization

A. W. Witoelar
a.w.witoelar@rug.nl
Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen 9747 AG, Netherlands

A. Ghosh
aghosh@cs.tcd.ie
School of Computer Science and Statistics, Trinity College, Dublin 2, Ireland

J. J. G. de Vries
gj.de.vries@philips.com
Philips Research Europe, Eindhoven 5656 AE, Netherlands

B. Hammer
hammer@in.tu-clausthal.de
Center of Excellence Cognitive Interaction Technology, Bielefeld University, 33615 Bielefeld, Germany

M. Biehl
m.biehl@rug.nl
Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen 9747 AG, Netherlands

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

resulting decision boundary coincides with the optimal decision boundary, thus yielding optimal generalization ability.

1 Introduction

Learning vector quantization (LVQ) constitutes a family of learning algorithms for nearest prototype classification of potentially high-dimensional data (Kohonen, 2001). The intuitive approach and computational efficiency of LVQ classifiers have motivated its application in various disciplines (see, e.g., Neural Networks Research Centre, 2002). Prototypes in LVQ algorithms represent typical features within a data set using the same feature space instead of the black box approach practiced in many other classifiers (e.g., feedforward neural networks or support vector machines). This makes them attractive to researchers outside the field of machine learning. Other advantages of LVQ algorithms are that they are easy to implement for multiclass classification problems and the algorithm complexity can be adjusted during training as required.

Numerous variants of the original LVQ prescriptions have been proposed for achieving better performance, such as LVQ 2.1 (Kohonen, 1990, 2001), LVQ 3 (Kohonen, 1990, 2001), generalized LVQ (GLVQ) (Hammer & Villmann, 2002; Sato & Yamada, 1995), and Robust Soft LVQ (RSLVQ) (Seo & Obermayer, 2003). Common themes of these modifications include an additional parameter that controls the selection of data to which the system is adapted and variation of the magnitude of prototype updates. We refer to these in general as *window schemes*. In the limiting case of hard or crisp learning schemes, updates are restricted to examples that fall into this window. For instance, LVQ 2.1 allows updates as long as the example is in the vicinity of the current decision boundary. Alternatively, learning schemes can implement a soft window (e.g., RSLVQ and GLVQ), which considers all examples but adapts the magnitude of the update according to their relative distances to the current decision boundary.

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

In this work, we closely examine the influence of window schemes for LVQ algorithms. Typical learning behavior is studied within a model

situation of high-dimensional gaussian clusters and competing prototypes. From this analysis, we can observe typical learning curves and the convergence properties, that is, the asymptotic behavior in the limit of an arbitrarily large number of examples.

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

2 Model

Throughout this letter, we study LVQ algorithms in a model situation: high-dimensional data are generated from a mixture of M gaussian clusters and presented to a system of two or three prototypes. We restrict ourselves to the analysis of isotropic and homogeneous clusters; each cluster σ generates only data with one of the class labels $y_\sigma = \{1, 2, \ldots, N_c\}$ where N_c is the number of classes. Examples $\{\xi^\mu, y^\mu_\sigma\}$ with $\xi^\mu \in I\!R^N$ are drawn independently according to the probability density function,

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

$$P(\xi|\sigma) = \frac{1}{(2\pi v_\sigma)^{N/2}} \exp\left[-\frac{1}{2v_\sigma}(\xi - \ell_\sigma \mathbf{B}_\sigma)^2\right], \tag{2.1}$$

where p_σ are the cluster-wise prior probabilities and $\sum_\sigma p_\sigma = 1$. The components of vectors ξ^μ from cluster σ^μ are random numbers with mean vectors $\ell_\sigma \mathbf{B}_\sigma$ and variance v_σ. The unit vectors $\mathbf{B}_\sigma$ determine the orientation of cluster centers. Similar densities have been studied in Barkai, Seung, and Sompolinsky (1993), Biehl (1994), Biehl et al. (2007), and Meir (1995).

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

We bring attention to the scaling of the model. The anisotropy of this data distribution is very weak: while the mean of cluster σ, given by $\ell_\sigma \mathbf{B}_\sigma$, is a vector of length $\mathcal{O}(1)$, the average squared length of the data vectors $(\xi)^2$ is in the order $\mathcal{O}(N)$.

Obviously this model is greatly simplified from practical situations. However, it represents an ideal scenario to analyze the learning algorithms considered here using gaussian mixture models, a common technique in many practical scenarios. We can expect that algorithms that do not perform well on this idealized model will also be inappropriate for real-life problems. Although more complex behaviors are expected in practical applications, the nontrivial effects already observed in this model will clearly influence the outcome under more general circumstances.

3 Algorithms

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

$$\mathbf{w}_S^\mu = \mathbf{w}_S^{\mu-1} + \frac{\eta}{N} f_S\big[d_1^\mu, \ldots, d_K^\mu, y_\sigma^\mu, \ldots\big]\big(\xi^\mu - \mathbf{w}_S^{\mu-1}\big), \tag{3.1}$$

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

3.1 LVQ 2.1. LVQ 2.1, proposed by Kohonen, aims at efficient separation between prototypes of different classes and has been shown to provide good classification results (Kohonen, 1990; Neural Networks Research Centre, 2002). Given an example ξ^μ, two nearest prototypes $\mathbf{w}_S$ and $\mathbf{w}_T$ are updated if the following conditions are met: (1) the classes c_S and c_T are different, and (2) either c_S or c_T is equal to y_σ^μ. The prototype with the correct class is moved toward the data, while the other is moved further away with $f_S = 1$, $f_T = -1$ if $c_S = y_\sigma^\mu$; $f_S = -1$, $f_T = +1$ else.

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

$$\min\left(\frac{d_T^\mu}{d_S^\mu}, \frac{d_S^\mu}{d_T^\mu}\right) > \rho, \quad \text{with} \quad \rho = \frac{1-\omega}{1+\omega}, \tag{3.2}$$

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

$$\lim_{N\to\infty} \min\left(\frac{\left(\xi^\mu - \mathbf{w}_T^{\mu-1}\right)^2}{\left(\xi^\mu - \mathbf{w}_S^{\mu-1}\right)^2}, \frac{\left(\xi^\mu - \mathbf{w}_S^{\mu-1}\right)^2}{\left(\xi^\mu - \mathbf{w}_T^{\mu-1}\right)^2}\right) = 1, \tag{3.3}$$

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

$$\left|(\xi^\mu - \mathbf{w}_T)^2 - (\xi^\mu - \mathbf{w}_S)^2\right| \leq k \ \min((\xi^\mu - \mathbf{w}_S)^2, (\xi^\mu - \mathbf{w}_T)^2), \tag{3.4}$$

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

$$f_S = \chi(c_S, y_\sigma^\mu) \sum_{T:c_T\neq c_S} \left(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}\right) \prod_{U\neq S,T} \Theta_{SU}\Theta_{TU}, \tag{3.5}$$

with $\chi(c_S, y_\sigma^\mu) = 1$ if $c_S = y_\sigma^\mu$ and $\chi(c_S, y_\sigma^\mu) = -1$ else. We use the shorthand notations $\Theta_{ji}^\delta \equiv \Theta(d_i^\mu - d_j^\mu - \delta)$ and $\Theta_{ji} \equiv \Theta_{ji}^0 = \Theta(d_i^\mu - d_j^\mu)$, where $\Theta(x)$ is the Heaviside function $\Theta(x) = 1$ if $x > 0$; 0 else.

We sum over prototypes $\{\mathbf{w}_T | c_T \neq c_S\}$, and terms $(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}) = \Theta(d_T^\mu - d_S^\mu + \delta) - \Theta(d_T^\mu - d_S^\mu - \delta)$ enforce the window condition. The product term $\prod_{U\neq S,T} \Theta_{SU}\Theta_{TU}$ singles out instances where $\mathbf{w}_S$ and $\mathbf{w}_T$ are the two closest prototypes. This form of f_S allows for the analysis given in section 4.

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

The performance of LFM can be improved by including data selection using the window rule in equation 3.4. We refer to this algorithm as LFM with a window (LFM-W), represented by the modulation function

$$f_S = \begin{cases} \sum_{K:c_K \neq y_\sigma} \left(\Theta_{KS} - \Theta^{\delta}_{KS}\right) \psi(S, K) & \text{if } c_S = y^\mu_\sigma \\ \sum_{J:c_J = y_\sigma} \left(\Theta_{SJ} - \Theta^{\delta}_{SJ}\right) \psi(J, S) & \text{else} \end{cases}, \tag{3.6}$$

with $\psi(J, K) = \prod_{T:c_T = y_\sigma} \Theta_{JT} \prod_{U:c_U \neq y_\sigma} \Theta_{KU}$, which identifies cases with $\mathbf{w}_J$ being the correct winner and $\mathbf{w}_K$ being the incorrect winner: $\psi(J, K) = 1$ if this condition is fulfilled and $\psi(J, K) = 0$ else. Terms in parentheses single out misclassified examples that fall into the window.

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

$$E = \sum_\mu \Phi(\tau(\xi^\mu)) \quad \text{with} \quad \tau(\xi^\mu) = C \cdot \frac{d^\mu_J - d^\mu_K}{d^\mu_J + d^\mu_K}, \tag{3.7}$$

where $\Phi(\tau)$ is a (usually nonlinear) monotonically increasing function, $\mathbf{w}_J$ is the nearest correct prototype, and $\mathbf{w}_K$ is the nearest incorrect prototype to the example ξ^μ. We insert the scaling parameter C, which will be required for high dimensions. Stochastic gradient procedure on equation 3.7 yields the learning rule

$$f_J = 2C \frac{\partial \Phi(\tau)}{\partial \tau} \frac{d^\mu_K}{\left(d^\mu_J + d^\mu_K\right)^2}, \quad f_K = -2C \frac{\partial \Phi(\tau)}{\partial \tau} \frac{d^\mu_J}{\left(d^\mu_J + d^\mu_K\right)^2}. \tag{3.8}$$

Here the usefulness of selecting a nonlinear $\Phi(\tau)$ is shown. For instance, in Hammer and Villmann (2002) and Sato and Yamada (1995), the sigmoid function is chosen: $\Phi(\tau) = 1/(1 + \exp(-\tau))$. The form of $\partial\Phi(\tau)/\partial\tau$, which has a single peak at $\tau = 0$, can be interpreted as a soft window around the decision boundary.

In the high-dimensional limit, we notice that $(d^\mu_J + d^\mu_K)$ is dominated by $(\xi^\mu)^2$-terms and effectively becomes a constant $\mathcal{O}(N)$-term: $1/N(d^\mu_J + d^\mu_K) = 1 + \mathcal{O}(1/N)$. Therefore, the denominator term in equation 3.7 becomes constant:

$$\lim_{N\to\infty} E = \lim_{N\to\infty} \sum_\mu \Phi\left(\frac{C}{d^\mu_J + d^\mu_K}\left(d^\mu_J - d^\mu_K\right)\right) = \sum_\mu \Phi\left(\frac{1}{v_G}\left(d^\mu_J - d^\mu_K\right)\right). \tag{3.9}$$

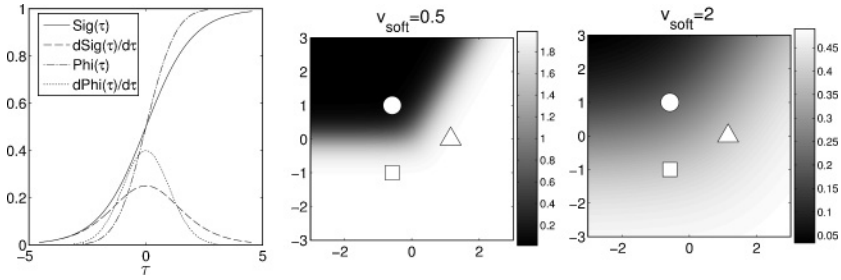

Figure 1: (Left) The form of the chosen $\Phi(\tau)$ in GLVQ, in comparison to the sigmoidal function $\mathrm{Sig}(\tau)$. The derivatives produce a soft window. (Middle and right) The RSLVQ modulation function f_S for class 1 (○) when presented with data from class 1. The figures display the difference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

To obtain a nonzero argument, C must also be in the order $\mathcal{O}(N)$, and we rescale using $v_G = (d_J^\mu + d_K^\mu)/C = \mathcal{O}(1)$. The parameter v_G determines the softness of the window, provided that an appropriate nonlinear $\Phi(\tau)$ is chosen. Note that GLVQ can be simplified to LVQ 2.1 without a window using the identity function $\Phi(\tau) = \tau$. The cost function in equation 3.7 becomes $E = \sum_\mu (d_J^\mu - d_K^\mu)/v_G$, where v_G could be set to 1 without changing its learning behavior. The modulation function is then reduced to $f_J = +1$, $f_K = -1$.

In this letter, we choose the cumulative normal distribution

$$\Phi(\tau) = \int_{-\infty}^{\tau} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt, \tag{3.10}$$

where $\partial\Phi(\tau)/\partial\tau = \phi(\tau) = (1/\sqrt{2\pi})\exp(-\tau^2/2)$. Note that this form implements a gaussian window similar to the sigmoidal cost described in Hammer and Villmann (2002) and Sato and Yamada (1995) and therefore produces a qualitatively similar behavior (see Figure 1 for the comparison).

