

University of Groningen

## Modeling Affective State using Learning Vector Quantization

de Vries, Jan

**IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.**

*Document Version*

Publisher's PDF, also known as Version of record

*Publication date:*

2014

[Link to publication in University of Groningen/UMCG research database](#)

*Citation for published version (APA):*

de Vries, J. (2014). *Modeling Affective State using Learning Vector Quantization*. [Thesis fully internal (DIV), University of Groningen]. [S.n.].

### Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

### Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

## Hoofdstuk 3

---

# LEARNING DYNAMICS OF LEARNING VECTOR QUANTIZATION

### Abstract

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

## 3.1 Introduction

Out of many methods for classification, Learning Vector Quantization 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, Helsinki (2002). Prototypes in LVQ algorithms represent typical features within a data set using the same feature space instead of the black-box approach practiced in many other classifiers, e.g. feedforward neural networks or support vector machines. This makes them attractive to researchers outside the field of machine learning. Other advantages of LVQ algorithms are (1) they are easy to implement for multi-class classification tasks and (2) the algorithm complexity can be adjusted during training as required.

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

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

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

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

## 3.2 Model

Throughout the paper, 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, i.e. each cluster  $\sigma$  generates only data with one of the class labels  $y_\sigma \in \{1, 2, \dots, N_c\}$  where  $N_c$  is the number of classes. Examples  $\{\xi^\mu, y_\sigma^\mu\}$  with  $\xi^\mu \in \mathbb{R}$  are drawn independently according to the probability density function

$$P(\xi) = \sum_{\sigma=1}^M p_\sigma P(\xi|\sigma) \quad \text{with} \quad P(\xi|\sigma) = \frac{1}{(2\pi v_\sigma)^{N/2}} \exp\left[-\frac{1}{2v_\sigma}(\xi - \ell_\sigma \mathbf{B}_\sigma)^2\right] \quad (3.1)$$

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

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

We bring attention to the readers on the scaling of the model. The anisotropy of this data distribution is very weak: while the mean of cluster  $\sigma$ , 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 analyse the considered learning algorithms and Gaussian modeling of feature vectors which is a common technique in many practical scenarios. While more complex behaviors are expected in practical applications, the non-trivial effects already observed in this model will clearly influence the outcome under more general circumstances.

## 3.3 Algorithms

We shortly review LVQ algorithms and their corresponding window schemes. For the two-class model defined in Section 3.2, we define an LVQ system as a set of  $K$

prototypes  $W = \{\mathbf{w}_S, c_S\}_{S=1}^K$  with  $\mathbf{w}_S \in \mathbb{R}$  and  $c_S = \{1, 2, \dots, N_c\}$ . 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  $\xi^\mu$ . In this chapter, we investigate several LVQ prescriptions which include window schemes.

### 3.3.1 LVQ 2.1

The algorithm of LVQ 2.1 has been presented in Chapter 1 describing the update step and window used to restrain stability issues. However, this window is ineffective for very high dimensional data, as we obtain  $\lim_{N \rightarrow \infty} (\xi^\mu - \mathbf{w}_S)^2 \approx (\xi^\mu)^2$  because  $(\xi^\mu)^2 = \mathcal{O}(N)$  terms dominate the other  $\mathcal{O}(1)$ -terms, i.e.  $(\mathbf{w}_S \cdot \xi^\mu)$  and  $(\mathbf{w}_S^2)$ . Consequently, this window definition does not work in very high dimensions, evidenced by

$$\lim_{N \rightarrow \infty} \min \left( \frac{(\xi^\mu - \mathbf{w}_T^{\mu-1})^2}{(\xi^\mu - \mathbf{w}_S^{\mu-1})^2}, \frac{(\xi^\mu - \mathbf{w}_S^{\mu-1})^2}{(\xi^\mu - \mathbf{w}_T^{\mu-1})^2} \right) = 1, \quad (3.2)$$

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

$$|(\xi^\mu - \mathbf{w}_T)^2 - (\xi^\mu - \mathbf{w}_S)^2| \leq k \min((\xi^\mu - \mathbf{w}_S)^2, (\xi^\mu - \mathbf{w}_T)^2) \quad (3.3)$$

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  $N \rightarrow \infty$ . Thus, the right hand side becomes  $k \cdot (\xi^\mu)^2$  and the condition is non-trivial 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  $-\delta \leq (d_T^\mu - d_S^\mu) \leq \delta$ ;  $\delta$  is positive. We describe these rules as the following modulation function

$$f_S = \chi(c_S, y_\sigma^\mu) \sum_{T: c_T \neq c_S} (\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}) \prod_{U \neq S, T} \Theta_{SU} \Theta_{TU} \quad (3.4)$$

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 notation  $\Theta_{ji}^\delta \equiv \Theta(d_i^\mu - d_j^\mu - \delta)$ , 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 3.4.

### 3.3.2 LFM-W

The performance of LFM, as described in Chapter 1 can be improved by including data selection of data using the window rule in Equation (3.3). We refer to this algorithm as LFM-W, represented by the modulation function

$$f_S = \begin{cases} \sum_{K:c_K \neq y_\sigma} (\Theta_{KS} - \Theta_{KS}^\delta) \psi(S, K) & \text{if } c_S = y_\sigma^\mu \\ \sum_{J:c_J = y_\sigma} (\Theta_{SJ} - \Theta_{SJ}^\delta) \psi(J, S) & \text{else.} \end{cases} \quad (3.5)$$

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

### 3.3.3 GLVQ

GLVQ, as presented in Chapter 1, where  $\Phi(\tau)$  can be used to define a window around the decision boundary. Here the usefulness of selecting a non-linear  $\Phi(\tau)$  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.