Plugging in the form of equation 3.9, we obtain the learning rules

$$f_J = +\frac{2}{v_G}\,\phi\left(\frac{d_J - d_K}{v_G}\right), \qquad f_K = -\frac{2}{v_G}\,\phi\left(\frac{d_J - d_K}{v_G}\right). \tag{3.11}$$

We can write the modulation function as

$$f_S = \begin{cases} \displaystyle\sum_{K:c_K \neq y_\sigma} \left(\frac{2}{v_G}\phi\left(\frac{d_S - d_K}{v_G}\right)\right)\psi(S,K) & \text{if } c_S = y_\sigma^\mu \\ \displaystyle -\sum_{J:c_J = y_\sigma} \left(\frac{2}{v_G}\,\phi\left(\frac{d_J - d_S}{v_G}\right)\right)\psi(J,S) & \text{else} \end{cases}, \tag{3.12}$$

with $\psi(J,K) = \prod_{T:c_T = y_\sigma} \Theta_{JT} \prod_{U:c_U \neq y_\sigma} \Theta_{KU}$.

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

$$L = \prod_{\mu} \frac{p(\xi^{\mu}, y_{\sigma}^{\mu}|W)}{p(\xi^{\mu}|W)} \text{ with } \begin{cases} p(\xi^{\mu}, y_{\sigma}^{\mu}|W) = \displaystyle\sum_{\{S:c_S=y_{\sigma}^{\mu}\}}^{K} P_S\, p(\xi^{\mu}|S) \\ p(\xi^{\mu}|W) = \displaystyle\sum_{y_{\sigma}=1}^{N_c} \sum_{\{S:c_S=y_{\sigma}^{\mu}\}}^{K} P_S\, p(\xi^{\mu}|S) \end{cases}, \tag{3.13}$$

where $p(\xi^{\mu}|W)$ approximates the actual probability density $P(\xi)$ (see equation 2.1). It is assumed that every component $\mathbf{w}_S$ of the mixture generates examples that belong to one class: c_S. N_{σ} is the number of classes, and P_S is the probability that the examples are generated by a particular component $\mathbf{w}_S$ and $p(\xi^{\mu}|S)$ is the conditional probability that $\mathbf{w}_S$ generates a particular example ξ^{μ}.

The learning rule is obtained by performing stochastic gradient descent on the cost function E with respect to $\mathbf{w}_S$. We examine it for a gaussian mixture ansatz as in Seo and Obermayer (2003), where it is chosen $p(\xi^{\mu}|S) = (2\pi v_S)^{(N/2)} \exp(-d_S^{\mu}/2v_S)$. Furthermore, every component is assumed to have equal probability $P(S) = 1/K$, $\forall S$ and equal variance $v_S = v_{\text{soft}}$, $\forall S$, where v_{soft} is called the softness hyperparameter. This gives the following modulation function

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

with the assignment probabilities,

$$P_{\sigma}(S|\xi^{\mu}) = \frac{\exp\left(-d_S^{\mu}/2v_{\text{soft}}\right)}{\sum_{j:c_j=\sigma^{\mu}} \exp\left(-d_j^{\mu}/2v_{\text{soft}}\right)},$$
$$P(S|\xi^{\mu}) = \frac{\exp\left(-d_S^{\mu}/2v_{\text{soft}}\right)}{\sum_{j} \exp\left(-d_j^{\mu}/2v_{\text{soft}}\right)} \tag{3.14}$$

(see Seo & Obermayer, 2003, for the derivations). $P_{\sigma}(S|\xi^{\mu})$ describes the posterior probability that ξ^{μ} is assigned to the component S of the mixture, given that the example is generated by the correct class. $P(S|\xi^{\mu})$ describes the posterior probability that ξ^{μ} is assigned to the component S of the complete mixture using all classes. As v_{soft} becomes smaller, the updates

become smaller for correctly classified examples and larger for incorrectly classified examples (see Figure 1).

Note that the limiting case of v_{soft} is particularly simple. The assignments of equation 3.14 become hard assignments:

$$P_\sigma(S|\xi^\mu) = \begin{cases} 1, & \text{if } d_S^\mu = \min\limits_{\{j:c_j=\sigma^\mu\}} \{d_j^\mu\} \\ 0, & \text{else} \end{cases},$$

$$P(S|\xi^\mu) = \begin{cases} 1, & \text{if } d_S^\mu = \min\limits_{\{j\}} \{d_j^\mu\} \\ 0, & \text{else} \end{cases}. \tag{3.15}$$

Plugging the above into equation 3.14, we obtain the learning rule for learning from mistakes (LFM), described in section 3.2.

4 Analysis

In this section, we describe the methods to analyze the learning dynamics in LVQ algorithms. Following the lines of the theory of online learning (see, e.g., Biehl & Mietzner, 1993; Biehl & Schwarze, 1993; Engel & van den Broeck, 2001; Saad, 1999), the system can be fully described in terms of a few characteristic quantities, so-called order parameters, in the thermodynamic limit $N \to \infty$. A suitable set of order parameters for the considered learning model is

$$R_{S\sigma}^\mu = \mathbf{w}_S^\mu \cdot \mathbf{B}_\sigma \quad Q_{ST}^\mu = \mathbf{w}_S^\mu \cdot \mathbf{w}_T^\mu. \tag{4.1}$$

Note that $R_{S\sigma}$ are the projections of prototype vectors $\mathbf{w}_S^\mu$ on the center vectors $\mathbf{B}_\sigma$, and Q_{ST}^μ correspond to the self- and cross-overlaps of the prototype vectors. From the generic update rule defined above, equation 3.1, we can derive the following recursions in terms of the order parameters:

$$\begin{aligned} \frac{R_{S\sigma}^\mu - R_{S\sigma}^{\mu-1}}{1/N} &= \eta f_S\left(b_\sigma^\mu - R_{S\sigma}^{\mu-1}\right) \\ \frac{Q_{ST}^\mu - Q_{ST}^{\mu-1}}{1/N} &= \eta\left[f_T\left(h_S^\mu - Q_{ST}^{\mu-1}\right) + f_S\left(h_T^\mu - Q_{ST}^{\mu-1}\right)\right] \\ &\quad + \eta^2 \frac{f_S f_T (\xi^\mu)^2}{N} + \mathcal{O}\left(\frac{1}{N}\right), \end{aligned} \tag{4.2}$$

where the input data vectors ξ^μ enter the system as their projections h_S^μ and b_σ^μ, defined as

$$h_S^\mu = \mathbf{w}_S^{\mu-1} \cdot \xi^\mu \quad b_\sigma^\mu = \mathbf{B}_\sigma \cdot \xi^\mu. \tag{4.3}$$

In the limit $N \to \infty$, the $\mathcal{O}(1/N)$ term can be neglected and the order parameters self-average (Reents & Urbanczik, 1998) with respect to the random sequence of examples. This means that fluctuations of the order parameters vanish, and the system dynamics can be described exactly in terms of their mean values. Also for $N \to \infty$, the rescaled quantity $\alpha \equiv \mu/N$ can be conceived as a continuous time variable. Accordingly, the dynamics can be described by a set of coupled ordinary differential equations (ODE) (Ghosh, Biehl, & Hammer, 2006) after performing an average over the sequence of input data:

$$\begin{aligned}\frac{dR_{S\sigma}}{d\alpha} &= \eta(\langle b_\sigma f_S\rangle - \langle f_S\rangle R_{S\sigma})\\ \frac{dQ_{ST}}{d\alpha} &= \eta(\langle h_S f_T\rangle - \langle f_T\rangle Q_{ST} + \langle h_T f_S\rangle - \langle f_S\rangle Q_{ST})\\ &\quad + \eta^2 \sum_\sigma p_\sigma \upsilon_\sigma \langle f_S f_T\rangle_\sigma, \end{aligned} \tag{4.4}$$

where $\langle . \rangle$ and $\langle . \rangle_\sigma$ are the averages over the density $P(\xi)$ and $P(\xi|\sigma)$. To simplify the last term of equation 4.4, we used

$$\begin{aligned}\lim_{N\to\infty} \langle f_S f_T \xi^2\rangle/N &= \lim_{N\to\infty} \sum_\sigma p_\sigma(\upsilon_\sigma N + \ell^2)\langle f_S f_T\rangle_\sigma/N\\ &= \sum_\sigma p_\sigma \upsilon_\sigma \langle f_S f_T\rangle_\sigma.\end{aligned}$$

In various sections in this letter, we investigate learning behaviors using small learning rates $\eta \to 0$ and neglect the η^2 terms in equation 4.4. Nontrivial behavior is expected only by taking the simultaneous limit $\eta \to 0, \alpha \to \infty$ and rescaling $\widetilde{\alpha} = \eta\alpha$ in equation 4.4.

Exploiting the limit $N \to \infty$ once more, the quantities h_S^μ, b_σ^μ become correlated gaussian quantities by means of the central limit theorem. Therefore, they are fully specified by first and second moments, detailed in appendix A:

$$\begin{aligned}&\left\langle h_S^\mu \right\rangle_\sigma = \ell_\sigma R_{S\sigma}^{\mu-1}, \quad \left\langle b_\tau^\mu \right\rangle_\sigma = \ell_\sigma \delta_{\tau\sigma}, \left\langle h_S^\mu h_T^\mu \right\rangle_\sigma - \left\langle h_S^\mu \right\rangle_\sigma \left\langle h_T^\mu \right\rangle_\sigma = \upsilon_\sigma Q_{ST}^{\mu-1}\\ &\left\langle b_\tau^\mu b_\rho^\mu \right\rangle_\sigma - \left\langle b_\tau^\mu \right\rangle_\sigma \left\langle b_\rho^\mu \right\rangle_\sigma = \upsilon_\sigma \mathbf{T}_{\tau\rho}, \quad \left\langle h_i^\mu b_\tau^\mu \right\rangle_\sigma - \left\langle h_i^\mu \right\rangle_\sigma \left\langle b_\tau^\mu \right\rangle_\sigma = \upsilon_\sigma R_{i\tau}^{\mu-1}, \end{aligned} \tag{4.5}$$

where S, T are prototype indices; τ, ρ, σ are cluster indices; δ is the Kronecker delta; and $\mathbf{T}_{\tau\rho} \equiv \mathbf{B}_\tau \cdot \mathbf{B}_\rho$ is an overlap measure between clusters.

Thus, the above averages $\langle f_S\rangle$, $\langle h_T f_S\rangle$, and $\langle b_T f_S\rangle$ reduce to gaussian integrations in $K + M$ dimensions and can be expressed in $\{R_{S\sigma}, Q_{ST}\}$ (see appendix B). For various algorithms and a system with two competing prototypes, the averages can be calculated analytically. For three or more prototypes, the mathematical treatment becomes more involved and requires multiple numerical integrations.

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

$$\epsilon_g = \sum_{\sigma=1}^{M} p_\sigma \epsilon_{g,\sigma} \text{ with } \epsilon_{g,\sigma} = \sum_{S:c_S \neq y_\sigma}^{K} \left\langle \prod_{T\neq S}^{K} \Theta_{ST} \right\rangle_\sigma , \tag{4.6}$$

which can be calculated from $\{R_{i\sigma}(\alpha),\ Q_{ij}(\alpha)\}$. For instance, for the simplest system with two clusters $\sigma = \{+,-\}$ and prototypes $\mathbf{w}_+$ and $\mathbf{w}_-$, the generalization error is written explicitly in terms of order parameters as

$$\epsilon_{g,\sigma} = \Phi\left(\frac{Q_{\sigma\sigma} - Q_{-\sigma,-\sigma} - 2\ell_\sigma (R_{\sigma,\sigma} - R_{-\sigma,\sigma})}{2\sqrt{v_\sigma}\sqrt{Q_{\sigma\sigma} - 2Q_{\sigma,-\sigma} + Q_{-\sigma,-\sigma}}} \right), \tag{4.7}$$

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

5 A Simple Case: Two Prototypes, Two Clusters

In this section, we discuss in detail the results of the analysis for the simplest nontrivial problem: two-prototype LVQ 2.1, GLVQ, LFM-W, and RSLVQ systems and $M = 2$ with one gaussian cluster per class. The model data are given in section 2. For simplicity, we denote the two clusters as $\sigma = \{+,-\}$ and without loss of generality can choose $\ell_+ = \ell_- = \ell$ and orthonormal $\mathbf{B}_\sigma$, that is, $\mathbf{B}_i \cdot \mathbf{B}_j = 1$ if $i = j$; 0 else.

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

In this simple scenario, it is possible to exactly calculate the best linear decision boundaries (BLD) by linear approximation of the Bayesian optimal decision boundary (see Biehl, Freking, Ghosh, & Reents, 2004, for

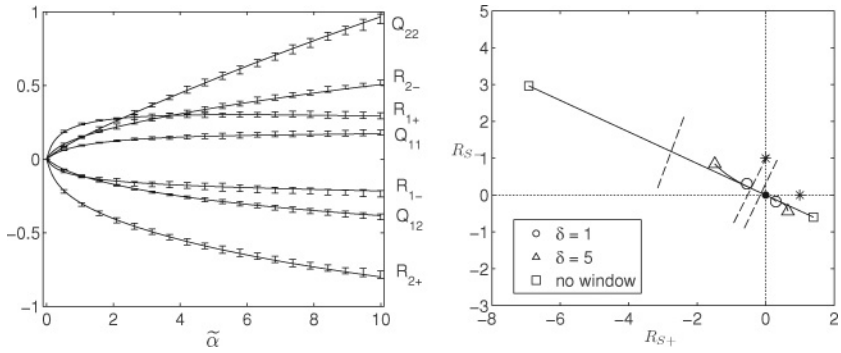

Figure 2: (Left) Evolution of the order parameters $\{R_{S\sigma}, Q_{ST}\}$ for LVQ 2.1 with $K = 2$, $M = 2$, $\ell_1 = \ell_2 = 1$, $p_1 = 0.7$, $v_1 = v_2 = 1$ and learning parameters $\eta = 0.1$ and $\delta = 1$. Solid lines represent $\{R_{S\sigma}, Q_{ST}\}$ obtained from the theoretical analysis,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

the calculations.) We compare the results from each algorithm to the best linearly achievable error ϵ_g^{bld}.

5.1 LVQ 2.1. We first examine two-prototype systems: $K = 2$. Figure 2 illustrates the evolution of order parameters under the influence of a window and the trajectories of the prototypes projected onto the $(\mathbf{B}_+, \mathbf{B}_-)$ subspace. Without additional constraints, LVQ 2.1 with two prototypes displays a strong divergent behavior in a system with unbalanced data: $p_+ \neq p_-$. The repulsion factor dominates for the prototype representing the weaker cluster, here $\mathbf{w}_2$. The order parameters associated with this prototype increase exponentially with $\widetilde{\alpha}$. As $\widetilde{\alpha} \to \infty$, $\mathbf{w}_2$ will be arbitrarily far away from the cluster centers, and the asymptotic generalization error is trivial, $\epsilon_g(\infty) = \min(p_+, p_-)$.

When the window scheme is implemented, $\mathbf{w}_2$ is repulsed until the data densities of both classes within the window become more balanced. Subsequently the order parameters change with more balance between both prototypes. The repulsion factor still dominates its counterpart; therefore, both prototypes still diverge: $R_{S\sigma}$ for both prototypes display a linear change with $\widetilde{\alpha}$ at large $\widetilde{\alpha}$, but the decision boundary remains stable. Trivial classification is prevented (see the generalization error curves ϵ_g versus $\widetilde{\alpha}$ in the left panel of Figure 3). Obviously, for smaller δ, a considerable amount of data is filtered out, and the initial learning stages slow significantly.

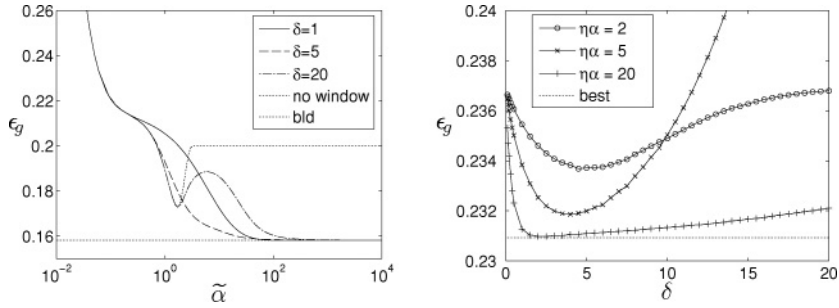

Figure 3: Generalization error ϵ_g for LVQ 2.1 with $K = 2$, $M = 2$, $\ell = 1$, $p_+ = 0.8$, $p_- = 0.2$, and $\eta \to 0$. (Left) ϵ_g versus $\widetilde{\alpha}$ using $\delta = 1, 5, 20$ and without a window. Note the logarithmic scaling on the horizontal axis. The asymptotic errors for all settings of δ converge at ϵ_g^{bld}, indicated by the dotted line. (Right) ϵ_g at fixed learning times $\widetilde{\alpha} = 2, 5$, and 20 as a function of δ.

Meanwhile for large δ, ϵ_g becomes nonmonotonic and converges more slowly.

Hence, the performance at finite $\widetilde{\alpha}$ is dependent on δ, displayed in Figure 3, and parameter settings are highly critical in practical applications. Given learning time $\widetilde{\alpha}$, an optimal choice of fixed δ exists, which clearly depends on the properties of the data. With larger $\widetilde{\alpha}$, ϵ_g becomes less sensitive toward δ, and the optimal setting of δ is smaller. Surprisingly, δ influences only the convergence speed, while the nontrivial asymptotic generalization error $\epsilon_g(\infty)$ is insensitive to the choice of δ and equals the best achievable error ϵ_g^{bld} for each setting. This can be explained as follows. We can compare the asymptotic decision boundary to the BLD: the angle between them is equal to the angle between $(\mathbf{w}_1 - \mathbf{w}_2)$ and $(\mathbf{B}_+ - \mathbf{B}_-)$. This is calculated, using equation 4.1 and the orthonormality of $\mathbf{B}_+$ and $\mathbf{B}_-$, as

$$\varphi = \arccos\left(\frac{R_{1+} + R_{1-} - R_{2+} + R_{2-}}{\sqrt{2}\,(Q_{11} - 2Q_{12} + Q_{22})}\right), \tag{5.1}$$

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

5.2 LFM-W. The LFM scheme performs updates identical to LVQ 2.1 with the condition that the example is misclassified. A detailed investigation into the characteristics of $K = 2$ unrestricted LFM has been presented in

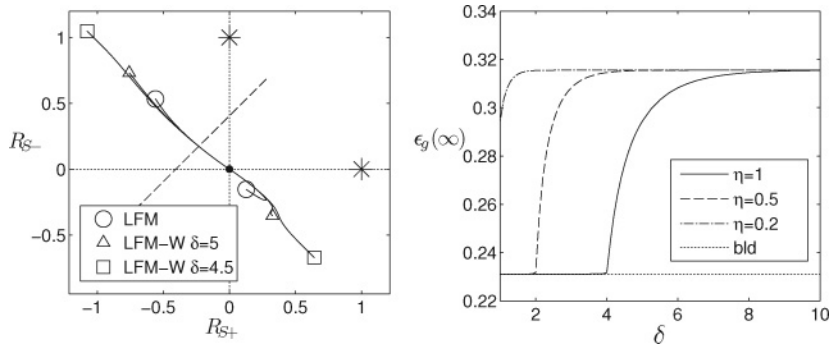

Figure 4: LFM-W with $p_+ = 0.6$, $\ell = 1$, $v_+ = v_- = 1$. (Left) Asymptotic prototype configuration for LFM and LFM-W $\delta = 5$ and 4.5, projected on $\{\mathbf{B}_+, \mathbf{B}_-\}$. Cluster centers $\ell\mathbf{B}_+$, $\ell\mathbf{B}_-$ are indicated by $*$. The projection of $\mathbf{w}_1$, $\mathbf{w}_2$ lies parallel to the symmetry axis $\ell(\mathbf{B}_+ - \mathbf{B}-)$, although they retain components orthogonal to the $\{\mathbf{B}_+, \mathbf{B}_-\}$ subspace. (Right) $\epsilon_g(\infty)$ as a function of the window size δ. The lines correspond to learning rates $\eta = 0.2, 0.5$, and 1.0.

Biehl et al. (2007). There, it was shown that LFM produces stable prototype configurations for finite learning rates η. The projection of the prototypes lies parallel to the symmetry axis $\ell(\mathbf{B}_+ - \mathbf{B}-)$, displayed in Figure 4. However the prototypes $\mathbf{w}_1$ and $\mathbf{w}_2$ retain components orthogonal to the two-dimensional subspace spanned by the cluster centers, indicated by $Q_{ST} > R_{S+}R_{T+} + R_{S-}R_{T-}$, which implies

$$|\mathbf{w}_S|^2 > |R_{S+}\mathbf{B}_+ + R_{S-}\mathbf{B}_-|^2.$$

The asymptotic generalization error $\epsilon_g(\infty)$ is suboptimal and insensitive to η: the asymptotic decision boundary remains at an angle φ from the optimal hyperplane (see equation 5.1) independent of η. The Euclidean distance between prototypes is given by the quantity

$$\Delta_q = \sqrt{(\mathbf{w}_1 - \mathbf{w}_2)^2} = \sqrt{Q_{11} - 2Q_{12} + Q_{22}}\,, \tag{5.2}$$

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

In this analysis, we observe that window schemes can dramatically improve the performance of LFM. When a window is used, the tilt of the decision boundary from the optimal hyperplane (φ in equation 5.1), is reduced, resulting in lower $\epsilon_g(\infty)$. We observe that $\epsilon_g(\infty)$ decreases along with reducing δ, displayed in the right panel of Figure 4. However, a critical window size δ_c exists where the LFM unexpectedly becomes divergent and no stationary state exists. Smaller windows filter examples, which produce

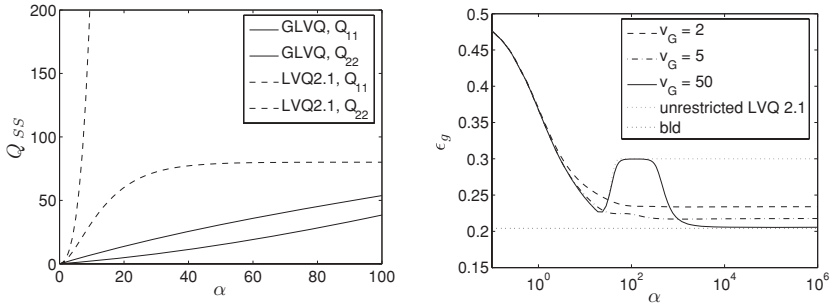

Figure 5: (Left) Q_{11} and Q_{22} for GLVQ (solid lines), compared to unrestricted LVQ 2.1 (dashed lines). The soft window of GLVQ slows the repulsion of one prototype, but the prototypes remain divergent. Here $p_+ = 0.7$, $\ell = 1$, $v_+ = 2$, $v_- = 5$, $\eta = 0.25$. (Right) Learning curves ϵ_g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

more repulsion in the orientation of the cluster centers, and we observe asymptotically larger Δ_q as δ decreases. This is clearly observed in Figure 4. Given a sufficiently small δ, it is possible that the repulsion factor entirely outweighs the attractive factor. At $\delta < \delta_c$, it performs similar to LVQ 2.1: the angle φ becomes zero, and $\epsilon_g(\infty)$ is close to the best achievable error.

Unlike the unrestricted case, the learning rate η can influence the asymptotic performance. The learning rate and window size are indirectly related, as shown in the right panel of Figure 4. For example, learning with small learning rates requires smaller windows to achieve optimal asymptotic error. Note that the influence of the window size depends heavily on the structure of the data. For various data models, efficient window settings may exist on only a very limited range, and window schemes may be ineffective to improve generalization performance while still maintaining stability.

5.3 GLVQ. Apart from the influence of v_G to the overall learning rate, small v_G corresponds to a sharp peak around the decision boundary, while large v_G corresponds to a very large window. Figure 5 displays the prototype lengths while using GLVQ: the soft window slows the strong repulsion of the prototype of the weaker cluster, as opposed to unrestricted LVQ 2.1. While both prototypes still diverge because the cost function at $N \to \infty$ is not bounded (see equation 3.9), the asymptotic ϵ_g remains nontrivial (see Figure 5).

Note that v_G directly relates to the overall learning rate η/v_G (refer to equation 3.12), which influences the level of noise in stochastic gradient procedures. We compare results with respect to v_G, while maintaining an

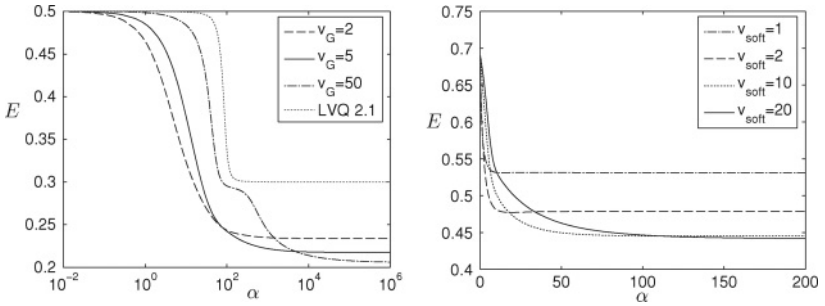

Figure 6: $p_+ = 0.7, \ell = 1, v_+ = 1, v_- = 1$. The cost functions for GLVQ with $\eta \to 0$ (left) and RSLVQ with $\eta/v_{\text{soft}} = 1$ (right) decrease monotonically, corresponding to a stochastic gradient descent.

equal overall learning rate by keeping η/v_G constant in Figure 5. Performance deteriorates at smaller v_G, where training slows down at intermediate stages and converges at a higher error. However, very large v_G allows strong repulsion of the weaker prototype, which results in nonmonotonic ϵ_g and long learning convergence times. Surprisingly, the soft GLVQ window is outperformed by the simple hard or crisp window of LVQ 2.1. This is caused by the long tail of the modulation function, which sums up into a large repulsion, whereas in the crisp window, only data near the decision boundary are considered.

Figure 6 displays the cost function during learning. In the initial learning stages, the minimization of the cost function E leads to fast decrease of ϵ_g. However, while the cost function continues to decrease monotonically, ϵ_g behaves nonmonotonically. While many techniques are developed to improve minimization procedures of E, it is important to evaluate the choice of E and its correlation to the desired generalization performance.

5.4 RSLVQ. Finally in this section, we study the influence of the softness parameter v_{soft} in the RSLVQ algorithm. Note that in Seo and Obermayer (2003), the learning rate η and softness parameter v_{soft} are treated independently using separate annealing schedules. In this section, we assume η decreases proportionally with v_{soft}, that is, a fixed overall learning rate η/v_{soft} is maintained.

We first investigate model scenarios with equal variance clusters $v_+ = v_-$ and unbalanced data $p_+ \neq p_-$. We observe the influence of v_{soft} on the learning curves, displayed on the left panel of Figure 7. The generalization error curve depends on v_{soft}: at large v_{soft}, ϵ_g may exhibit nonmonotonic behavior, reminiscent of LVQ 2.1. Because of this behavior, the learning process may require long learning times before reaching the asymptotic configuration. This is an important consideration for practical applications, which often

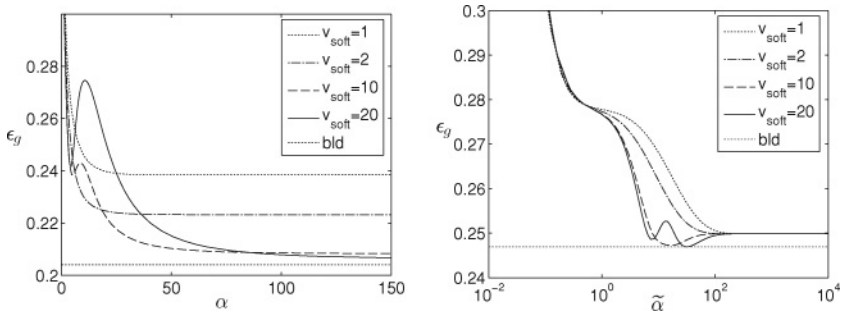

Figure 7: Learning curves ϵ_g for RSLVQ using softness parameter $v_{\text{soft}} = 1, 2, 10$, and 20. (Left) $p_+ = 0.7$ and equal variance $v_+ = v_- = 1$ with fixed overall learning rate $\eta/v_{\text{soft}} = 1$. (Right) $p_+ = 0.6$ and unequal variance $v_+ = 1$, $v_- = 4$ with $\eta/v_{\text{soft}} \to 0$. The asymptotic errors are independent of v_{soft} at small learning rates, but at a suboptimal value. Note the logarithmic scale of $\widetilde{\alpha}$.

uses early stopping strategies to avoid overtraining. Meanwhile, the algorithm minimizes the cost function E in equation 3.13 monotonically (see Figure 6). Thus, the decrease in E does not always result in a decrease of ϵ_g.

A major advantage of the RSLVQ algorithm is the convergence of prototypes: a stationary configuration of order parameters exists for finite v_{soft}. The asymptotic configuration of prototypes is displayed in Figure 8. At $\widetilde{\alpha} \to \infty$, the softness parameter controls only the distance between the two prototypes: Δ_q as defined in equation 5.2 decreases linearly with v_{soft}. Note that under the conditions $p_+ = 0.5$, $v_{\text{soft}} = v_+ = v_-$, and initialization of prototypes on the symmetry axis, each prototype is located at its corresponding cluster center: the RSLVQ mixture model exactly matches the actual input density.

Figures 7 compare the asymptotic errors in the case of $\eta/v_{\text{soft}} = 1$ (left) and small learning rates $\eta/v_{\text{soft}} \to 0$ (right). In the former case, performance improves with large v_{soft}: at small v_{soft}, the system converges at high ϵ_g similar to LFM, while at larger v_{soft}, it approaches the best linear decision. Meanwhile, at small learning rates, the asymptotic error becomes independent of v_{soft}. Therefore, given sufficiently small learning rates, RSLVQ becomes robust with regard to its softness parameter.

In the equal variance scenario, the asymptotic decision boundary always converges to the best linear decision boundary for all settings of $\{p_+, p_-\}$, and RSLVQ outperforms both LFM and LVQ 2.1, as it provides robustness, stability, and low generalization error.

On the other hand, a scenario with unequal class variances presents an interesting case where RSLVQ with global v_{soft} fails to match the model. RSLVQ remains robust; the decision boundary converges to identical configurations for all settings of v_{soft} (see Figure 8). However, the asymptotic results are suboptimal. While RSLVQ is insensitive to the priors of the

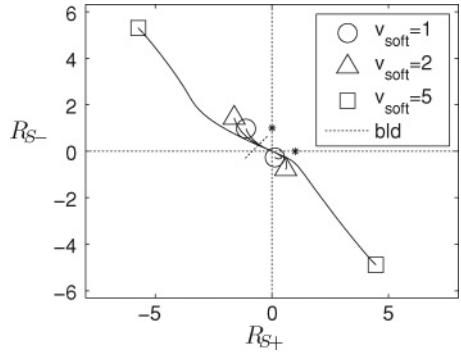

Figure 8: Trajectories of prototypes of the system in the left panel of Figure 7. Prototypes are projected on the space Span($\mathbf{B}_+, \mathbf{B}_-$) for $v_{\text{soft}} = 1$ (circle), 2 (triangle), and 5 (square).

clusters, its performance with regard to the best achievable error is sensitive to the cluster variances; for example, at highly unbalanced σ_+/σ_-, RSLVQ generalizes poorly and is outperformed by the simpler LVQ 2.1. In practical applications, v_{soft} may be set locally for each prototype to accommodate such scenarios, but this case cannot be treated along the lines of the analysis here in a straightforward way.

6 Optimal Window Schedules

We have observed in sections 5.1 and 5.2 the learning curves and asymptotics of LVQ 2.1 and LFM-W with regard to fixed window parameters. Although small windows allow optimal $\epsilon_g(\alpha \to \infty)$, their obvious disadvantages are slower initial learning and convergence speed. This suggests that online performance can be improved by adjusting the window along with the number of examples presented. In this section, we treat the window parameter as dynamic properties during learning, $\delta(\alpha)$, similar to the annealing method in Seo and Obermayer (2006) or the gradient-based optimization in Bengio (2000). In contrast to the proposed methods, we can formally minimize $d\epsilon_g(\alpha)/d\alpha$ with respect to δ using the knowledge of the input density. Hence we calculate the locally optimal $\delta^*(\alpha)$-schedule by finding the condition

$$\delta^*(\alpha) = \arg\min_{\delta} \left(\mathbf{u}(\alpha) \cdot \frac{d\mathbf{O}(\alpha)}{d\alpha} \right) = 0 \;\text{ with }\; \mathbf{u}(\alpha) = \sum_{\sigma=1}^{M} p_\sigma \frac{d\epsilon_{g,\sigma}(\alpha)}{d\mathbf{O}}, \tag{6.1}$$

where we use the shorthand $\mathbf{O}$ for the set of order parameters. For a system with two prototypes $\{\mathbf{w}_+.\mathbf{w}_-\}$ and two clusters $\sigma = \{+, -\}$,

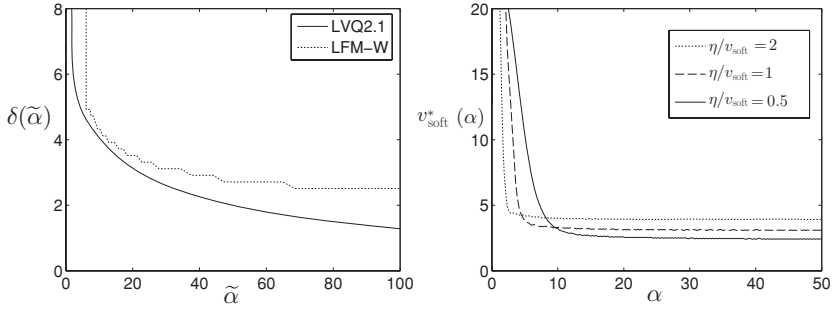

Figure 9: (Left) Optimal window schedule $\delta(\alpha)$ for LVQ 2.1 and LFM, obtained by formally minimizing $d\epsilon_g/d\alpha$ with respect to $\delta(\alpha)$. (Right) Optimal softness parameter for RSLVQ with fixed $\eta/v_{\rm soft} = 2$, 1, and 0.5.

$\mathbf{O} = \{R_{++}, R_{+-}, R_{-+}, R_{--}, Q_{++}, Q_{+-}, Q_{--}\}^T$ and derivating from equation 4.7, we obtain

$$\frac{d\epsilon_{g_\sigma}(\alpha)}{d\mathbf{O}} = \frac{1}{2\sqrt{v_\sigma}\Delta_q}\,\phi\left(\frac{Z_\sigma}{2\sqrt{v_\sigma}\Delta_q}\right).\mathbf{A}_\sigma \quad \text{with}$$

$$\mathbf{A}_+ = \begin{bmatrix} -2\ell \\ 0 \\ +2\ell \\ 0 \\ 1 - Z_+/\left(2\Delta_q^2\right) \\ Z_+/\Delta_q^2 \\ -1 - Z_+/\left(2\Delta_q^2\right) \end{bmatrix}, \quad \mathbf{A}_- = \begin{bmatrix} 0 \\ +2\ell \\ 0 \\ -2\ell \\ -1 - Z_-/\left(2\Delta_q^2\right) \\ Z_-/\Delta_q^2 \\ 1 - Z_-/\left(2\Delta_q^2\right) \end{bmatrix},$$

with $Z_\sigma = Q_{\sigma\sigma} - Q_{-\sigma,-\sigma} - 2\ell(R_{\sigma\sigma} - R_{-\sigma\sigma})$ and Δ_q defined in equation 5.2 (see appendix D for the calculations).

We plug in $d\mathbf{O}/d\alpha$ for the corresponding algorithm and numerically calculate $\delta^*(\alpha)$ from equation 6.1 at each learning step. We find that the learning curve is improved with initially large δ, which is decreased during training, following the curve in Figure 9. This suggests that practical schedules with gradual reduction of window sizes are indeed suitable for this particular learning problem.

While this approach locally minimizes generalization error, this strategy does not always lead to minimization of ϵ_g over a time span (i.e., a globally optimal schedule), which requires calculations along the lines of variational optimization (see Biehl, 1994; Saad & Rattray, 1997) for its application of optimal learning rates in multilayered neural networks. Obviously a priori knowledge of the input density is not available in practical

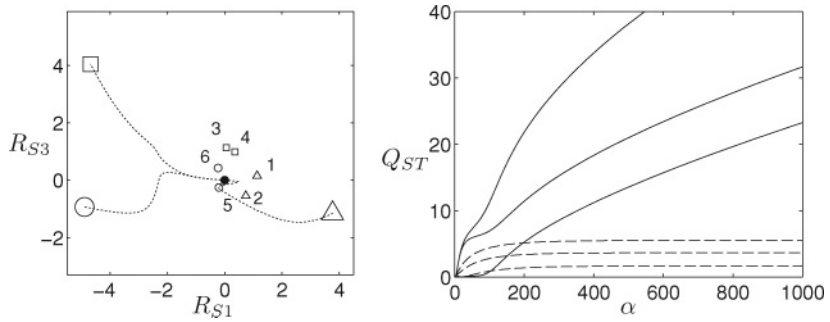

Figure 10: (Left) Snapshot at $\alpha = 50$ of an LVQ 2.1 system, $\delta = 1$ with $K = 3$ and $M = 6$ randomly generated isotropic clusters projected on the $(\mathbf{B}_1, \mathbf{B}_3)$ subspace. The solid dot marks the initial position of all prototypes and solid lines mark the trajectories of the prototypes. (Right) $p_\triangle = 0.5$, $p_\square = 0.3$, $p_\circ = 0.2$. Solid lines represent, from bottom to top, prototype vector lengths Q_{11}, Q_{22}, Q_{33} for LVQ2.1 $\delta = 10$. Dashed lines represent the result for RSLVQ $v_{\text{soft}} = 2$.

situations. Nevertheless, this minimization technique provides an upper bound of the achievable performance of the learning scheme for a given model.

Figure 7 displays that although large v_{soft} for RSLVQ allows a faster initial learning, it also can yield nonmonotonic learning curves. We can avoid the nonmonotonic behavior and maximize the decrease of ϵ_g by applying a variational approach analogous to equation 6.1 in order to calculate the locally optimal softness parameter schedule $v^*_{\text{soft}}(\alpha)$. While fixing the value of η/v_{soft}, we produce the locally optimal softness schedule $v^*_{\text{soft}}(\alpha)$ in Figure 9, where $v^*_{\text{soft}}(\alpha)$ is initially large and decreases to saturate at a constant value. Note that this value depends on the learning rate; for example, it decreases with η/v_{soft}. In calculations with $\eta \to 0$, we obtain the limit $v^*_{\text{soft}}(\infty) \to 0$, which is the clearly suboptimal LFM. Therefore, an analysis of optimal RSLVQ schedule requires $\eta > 0$.

7 Three-Prototype Systems

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

We first look at multiclass problems with $N_c = 3$ classes. An example is shown in Figure 10 for LVQ 2.1 with $M = 6$ clusters selected with random variances and random deviation from the original class centers. The clusters are separable only in M out of N dimensions. In all our

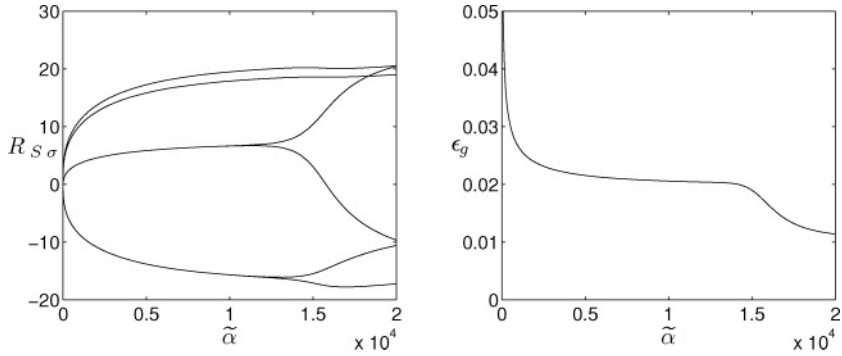

Figure 11: Unspecialized phase induces long learning plateaus, shown with LFM-W $K = 3, c_K = \{\pm 1\}$ and input density $M = 6$ and $N_c = 2, c_\sigma = \{\pm 1\}$. (Left) Several order parameters display a specialization phase between prototypes of the same class. (Right) Generalization error.

observations, we find that the behaviors of $K = 3$ systems are qualitatively similar to $K = 2$ systems. For LVQ 2.1, the learning curves vary according to the window sizes, but its asymptotic generalization error is independent of δ. Due to the presence of other prototypes, the repulsion on a weaker class prototype is reduced. However, the prototypes remain divergent (e.g., Figure 10). Meanwhile for LFM-W, the asymptotic performance is sensitive to δ whose range of effective window sizes depends strongly on the learning parameters. For GLVQ, the prototypes are divergent with a higher asymptotic error than LVQ 2.1, and thus it performs poorly. Finally, for RSLVQ, the prototypes remain stable, and the asymptotic generalization performance is robust with regard to settings of v_{soft}, but it is outperformed by LVQ 2.1. Hence, the results are consistent with the $K = 2$ system, and the preceding analysis is valid qualitatively to at least systems of M clusters and one prototype per class within the model restrictions.

To allow more complex decision boundaries, practical LVQ applications frequently employ several prototypes within a class. We investigate a two-class system $N_c = 2$, $y_\sigma = \{+, -\}$ using $K = 3$ prototypes with labels $c_S = \{+, +, -\}$ and observe the nontrivial interaction between similarly labeled prototypes, here $\mathbf{w}_1$ and $\mathbf{w}_2$. While prototypes of different classes immediately separate in the initial training phase, prototypes of the same class remain identical in the M-dimensional space (see Figure 11). The latter prototypes differ only in dimensions that are not related for classification and produce a suboptimal decision boundary. This may proceed for a long learning period before these prototypes begin to specialize; each prototype produces a bigger overlap $R_{S\sigma}$ with a distinct group of clusters. The specialization phase produces a sudden decrease of ϵ_g, displayed in the right panel of Figure 11. This phenomenon is highly reminiscent of symmetry-breaking

Table 1: Asymptotic Properties of LVQ Algorithms.

	LVQ 2.1	LFM-W	GLVQ	RSLVQ
Stability	Divergent	Convergent[a]	Divergent	Convergent
Sensitivity with regard to parameters	Robust	Dependent	Dependent	Robust
Generalization ability	Optimal	Suboptimal	Suboptimal	Suboptimal

[a]Under the condition that δ is larger than critical window size δ_c.

effects observed in unsupervised learning, such as winner-takes-all vector quantization (VQ) (Biehl, 1994; Witoelar et al., 2008) or multilayer neural networks (Saad & Solla, 1995).

Learning parameters highly influence the nature of the transition; for example, large learning rates and smaller windows prolong the unspecialized phase, and therefore they are critical to the success of learning. Symmetry breaking may require exceedingly long learning times, resulting in learning plateaus that dominate the training process and present a challenge in practical situations with very high-dimensional data. In more extreme circumstances, the system may not escape the unspecialized state at all, and the optimal classification cannot be obtained. Details of the symmetry-breaking properties with regard to parameters will be investigated in future publications.

8 Conclusion

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

Our analysis demonstrates the influence of windows on the learning curves and the advantages and drawbacks of each algorithm within the model scenarios. A summary is described in Table 1. Asymptotically LVQ 2.1 achieves optimal performance in all scenarios, but stability remains an issue in terms of diverging prototypes. LFM-W shows a remarkable improvement in performance over LFM. Unfortunately, the introduction of a window may also influence its stability, and therefore it is highly parameter sensitive; only a narrow range of window size can improve the overall performance. GLVQ behaves similarly to LVQ 2.1. While GLVQ reduces the initial strong overshooting of LVQ 2.1, the prototypes remain divergent, and GLVQ produces higher generalization errors or long convergence times. RSLVQ attempts to combine the advantages of both LFM

and LVQ 2.1 by providing stability and optimal performance. However, an important issue of RSLVQ lies on its approximation of the data structure; it performs well when the actual input density are isotropic gaussian clusters with equal variance. If the assumptions depart from the input density, the results become suboptimal, and RSLVQ can even be outperformed by the simpler LVQ 2.1 and LFM-W. In all scenarios, RSLVQ displays robustness of its classification behavior with respect to the softness parameter, given sufficiently low learning rates.

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

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

Appendix A: Statistics of the Projections

For convenience, we combine the projections $h_S = \mathbf{w}_S \cdot \xi$ and $b_\sigma = \mathbf{B}_\sigma \cdot \xi$ defined in equation 4.3 into a D-dimensional vector, where $D = K + M$, as

$$\mathbf{x} = \begin{pmatrix} h_1^\mu & h_2^\mu & \dots & h_K^\mu & b_1^\mu & b_2^\mu & \dots & b_M^\mu \end{pmatrix}^T . \tag{A.1}$$

In our analysis of online learning, we assume that ξ is statistically independent from $\mathbf{w}_S$ because ξ^μ is uncorrelated to all previous data and $\mathbf{w}_S^{\mu-1}$. Therefore, we observe that h_S and b_σ become correlated gaussian random quantities following the central limit theorem and can be fully described by their first and second moments—its conditional averages $\mu_\sigma = \langle \mathbf{x} \rangle_\sigma$ and conditional covariance matrix $C_\sigma = \langle \mathbf{x} \cdot \mathbf{x}^T \rangle_\sigma$. We compute these averages in the following.

A.1 First-Order Statistics. We compute the averages of the components of $\mathbf{x}$ as follows:

$$\langle h_i \rangle_\sigma = \int_{\mathbb{R}^N} \xi \cdot \mathbf{w}_i \, p(\xi|\sigma) d\xi = \mathbf{w}_i \cdot \int_{\mathbb{R}^N} \xi \, p(\xi|\sigma) d\xi = \mathbf{w}_i \cdot \ell_\sigma \mathbf{B}_\sigma = \ell_\sigma R_{i\sigma} \tag{A.2}$$

$$\langle b_\tau \rangle_\sigma = \int_{\mathbb{R}^N} \xi \cdot \mathbf{B}_\tau \, p(\xi|\sigma) d\xi = \mathbf{B}_\tau \cdot \int_{\mathbb{R}^N} \xi \, p(\xi|\sigma) d\xi = \mathbf{B}_\tau \cdot \ell_\sigma \mathbf{B}_\sigma = \ell_\sigma T_{\tau\sigma}, \tag{A.3}$$

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

$$\mu = \ell_\sigma \begin{pmatrix} R_{1\sigma} & R_{2\sigma} & \dots & R_{K\sigma} & T_{1\sigma} & T_{2\sigma} & \dots & T_{M\sigma} \end{pmatrix}^T. \tag{A.4}$$

A.2 Second-Order Statistics. To compute the conditional variance $\langle \mathbf{x}_n \mathbf{x}_m \rangle_\sigma - \langle \mathbf{x}_n \rangle_\sigma \langle \mathbf{x}_m \rangle_\sigma$, we first look at the average

$$\begin{aligned}
\langle h_i h_j \rangle_\sigma &= \left\langle \left(\sum_{k=1}^N (\mathbf{w}_i)_k (\xi)_k \right) \left(\sum_{l=1}^N (\mathbf{w}_j)_l (\xi)_l \right) \right\rangle_\sigma \\
&= \left\langle \sum_{k=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_k (\xi)_k (\xi)_k + \sum_{k=1}^N \sum_{l=1, l\neq k}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_l (\xi)_k (\xi)_l \right\rangle_\sigma \\
&= \sum_{k=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_k \langle (\xi)_k (\xi)_k \rangle_\sigma + \sum_{k=1}^N \sum_{l=1, l\neq k}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_l \langle (\xi)_k (\xi)_l \rangle_\sigma . \\
&= \sum_{k=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_k \left(\upsilon_\sigma + \ell_\sigma^2 (\mathbf{B}_\sigma)_k (\mathbf{B}_\sigma)_k \right) \\
&\quad + \sum_{k=1}^N \sum_{l=1, l\neq k}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_l \ell_\sigma^2 (\mathbf{B}_\sigma)_k (\mathbf{B}_\sigma)_l \\
&= \upsilon_\sigma \sum_{k=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_k + \ell_\sigma^2 \sum_{k=1}^N \sum_{l=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_l (\mathbf{B}_\sigma)_k (\mathbf{B}_\sigma)_l \\
&= \upsilon_\sigma \mathbf{w}_i \cdot \mathbf{w}_j + \ell_\sigma^2 (\mathbf{w}_i \cdot \mathbf{B}_\sigma)(\mathbf{w}_j \cdot \mathbf{B}_\sigma) = \upsilon_\sigma Q_{ij} + \ell_\sigma^2 R_{i\sigma} R_{j\sigma}.
\end{aligned} \tag{A.5}$$

Here we exploit the following:

$$\langle(\xi)_k(\xi)_k\rangle_\sigma = \upsilon_\sigma + \langle(\xi)_k\rangle_\sigma\langle(\xi)_k\rangle_\sigma = \upsilon_\sigma + \ell_\sigma^2(\mathbf{B}_\sigma)_k(\mathbf{B}_\sigma)_k \quad \text{and}$$
$$\langle(\xi)_k(\xi)_l\rangle_\sigma = \langle(\xi)_k\rangle_\sigma\langle(\xi)_l\rangle_\sigma = \ell_\sigma^2(\mathbf{B}_\sigma)_k(\mathbf{B}_\sigma)_l.$$

Hence we obtain the conditional second-order moment, from equations A.5 and A.2:

$$\begin{aligned}\langle h_i h_j\rangle_\sigma - \langle h_i\rangle_\sigma\langle h_j\rangle_\sigma &= \upsilon_\sigma Q_{ij} + \ell_\sigma^2 R_{i\sigma}R_{j\sigma} - \ell_\sigma R_{i\sigma}\ell_\sigma R_{j\sigma}\\ &= \upsilon_\sigma Q_{ij}. \end{aligned} \tag{A.6}$$

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

$$\langle b_\tau b_\rho\rangle_\sigma - \langle b_\tau\rangle_\sigma\langle b_\rho\rangle_\sigma = \upsilon_\sigma T_{\tau\rho} + \ell_\sigma^2 T_{\tau\sigma}T_{\rho\sigma} - \ell_\sigma T_{\tau\sigma}\ell_\sigma T_{\rho\sigma} = \upsilon_\sigma T_{\tau\rho} \tag{A.7}$$
$$\langle h_i b_\tau\rangle_\sigma - \langle h_i\rangle_\sigma\langle b_\tau\rangle_\sigma = \upsilon_\sigma R_{i\tau} + \ell_\sigma^2 R_{i\sigma}T_{\tau\sigma} - \ell_\sigma R_{i\sigma}\ell_\sigma T_{\tau\sigma} = \upsilon_\sigma R_{i\tau}. \tag{A.8}$$

The conditional covariance matrix $C_\sigma = \langle \mathbf{x}\cdot\mathbf{x}^T\rangle_\sigma$ can be written in terms of order parameters as

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

Appendix B: Form of the Differential Equations

In order to perform the ordinary differential equations described in equation 4.4, we need to plug in the values of

$$\langle f_S\rangle, \quad \langle \mathbf{x}_n f_S\rangle, \quad \text{and} \quad \langle f_S f_T\rangle. \tag{B.1}$$

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

B.1 LVQ 2.1. The general modulation function for LVQ 2.1 is described in equation 3.5 as

$$f_S = \chi(c_S, y_\sigma^\mu) \sum_{T: c_T \neq c_S} \left(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}\right) \prod_{U \neq S,T} \Theta_{SU}\Theta_{TU},$$

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

$$\begin{aligned}
\Theta_{ST}^{\delta} &= \Theta\left(d_T - d_S - \delta\right) \\
&= \Theta\left(-2\mathbf{w}_T \cdot \xi^\mu + \mathbf{w}_T^2 + 2\mathbf{w}_S \cdot \xi^\mu - \mathbf{w}_S^2 - \delta\right) \\
&= \Theta\left(-2h_T^\mu + 2h_S^\mu + Q_{TT} - Q_{SS} - \delta\right) \\
&= \Theta\left(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}^\delta\right),
\end{aligned} \tag{B.2}$$

$$\text{with } \alpha_{ST} = (0, \ldots, \underbrace{+2}_{\text{at } S}, \ldots, \underbrace{-2}_{\text{at } T}, \ldots, 0) \text{ and } \beta_{ST}^\delta = Q_{SS} - Q_{TT} - \delta.$$

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

$$f_S = \chi(c_S, y_\sigma^\mu)\left(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}\right). \tag{B.3}$$

And the required averages over the joint density, equation B.1, are calculated as

$$\begin{aligned}
\langle f_S \rangle &= \left\langle \chi(c_S, y_\sigma^\mu)\left(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}\right)\right\rangle = \sum_{\sigma=1}^{M} p_\sigma \chi(c_S, y_\sigma) \left\langle \Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta} \right\rangle_\sigma \\
\langle \mathbf{x}_n f_S \rangle &= \sum_{\sigma=1}^{M} p_\sigma \chi(c_S, y_\sigma) \left\langle \mathbf{x}_n\left(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}\right)\right\rangle_\sigma \\
\langle f_S f_S \rangle &= \left\langle \chi(c_S, y_\sigma^\mu)^2 \left(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}\right)^2 \right\rangle = \sum_{\sigma=1}^{M} p_\sigma \left\langle \Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta} \right\rangle_\sigma \\
\langle f_S f_T \rangle &= \left\langle \chi(c_S, y_\sigma^\mu)\chi(c_T, y_\sigma^\mu)\left(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}\right)^2 \right\rangle = -\sum_{\sigma=1}^{M} p_\sigma \left\langle \Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta} \right\rangle_\sigma .
\end{aligned} \tag{B.4}$$

The quantities $\langle(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta})\rangle_\sigma$ and $\langle \mathbf{x}_n(\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta})\rangle_\sigma$ are calculated in Appendix C.

B.2 LFM-W. The general modulation function for LFM-W is described in equation 3.6 as

$$f_S = \begin{cases} \sum_{K:c_K \neq y_\sigma} \left(\Theta_{KS} - \Theta^\delta_{KS}\right) \psi(S,K) & \text{if } c_S = y^\mu_\sigma \\ \sum_{J:c_J = y_\sigma} \left(\Theta_{SJ} - \Theta^\delta_{SJ}\right) \psi(J,S) & \text{else} \end{cases}, \tag{B.5}$$

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

$$\begin{aligned} \langle f_S \rangle &= \sum_{\sigma:y_\sigma=c_S}^{M} p_\sigma \left\langle \Theta_{TS} - \Theta^\delta_{TS} \right\rangle_\sigma + \sum_{\sigma:y_\sigma \neq c_S}^{M} p_\sigma \left\langle \Theta_{ST} - \Theta^\delta_{ST} \right\rangle_\sigma \\ \langle \mathbf{x}_n f_S \rangle &= \sum_{\sigma:y_\sigma=c_S}^{M} p_\sigma \left\langle \mathbf{x}_n \left(\Theta_{TS} - \Theta^\delta_{TS}\right) \right\rangle_\sigma + \sum_{\sigma:y_\sigma \neq c_S}^{M} p_\sigma \left\langle \mathbf{x}_n \left(\Theta_{ST} - \Theta^\delta_{ST}\right) \right\rangle_\sigma . \end{aligned} \tag{B.6}$$

B.3 GLVQ. The general modulation function for GLVQ is described in equation 3.12 as

$$f_S = \begin{cases} \sum_{K:c_K \neq y_\sigma} \left(\frac{2}{v_G} \phi\left(\frac{d_S - d_K}{v_G}\right)\right) \psi(S,K) & \text{if } c_S = y^\mu_\sigma \\ -\sum_{J:c_J = y_\sigma} \left(\frac{2}{v_G} \phi\left(\frac{d_J - d_S}{v_G}\right)\right) \psi(J,S) & \text{else} \end{cases}. \tag{B.7}$$

For two prototypes,

$$\begin{aligned} \langle f_S \rangle &= \sum_{\sigma:y_\sigma=c_S}^{M} p_\sigma \frac{2}{v_G} \langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma \\ &\quad - \sum_{\sigma:y_\sigma \neq c_S}^{M} p_\sigma \frac{2}{v_G} \langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma \\ \langle \mathbf{x}_n f_S \rangle &= \sum_{\sigma:y_\sigma=c_S}^{M} p_\sigma \frac{2}{v_G} \langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma \\ &\quad - \sum_{\sigma:y_\sigma \neq c_S}^{M} p_\sigma \frac{2}{v_G} \langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma , \end{aligned} \tag{B.8}$$

$$\text{with } \alpha_{ST} = \left\{ \ldots, \underbrace{-\frac{2}{v_G}}_{\text{at } S}, \ldots, \underbrace{+\frac{2}{v_G}}_{\text{at } T}, \ldots 0, 0 \right\}, \quad \beta_{ST} = -\frac{Q_{SS} - Q_{TT}}{v_G}.$$

The quantities $\left\langle \phi\left(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}\right)\right\rangle_{\sigma}$ are found in equation C.7 in appendix C.

B.4 RSLVQ. With one prototype representing each class, equation 3.14 becomes

$$\begin{aligned} P_{\sigma}(S|\xi^{\mu}) &= \frac{\exp\left(-(\xi^{\mu} - \mathbf{w}_S^{\mu})^2/2v_{\text{soft}}\right)}{\exp\left(-(\xi^{\mu} - \mathbf{w}_S^{\mu})^2/2v_{\text{soft}}\right)} = 1 \\ P(S|\xi^{\mu}) &= \frac{\exp\left(-(\xi^{\mu} - \mathbf{w}_S^{\mu})^2/2v_{\text{soft}}\right)}{\sum_{T=1}^{K} \exp\left(-(\xi^{\mu} - \mathbf{w}_T^{\mu})^2/2v_{\text{soft}}\right)} \\ &= \frac{1}{1 + \sum_{T \neq S}^{K} \exp\left(\frac{1}{2v_{\text{soft}}}\left(-2\xi^{\mu}\mathbf{w}_S^{\mu} + (\mathbf{w}_S^{\mu})^2 + 2\xi^{\mu}\mathbf{w}_T^{\mu} - (\mathbf{w}_T^{\mu})^2\right)\right)} \\ &= \frac{1}{1 + \sum_{T \neq S}^{K} \exp\left(\frac{1}{2v_{\text{soft}}}\left(-2h_S + Q_{SS} + 2h_T - Q_{TT}\right)\right)} \\ &= \frac{1}{1 + \sum_{T \neq S}^{K} \exp\left(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}\right)}, \end{aligned} \tag{B.9}$$

where we defined

$$\alpha_{ST} = \left\{ \ldots, \underbrace{-\frac{1}{v_{\text{soft}}}}_{\text{at } S}, \ldots, \underbrace{+\frac{1}{v_{\text{soft}}}}_{\text{at } T}, \ldots 0, 0 \right\}, \; \beta_{ST} = -\frac{Q_{SS} - Q_{TT}}{2v_{\text{soft}}}.$$

Therefore, the RSLVQ modulation function becomes

$$f_S = \frac{1}{v_{\text{soft}}}\left(\delta(c_S, y_{\sigma}^{\mu}) - \Omega_S\right), \tag{B.10}$$

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

$$\Omega_S = \frac{1}{1 + \sum_{T \neq S}^{K} \exp\left(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}\right)}. \tag{B.11}$$

We obtain the averages

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

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

$$= \begin{cases} \frac{1}{v_{\text{soft}}} \left(\sum_{\sigma: y_\sigma = c_S} p_\sigma \ell_\sigma R_{n\sigma} - \sum_\sigma p_\sigma \langle \mathbf{x}_n \Omega_S \rangle_\sigma \right) & \text{if } n \le K \\ \frac{1}{v_{\text{soft}}} \left(\sum_{\sigma: y_\sigma = c_S} p_\sigma \ell_\sigma T_{(n-K)\sigma} - \sum_\sigma p_\sigma \langle \mathbf{x}_n \Omega_S \rangle_\sigma \right) & \text{if } n > K. \end{cases} \tag{B.12}$$

The required quantities $\langle \Omega_S \rangle_\sigma$ and $\langle \mathbf{x}_n \Omega_S \rangle_\sigma$ are supplied in appendix C.

Appendix C: Gaussian Averages

C.1 Two Prototypes. For generic functions $f_{ab} \equiv f(\alpha_{ab} \cdot \mathbf{x} - \beta_{ab})$, the quantities $\langle f_{ab} \rangle_\sigma$ and $\langle \mathbf{x}_n f_{ab} \rangle_\sigma$ are required:

$$\begin{aligned} \langle f_{ab} \rangle_\sigma &= \frac{1}{(2\pi)^{D/2} (\det(C_\sigma))^{1/2}} \int_{\mathbb{R}^D} f(\alpha_{ab} \cdot \mathbf{x} - \beta_{ab}) \\ &\quad \times \exp\left(-\frac{1}{2} (\mathbf{x} - \mu)^T C_\sigma^{-1} (\mathbf{x} - \mu) \right) d\mathbf{x} \\ &= \frac{1}{(2\pi)^{D/2} (\det(C_\sigma))^{1/2}} \int_{\mathbb{R}^D} f\left(\alpha_{ab} \cdot \mathbf{x}' + \alpha_{ab} \cdot \mu - \beta_{ab} \right) \\ &\quad \times \exp\left(-\frac{1}{2} (\mathbf{x}')^T C_\sigma^{-1} (\mathbf{x}') \right) d\mathbf{x}' \\ &= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} f\left(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \widetilde{\beta}_{ab,\sigma} \right) \exp\left(-\frac{1}{2} \mathbf{y}^2 \right) d\mathbf{y}, \end{aligned} \tag{C.1}$$

with $\widetilde{\beta}_{ab,\sigma} = \alpha_{ab} \cdot \mathbf{x} - \beta_{ab}$. Rotating the coordinate system, we obtain

$$\begin{aligned} \langle f_{ab} \rangle_\sigma &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f\left(\|\alpha_{ab} C_\sigma^{-1/2}\| \widetilde{y} + \widetilde{\beta}_{ab,\sigma} \right) \exp\left(-\frac{1}{2} \widetilde{y}^2 \right) d\widetilde{y} \\ &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f\left(\widetilde{\alpha}_{ab,\sigma} \widetilde{y} + \widetilde{\beta}_{ab,\sigma} \right) \exp\left(-\frac{1}{2} \widetilde{y}^2 \right) d\widetilde{y}, \end{aligned} \tag{C.2}$$

with $\widetilde{\alpha}_{ab,\sigma} = \|\alpha_{ab} C_\sigma^{-1/2}\|$. Next we calculate the quantity

$$\begin{aligned}
\langle \mathbf{x}_n f_{ab} \rangle_\sigma &= \frac{1}{(2\pi)^{D/2} (\det(C_\sigma))^{1/2}} \int_{\mathbb{R}^D} \mathbf{x}_n f\left(\alpha_{ab} \cdot \mathbf{x} - \beta_{ab}\right) \\
&\quad \times \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T C_\sigma^{-1} (\mathbf{x}-\mu)\right) d\mathbf{x} \\
&= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \left(C_\sigma^{1/2}\mathbf{y} + \mu\right)_n f\left(\alpha_{ab} C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma}\right) \\
&\quad \times \exp\left(-\frac{1}{2}\mathbf{y}^2\right) d\mathbf{y} \\
&= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \left(C_\sigma^{1/2}\mathbf{y}\right)_n f\left(\alpha_{ab} C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma}\right) \exp\left(-\frac{1}{2}\mathbf{y}^2\right) d\mathbf{y} \\
&\quad + (\mu)_n \langle f_{ab} \rangle_\sigma \,. \qquad \text{(C.3)}
\end{aligned}$$

C.1.1 LVQ 2.1, LFM-W. The following quantities are required for two prototypes: LVQ2.1 and LFM-W:

$$\left\langle \Theta_{ab}^\delta - \Theta_{ab}^\gamma \right\rangle_\sigma = \Phi\left(\frac{\widetilde{\beta}_{ab,\sigma}^\delta}{\widetilde{\alpha}_{ab,\sigma}}\right) - \Phi\left(\frac{\widetilde{\beta}_{ab,\sigma}^\gamma}{\widetilde{\alpha}_{ab,\sigma}}\right) \qquad \text{(C.4)}$$

$$\begin{aligned}
&\left\langle \mathbf{x}_n \left(\Theta_{ab}^\delta - \Theta_{ab}^\gamma\right) \right\rangle_\sigma \\
&\quad = \frac{(C_\sigma \alpha_{ab})_n}{\sqrt{2\pi}\widetilde{\alpha}_{ab,\sigma}} \left\{ \exp\left[-\frac{1}{2}\left(\frac{\widetilde{\beta}_{ab,\sigma}^\delta}{\widetilde{\alpha}_{ab,\sigma}}\right)^2\right] - \exp\left[-\frac{1}{2}\left(\frac{\widetilde{\beta}_{ab,\sigma}^\gamma}{\widetilde{\alpha}_{ab,\sigma}}\right)^2\right] \right\} \\
&\qquad + (\mu_\sigma)_n \left[\Phi\left(\frac{\widetilde{\beta}_{ab,\sigma}^\delta}{\widetilde{\alpha}_{ab,\sigma}}\right) - \Phi\left(\frac{\widetilde{\beta}_{ab,\sigma}^\gamma}{\widetilde{\alpha}_{ab,\sigma}}\right) \right] \qquad \text{(C.5)}
\end{aligned}$$

C.1.2 GLVQ. For GLVQ, the quantities $\langle \phi_{ab} \rangle_\sigma$ and $\langle \mathbf{x}_n \phi_{ab} \rangle_\sigma$ are required:

$$\begin{aligned}
\langle \phi_{ab} \rangle_\sigma &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \phi\left(\widetilde{\alpha}_{ab,\sigma}\widetilde{y} + \widetilde{\beta}_{ab,\sigma}\right) \exp\left(-\frac{1}{2}\widetilde{y}^2\right) d\widetilde{y} \\
&= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\widetilde{\beta}_{ab,\sigma}^2\right) \\
&\quad \times \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(\widetilde{\alpha}_{ab,\sigma}^2 + 1)\widetilde{y}^2 - \widetilde{\alpha}_{ab,\sigma}\widetilde{\beta}_{ab,\sigma}\widetilde{y}\right) d\widetilde{y}. \qquad \text{(C.6)}
\end{aligned}$$

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

$$\begin{aligned}\langle \phi_{ab} \rangle_\sigma &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\widetilde{\beta}^2_{ab,\sigma}\right) \frac{1}{\sqrt{\widetilde{\alpha}^2_{ab,\sigma}+1}} \exp\left(\frac{\widetilde{\alpha}^2_{ab,\sigma}\widetilde{\beta}^2_{ab,\sigma}}{2(\widetilde{\alpha}^2_{ab,\sigma}+1)}\right) \\ &= \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\widetilde{\alpha}^2_{ab,\sigma}+1}} \exp\left(-\frac{1}{2}\widetilde{\beta}^2_{ab,\sigma}\left(1 - \frac{\widetilde{\alpha}^2_{ab,\sigma}}{(\widetilde{\alpha}^2_{ab,\sigma}+1)}\right)\right) \\ &= \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\widetilde{\alpha}^2_{ab,\sigma}+1}} \exp\left(-\frac{1}{2}\frac{\widetilde{\beta}^2_{ab,\sigma}}{\widetilde{\alpha}^2_{ab,\sigma}+1}\right). \end{aligned} \tag{C.7}$$

C.1.3 RSLVQ. For RSLVQ, the quantities $\langle \Omega_{ab} \rangle_\sigma$ and $\langle \mathbf{x}_n \Omega_{ab} \rangle_\sigma$ are required:

$$\langle \Omega_{ab} \rangle_\sigma = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \frac{1}{1+\exp(\widetilde{\alpha}_{ab,\sigma}\widetilde{y} + \widetilde{\beta}_{ab,\sigma})} \exp\left(-\frac{1}{2}\widetilde{y}^2\right) d\widetilde{y}. \tag{C.8}$$

This one-dimensional integration has to be solved numerically:

$$\begin{aligned}\langle \mathbf{x}_n \Omega_{ab} \rangle_\sigma &= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \frac{\left(C_\sigma^{1/2}\mathbf{y}\right)_n}{1+\exp\left(\alpha_{ab} C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma}\right)} \exp\left(-\frac{1}{2}\mathbf{y}^2\right) d\mathbf{y} \\ &\quad + (\mu)_n \langle \Omega_{ab} \rangle_\sigma \\ &= \frac{1}{(2\pi)^{D/2}} \sum_{j=1}^{D} I_j + (\mu)_n \langle \Omega_{ab} \rangle_\sigma , \end{aligned} \tag{C.9}$$

where

$$I_j = \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}} \frac{(C_\sigma^{1/2})_{nj}(\mathbf{y})_n}{1+\exp(\alpha_{ab} C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma})} \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) d(\mathbf{y})_j. \tag{C.10}$$

Applying integration by parts $\int u dv = uv - \int v du$ with

$$\begin{aligned} u &= \frac{1}{1+\exp(\alpha_{ab} C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma})} \\ v &= (C_\sigma^{1/2})_{nj} \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) \end{aligned}$$

$$du = -\frac{\exp(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma})}{\left(1+\exp(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma})\right)^2}\frac{\partial}{\partial(\mathbf{y})_j}\left(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y}\right)d(\mathbf{y})_j$$

$$dv = -(C_\sigma^{1/2})_{nj}(\mathbf{y})_j \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right)d(\mathbf{y})_j, \tag{C.11}$$

we obtain

$$\begin{aligned} I_j = &\left[\frac{1}{1+\exp(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma})}(C_\sigma^{1/2})_{nj}\exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right)\right]_{-\infty}^{\infty} \\ &-\int_{\mathbb{R}}\frac{(C_\sigma^{1/2})_{nj}\exp(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma})}{\left(1+\exp(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma})\right)^2}\frac{\partial}{\partial(\mathbf{y})_j}\left(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y}\right) \\ &\times\exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right)d(\mathbf{y})_j. \end{aligned} \tag{C.12}$$

$$\begin{aligned} \langle \mathbf{x}_n\Omega_{ab}\rangle_\sigma = &-\frac{1}{(2\pi)^{D/2}}\sum_{j=1}^{D}\int_{\mathbb{R}}\frac{(C_\sigma^{1/2})_{nj}\exp\left(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma}\right)}{\left(1+\exp\left(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma}\right)\right)^2} \\ &\frac{\partial}{\partial(\mathbf{y})_j}\left(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y}\right)\exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right)d(\mathbf{y})_j + (\mu)_n\langle\Omega_{ab}\rangle_\sigma \\ = &-\frac{1}{(2\pi)^{D/2}}(C_k\alpha_{ab})_n\int_{\mathbb{R}}\frac{\exp\left(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma}\right)}{\left(1+\exp\left((\alpha_{ab}C_\sigma^{-1/2}\mathbf{y} + \widetilde{\beta}_{ab,\sigma}\right)\right)^2} \\ &\times\exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right)d(\mathbf{y})_j. \end{aligned} \tag{C.13}$$

After applying rotation,

$$\begin{aligned} \langle \mathbf{x}_n\Omega_{ab}\rangle_\sigma = &-\frac{(C_k\alpha_{ab})_n}{\sqrt{2\pi}}\int_{\mathbb{R}}\frac{\exp\|\alpha_{ab}C_\sigma^{-1/2}\|\widetilde{y} + \widetilde{\beta}_{ab,\sigma})}{\left(1+\exp\left(\|\alpha_{ab}C_\sigma^{-1/2}\|\widetilde{y} + \widetilde{\beta}_{ab,\sigma}\right)\right)^2} \\ &\times\exp\left(-\frac{1}{2}\widetilde{y}^2\right)d\widetilde{y} + (\mu_k)_n\langle\Omega_{ab}\rangle_\sigma \\ = &-\frac{(C_k\alpha_{ab})_n}{\sqrt{2\pi}}\int_{\mathbb{R}}\frac{\exp\left(\widetilde{\alpha}_{ab,\sigma}\widetilde{y} + \widetilde{\beta}_{ab,\sigma}\right)}{\left(1+\exp\left(\widetilde{\alpha}_{ab,\sigma}\widetilde{y} + \widetilde{\beta}_{ab,\sigma}\right)\right)^2}\exp\left(-\frac{1}{2}\widetilde{y}^2\right)d\widetilde{y} \\ &+(\mu_k)_n\langle\Omega_{ab}\rangle_\sigma, \end{aligned} \tag{C.14}$$

which is also solved numerically.

C.2 Three Prototypes. For generic function $f_{ab}f_{cd} \equiv f(\alpha_{ab}\cdot\mathbf{x}-\beta_{ab})\, f(\alpha_{cd}\cdot\mathbf{x}-\beta_{cd})$, the quantities $\langle f_{ab}f_{cd}\rangle_k$ and $\langle \mathbf{x}_n f_{ab}f_{cd}\rangle_k$ are required:

$$\langle f_{ab}f_{cd}\rangle_\sigma = \frac{1}{(2\pi)^{D/2}}\int_{\mathbb{R}^D} f\left(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y}+\widetilde{\beta}_{ab,\sigma}\right) f\left(\alpha_{cd}C_\sigma^{-1/2}\mathbf{y}+\widetilde{\beta}_{cd}\right) \times \exp\left(-\frac{1}{2}\mathbf{y}^2\right) d\mathbf{y}. \tag{C.15}$$

Next we calculate the quantity

$$\langle \mathbf{x}_n f_{ab}f_{cd}\rangle_\sigma = \frac{1}{(2\pi)^{D/2}}\int_{\mathbb{R}^D} \left(C_\sigma^{1/2}\mathbf{y}\right)_n f\left(\alpha_{ab}C_\sigma^{-1/2}\mathbf{y}+\widetilde{\beta}_{ab,\sigma}\right) \times f\left(\alpha_{cd}C_\sigma^{-1/2}\mathbf{y}+\widetilde{\beta}_{cd}\right) \times \exp\left(-\frac{1}{2}\mathbf{y}^2\right) d\mathbf{y} + (\mu)_n \langle f_{ab}f_{cd}\rangle_\sigma . \tag{C.16}$$

The quantities $\langle\Theta_{ab}\Theta_{cd}\rangle_\sigma$ and $\langle \mathbf{x}_n\Theta_{ab}\Theta_{cd}\rangle_\sigma$ have been calculated in Witoelar et al. (2008) as follows:

$$\langle\Theta_{ab}\Theta_{cd}\rangle_\sigma = \frac{1}{\sqrt{2\pi}}\int_{-\frac{\widetilde{\beta}_{ab,\sigma}}{\widetilde{\alpha}_{ab,\sigma}}}^{\infty} \exp\left(-\frac{1}{2}y_1'^2\right)\Phi\left(\frac{\widetilde{\beta}_{cd,\sigma}\widetilde{\alpha}_{ab,\sigma}+(\alpha_{cd}C_\sigma\alpha_{ab})y_1'}{\sqrt{\widetilde{\alpha}_{cd,\sigma}^2\widetilde{\alpha}_{ab,\sigma}^2-(\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) dy_1' \tag{C.17}$$

$$\langle(\mathbf{x})_n\Theta_{ab}\Theta_{cd}\rangle_\sigma = \frac{(C_\sigma\alpha_{ab})_n}{\sqrt{(2\pi)}\widetilde{\alpha}_{ab,\sigma}}\exp\left(-\frac{1}{2}\frac{\widetilde{\beta}_{ab,\sigma}^2}{\widetilde{\alpha}_{ab,\sigma}^2}\right)\Phi\left(\frac{\widetilde{\beta}_{cd,\sigma}\widetilde{\alpha}_{ab,\sigma}^2-\widetilde{\beta}_{ab,\sigma}(\alpha_{cd}C_\sigma\alpha_{ab})}{\widetilde{\alpha}_{ab,\sigma}\sqrt{\widetilde{\alpha}_{cd,\sigma}^2\widetilde{\alpha}_{ab,\sigma}^2-(\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) + \frac{(C_\sigma\alpha_{cd})_n}{\sqrt{(2\pi)}\widetilde{\alpha}_{cd,\sigma}}\exp\left(-\frac{1}{2}\frac{\widetilde{\beta}_{cd,\sigma}^2}{\widetilde{\alpha}_{cd,\sigma}^2}\right) \times \Phi\left(\frac{\widetilde{\beta}_{ab,\sigma}\widetilde{\alpha}_{cd,\sigma}^2-\widetilde{\beta}_{cd,\sigma}(\alpha_{ab}C_\sigma\alpha_{cd})}{\widetilde{\alpha}_{cd,\sigma}\sqrt{\widetilde{\alpha}_{cd,\sigma}^2\widetilde{\alpha}_{ab,\sigma}^2-(\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) + (\mu_\sigma)_n\langle\Theta_{ab}\Theta_{cd}\rangle_\sigma . \tag{C.18}$$

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

$$\begin{aligned}
&\langle(\Theta_{ab}^{\delta}-\Theta_{ab}^{\gamma})\Theta_{cd}\rangle_{\sigma}\\
&\quad=\frac{1}{\sqrt{2\pi}}\int_{-\frac{\widetilde{\beta}_{ab,\sigma}^{\delta}}{\widetilde{\alpha}_{ab,\sigma}}}^{-\frac{\widetilde{\beta}_{ab,\sigma}^{\gamma}}{\widetilde{\alpha}_{ab,\sigma}}}\exp\left(-\frac{1}{2}y_1'^2\right)\Phi\left(\frac{\widetilde{\beta}_{cd,\sigma}\widetilde{\alpha}_{ab,\sigma}+(\alpha_{cd}C_{\sigma}\alpha_{ab})y_1'}{\sqrt{\widetilde{\alpha}_{cd,\sigma}^2\widetilde{\alpha}_{ab,\sigma}^2-(\alpha_{cd}C_{\sigma}\alpha_{ab})^2}}\right)dy_1'\\
&\langle(\mathbf{x})_n(\Theta_{ab}^{\delta}-\Theta_{ab}^{\gamma})\Theta_{cd}\rangle_{\sigma}\\
&\quad=\frac{(C_{\sigma}\alpha_{ab})_n}{\sqrt{(2\pi)}\widetilde{\alpha}_{ab,\sigma}}\exp\left(-\frac{1}{2}\frac{(\widetilde{\beta}_{ab,\sigma}^{\delta})^2}{\widetilde{\alpha}_{ab,\sigma}^2}\right)\\
&\qquad\times\Phi\left(\frac{\widetilde{\beta}_{cd,\sigma}\widetilde{\alpha}_{ab,\sigma}^2-\widetilde{\beta}_{ab,\sigma}^{\delta}(\alpha_{cd}C_{\sigma}\alpha_{ab})}{\widetilde{\alpha}_{ab,\sigma}\sqrt{\widetilde{\alpha}_{cd,\sigma}^2\widetilde{\alpha}_{ab,\sigma}^2-(\alpha_{cd}C_{\sigma}\alpha_{ab})^2}}\right)\\
&\qquad+\frac{(C_{\sigma}\alpha_{cd})_n}{\sqrt{(2\pi)}\widetilde{\alpha}_{cd,\sigma}}\exp\left(-\frac{1}{2}\frac{\widetilde{\beta}_{cd,\sigma}^2}{\widetilde{\alpha}_{cd,\sigma}^2}\right)\\
&\qquad\times\Phi\left(\frac{\widetilde{\beta}_{ab,\sigma}^{\delta}\widetilde{\alpha}_{cd,\sigma}^2-\widetilde{\beta}_{cd,\sigma}(\alpha_{ab}C_{\sigma}\alpha_{cd})}{\widetilde{\alpha}_{cd,\sigma}\sqrt{\widetilde{\alpha}_{cd,\sigma}^2\widetilde{\alpha}_{ab,\sigma}^2-(\alpha_{cd}C_{\sigma}\alpha_{ab})^2}}\right)\\
&\qquad-\frac{(C_{\sigma}\alpha_{ab})_n}{\sqrt{(2\pi)}\widetilde{\alpha}_{ab,\sigma}}\exp\left(-\frac{1}{2}\frac{(\widetilde{\beta}_{ab,\sigma}^{\gamma})^2}{\widetilde{\alpha}_{ab,\sigma}^2}\right)\\
&\qquad\times\Phi\left(\frac{\widetilde{\beta}_{cd,\sigma}\widetilde{\alpha}_{ab,\sigma}^2-\widetilde{\beta}_{ab,\sigma}^{\gamma}(\alpha_{cd}C_{\sigma}\alpha_{ab})}{\widetilde{\alpha}_{ab,\sigma}\sqrt{\widetilde{\alpha}_{cd,\sigma}^2\widetilde{\alpha}_{ab,\sigma}^2-(\alpha_{cd}C_{\sigma}\alpha_{ab})^2}}\right)\\
&\qquad-\frac{(C_{\sigma}\alpha_{cd})_n}{\sqrt{(2\pi)}\widetilde{\alpha}_{cd,\sigma}}\exp\left(-\frac{1}{2}\frac{\widetilde{\beta}_{cd,\sigma}^2}{\widetilde{\alpha}_{cd,\sigma}^2}\right)\\
&\qquad\times\Phi\left(\frac{\widetilde{\beta}_{ab,\sigma}^{\gamma}\widetilde{\alpha}_{cd,\sigma}^2-\widetilde{\beta}_{cd,\sigma}(\alpha_{ab}C_{\sigma}\alpha_{cd})}{\widetilde{\alpha}_{cd,\sigma}\sqrt{\widetilde{\alpha}_{cd,\sigma}^2\widetilde{\alpha}_{ab,\sigma}^2-(\alpha_{cd}C_{\sigma}\alpha_{ab})^2}}\right)\\
&\quad+(\mu_{\sigma})_n\langle(\Theta_{ab}^{\delta}-\Theta_{ab}^{\gamma})\Theta_{cd}\rangle_{\sigma}. \qquad \text{(C.19)}
\end{aligned}$$

For LVQ 2.1, the following average is required:

$$\langle(\Theta_{ab}^{\delta}-\Theta_{ab}^{\gamma})\Theta_{ac}\Theta_{bc}\rangle_{\sigma}=\frac{1}{\sqrt{2\pi}}\int_{y_{1,\min}}^{y_{1,\max}}\exp\left(-\frac{1}{2}y_1'^2\right)\Phi(-y_2^*)dy_1 \qquad \text{(C.20)}$$

$$\text{with } y_{1,\min} = -\frac{\widetilde{\beta}^{\delta}_{ab,\sigma}}{\widetilde{\alpha}_{ab,\sigma}}, \quad y_{1,\max} = -\frac{\widetilde{\beta}^{\gamma}_{ab,\sigma}}{\widetilde{\alpha}_{ab,\sigma}}$$
$$y_2{}^* = \min\left(\frac{-\widetilde{\beta}_{ac} - (\alpha_{ac}C_\sigma^{1/2}e_1)y_1}{\alpha_{ac}C_\sigma^{1/2}e_2}, \frac{-\widetilde{\beta}_{bc} - (\alpha_{bc}C_\sigma^{1/2}e_1)y_1}{\alpha_{bc}C_\sigma^{1/2}e_2}\right) \quad \text{(C.21)}$$

$$\langle \mathbf{x}_n(\Theta^{\delta}_{ab} - \Theta^{\gamma}_{ab})\Theta_{ac}\Theta_{bc}\rangle_\sigma$$
$$= I_{ab} + I_{ac} + I_{bc} + (\mu)_n\langle \mathbf{x}_n(\Theta^{\delta}_{ab} - \Theta^{\gamma}_{ab})\Theta_{ac}\Theta_{bc}\rangle_\sigma, \quad \text{(C.22)}$$

where

$$I_{ab} = \frac{(C_\sigma\alpha_{ab})_n}{\sqrt{2\pi}\widetilde{\alpha}_{ab,\sigma}}\Bigg[\exp\left(-\frac{1}{2}\left(-\frac{\widetilde{\beta}^{\delta}_{ab,\sigma}}{\widetilde{\alpha}_{ab,\sigma}}\right)^2\right)\left(\Phi(-y^{\delta}_{2,\min}) - \Phi(-y^{\delta}_{2,\max})\right)$$
$$-\exp\left(-\frac{1}{2}\left(-\frac{\widetilde{\beta}^{\gamma}_{ab,\sigma}}{\widetilde{\alpha}_{ab,\sigma}}\right)^2\right)\left(\Phi(-y^{\gamma}_{2,\min}) - \Phi(-y^{\gamma}_{2,\max})\right)\Bigg] \quad \text{(C.23)}$$

$$I_{ac} = \frac{(C_\sigma\alpha_{ac})_n}{\sqrt{2\pi}\widetilde{\alpha}_{ac,\sigma}}\Bigg[\exp\left(-\frac{1}{2}(z)^2\right)\left(\Phi(-y_{2,\min}) - \Phi(-y_{2,\max})\right)$$
$$-\exp\left(-\frac{1}{2}(z)^2\right)\left(\Phi(-y_{2,\min}) - \Phi(-y_{2,\max})\right)\Bigg] \quad \text{(C.24)}$$

$$\text{with} \quad y_{1,\min} = -\frac{\widetilde{\beta}^{\delta}_{ab,\sigma}}{\widetilde{\alpha}_{ab,\sigma}}, \quad y_{1,\max} = -\frac{\widetilde{\beta}^{\gamma}_{ab,\sigma}}{\widetilde{\alpha}_{ab,\sigma}}$$
$$y_2{}^* = \min\left(\frac{-\widetilde{\beta}_{ac} - (\alpha_{ac}C_\sigma^{1/2}e_1)y_1}{\alpha_{ac}C_\sigma^{1/2}e_2}, \frac{-\widetilde{\beta}_{bc} - (\alpha_{bc}C_\sigma^{1/2}e_1)y_1}{\alpha_{bc}C_\sigma^{1/2}e_2}\right). \quad \text{(C.25)}$$

Appendix D: Generalization Error

D.1 Two Prototypes. We compute the generalization error from equation 4.6 as follows. For two prototypes $\mathbf{w}_+$ and $\mathbf{w}_-$, we calculate $\epsilon_g = \sum p_\sigma \epsilon_{g,\sigma}$ with

$$\epsilon_{g,\sigma} = \langle\Theta_{-\sigma\sigma}\rangle_+ = \Phi\left(\frac{\widetilde{\beta}_{-\sigma\sigma,\sigma}}{\widetilde{\alpha}_{-\sigma\sigma,\sigma}}\right), \quad \text{(D.1)}$$

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

$$\epsilon_{g,\sigma} = \Phi\left(\frac{Q_{\sigma\sigma} - Q_{-\sigma,-\sigma} - 2\ell_\sigma(R_{\sigma,\sigma} - R_{-\sigma,\sigma})}{2\sqrt{v_\sigma}\sqrt{Q_{\sigma\sigma} - 2Q_{\sigma,-\sigma} + Q_{-\sigma,-\sigma}}}\right). \tag{D.2}$$

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

$$\frac{d\epsilon_{g\sigma}}{d\mathbf{O}} = \frac{1}{\sqrt{2\pi}2\sqrt{v_\sigma}} \exp\left(-\frac{1}{2}\left[\frac{Z_\sigma}{2\sqrt{v_\sigma}\Delta_q}\right]^2\right) \frac{d}{d\mathbf{O}} \frac{Z_\sigma}{\Delta_q}, \tag{D.3}$$

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

$$\frac{d}{d\mathbf{O}} \frac{Z_+}{\Delta_q} = \begin{bmatrix} -2\ell/\Delta_q \\ 0 \\ +2\ell/\Delta_q \\ 0 \\ 1/\Delta_q - Z_+/(2\Delta_q^3) \\ Z_+/\Delta_q^3 \\ -1/\Delta_q - Z_+/(2\Delta_q^3) \end{bmatrix}, \quad \frac{d}{d\mathbf{O}} \frac{Z_-}{\Delta_q} = \begin{bmatrix} 0 \\ +2\ell/\Delta_q \\ 0 \\ -2\ell/\Delta_q \\ -1/\Delta_q - Z_-/(2\Delta_q^3) \\ Z_-/\Delta_q^3 \\ 1/\Delta_q - Z_-/(2\Delta_q^3) \end{bmatrix} \tag{D.4}$$

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

$$\frac{d\epsilon_{g\sigma}}{d\mathbf{O}} = \sum_\sigma \frac{d\epsilon_{g\sigma}}{d\mathbf{O}} = \frac{1}{2\sqrt{2\pi}\sqrt{v}} \exp\left(-\frac{1}{2}\left[\frac{Z}{2\sqrt{v}\Delta_q}\right]^2\right) \begin{bmatrix} -\ell/\Delta_q \\ +\ell/\Delta_q \\ +\ell/\Delta_q \\ -\ell/\Delta_q \\ -Z/(2\Delta_q^3) \\ Z/\Delta_q^3 \\ -Z/(2\Delta_q^3) \end{bmatrix}. \tag{D.5}$$

D.2 Three Prototypes. To compute the generalization error in systems with three prototypes $\mathbf{w}_S, \mathbf{w}_T, \mathbf{w}_U$, we require the quantity

$$\epsilon_{g,\sigma} = \sum_{S:c_S \neq y_\sigma}^{K} \langle \Theta_{ST} \Theta_{SU} \rangle_\sigma \ , \tag{D.6}$$

where the averages are written in equation C.17.

References

Barkai, N., Seung, H. S., & Sompolinsky, H. (1993). Scaling laws in learning of classification tasks. *Phys. Rev. Lett., 70*, 3167–3170.

Bengio, Y. (2000). Gradient-based optimization of hyperparameters. *Neural Comput., 12*(8), 1889–1900.

Biehl, M. (1994). An exactly solvable model of unsupervised learning. *Europhysics Letters, 25*, 391–396.

Biehl, M., & Caticha, N. (2003). The statistical mechanics of on-line learning and generalization. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 1095–1098). Cambridge, MA: MIT Press.

Biehl, M., Freking, A., Ghosh, A., & Reents, G. (2004). A theoretical framework for analysing the dynamics of LVQ: A statistical physics approach (Tech. Rep. 2004-9-02). Groningen, Netherlands: Mathematics and Computing Science, University of Groningen. Available online at http://www.cs.rug.nl/∼biehl.

Biehl, M., Ghosh, A., & Hammer, B. (2007). Dynamics and generalization ability of LVQ algorithms. *J. Mach. Learning Res., 8*, 323–360.

Biehl, M., & Mietzner, A. (1993). Statistical mechanics of unsupervised learning. *Europhysics Letters, 27*, 421–426.

Biehl, M., & Schwarze, H. (1993). Learning drifting concepts with neural networks. *Journal of Physics A: Mathematical and General, 26*, 2651–2665.

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

Ghosh, A., Biehl, M., & Hammer, B. (2006). Performance analysis of LVQ algorithms: A statistical physics approach. *Neural Networks, 19*, 817–829.

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

Kohonen, T. (1990). Improved versions of learning vector quantization. *1990 IJCNN International Joint Conference on Neural Networks* (Vol. 1, pp. 545–550). Mahwah, NJ: Erlbaum.

Kohonen, T. (2001). *Self organising maps* (3rd ed.). Berlin: Springer.

Meir, R. (1995). Empirical risk minimization versus maximum-likelihood estimation: A case study. *Neural Computation, 7*, 144–157.

Neural Networks Research Centre, Helsinki. (2002). *Bibliography on the self-organizing maps (SOM) and learning vector quantization (LVQ)*. Otaniemi: Helsinki University of Technology. Available online at http://liinwww.ira.uka.de/bibliography/Neural/SOM.LVQ.html.

Reents, G., & Urbanczik, R. (1998). Self averaging and on-line learning. *Phys. Rev. Letter, 80*, 5445–5448.

Saad, D. (Ed.). (1999). *Online learning in neural networks*. Cambridge: Cambridge University Press.

Saad, D., & Rattray, M. (1997). Globally optimal parameters for on-line learning in multilayer neural networks. *Phys. Rev. Lett., 79*, 2578–2581.

Saad, D., & Solla, S. A. (1995). On-line learning in soft committee machines. *Phys. Rev. E, 52*, 4225–4243.

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

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

Seo, S., & Obermayer, K. (2006). Dynamic hyper parameter scaling method for LVQ algorithms. In *International Joint Conference on Neural Networks*. Piscataway, NJ: IEEE Press.

Witoelar, A., Biehl, M., Ghosh, A., & Hammer, B. (2008). Learning dynamics and robustness of vector quantization and neural gas. *Neurocomputing, 71*, 1210–1219.

Received January 21, 2010; accepted May 10, 2010.