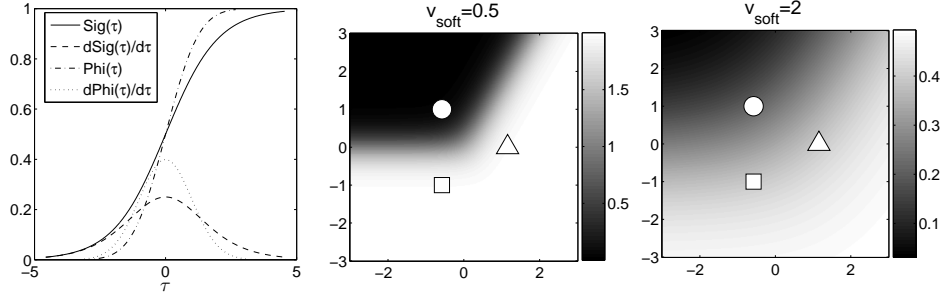
In the high dimensional limit, we notice that  $(d_J^\mu + d_K^\mu)$  is dominated by  $(\xi^\mu)^2$ -terms and, effectively, becomes a constant  $\mathcal{O}(N)$ -term:  $1/N(d_J^\mu + d_K^\mu) = 1 + \mathcal{O}(1/N)$ . Therefore the denominator term in (2.5) becomes constant:

$$\lim_{N \rightarrow \infty} E = \lim_{N \rightarrow \infty} \sum_{\mu} \Phi \left( \frac{C}{d_J^\mu + d_K^\mu} (d_J^\mu - d_K^\mu) \right) = \sum_{\mu} \Phi \left( \frac{1}{v_G} (d_J^\mu - d_K^\mu) \right). \quad (3.6)$$

To obtain a non-zero 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 non-linear  $\Phi(\tau)$  is chosen. Note that GLVQ can be simplified to LVQ 2.1 without a window using the identity function  $\Phi(\tau) = \tau$ . The cost function in (2.5) 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 chapter, we choose the cumulative normal distribution

$$\Phi(\tau) = \int_{-\infty}^{\tau} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt \quad (3.7)$$



**Figure 3.1:** Left panel: The form of the chosen  $\Phi(\tau)$  in GLVQ, in comparison to the sigmoidal function  $\text{Sig}(\tau)$ . The derivatives produce a soft window. Middle and right panel: The RSLVQ modulation function  $f_S$  for class 1 (o) when presented with data from class 1. The figures display the difference between smaller  $v_{\text{soft}}$  (left) and larger  $v_{\text{soft}}$  (right).

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 3.1 for the comparison.

Plugging in the form of (3.6), we obtain the learning rules

$$f_J = +\frac{2}{v_G} \phi\left(\frac{d_J - d_K}{v_G}\right), \quad f_K = -\frac{2}{v_G} \phi\left(\frac{d_J - d_K}{v_G}\right) \quad (3.8)$$

We can write the modulation function as

$$f_S = \begin{cases} \sum_{K:c_K \neq y_\sigma} \left( \frac{2}{v_G} \phi\left(\frac{d_S - d_K}{v_G}\right) \right) \psi(S, K) & \text{if } c_S = y_\sigma^\mu \\ - \sum_{J:c_J = y_\sigma} \left( \frac{2}{v_G} \phi\left(\frac{d_J - d_S}{v_G}\right) \right) \psi(J, S) & \text{else.} \end{cases} \quad (3.9)$$

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

### 3.3.4 RSLVQ

In addition to the description of RSLVQ in Chapter 1, we can make the following observations: As  $v_{\text{soft}}$  becomes smaller, the updates become smaller for correctly classified examples and larger for incorrectly classified examples, see Figure 3.1.

Note that the limiting case of  $v_{\text{soft}}$  is particularly simple. The assignments of Equation (2.12) become hard assignments, i.e.

$$P_\sigma(S|\xi^\mu) = \begin{cases} 1, & \text{if } d_S^\mu = \min_{\{j:c_j=\sigma^\mu\}} \{d_j^\mu\} \\ 0, & \text{else} \end{cases}, \quad P(S|\xi^\mu) = \begin{cases} 1, & \text{if } d_S^\mu = \min_{\{j\}} \{d_j^\mu\} \\ 0, & \text{else} \end{cases} \quad (3.10)$$

Plugging the above into (2.11), we obtain the learning rule for LFM, described in Section 3.3.2.

### 3.4 Analysis

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

$$R_{S\sigma}^\mu = \mathbf{w}_S^\mu \cdot \mathbf{B}_\sigma \quad Q_{ST}^\mu = \mathbf{w}_S^\mu \cdot \mathbf{w}_T^\mu. \quad (3.11)$$

Note that  $R_{S\sigma}$  are the projections of prototype vectors  $\mathbf{w}_S^\mu$  on the center vectors  $\mathbf{B}_\sigma$  and  $Q_{ST}^\mu$  correspond to the self- and cross- overlaps of the prototype vectors. From the generic update rule defined above, Equation (2.1), we can derive the following recursions in terms of the order parameters:

$$\begin{aligned} \frac{R_{S\sigma}^\mu - R_{S\sigma}^{\mu-1}}{1/N} &= \eta f_S(b_\sigma^\mu - R_{S\sigma}^{\mu-1}) \\ \frac{Q_{ST}^\mu - Q_{ST}^{\mu-1}}{1/N} &= \eta [f_T(h_S^\mu - Q_{ST}^{\mu-1}) + f_S(h_T^\mu - Q_{ST}^{\mu-1})] + \eta^2 \frac{f_S f_T(\xi^\mu)^2}{N} + \mathcal{O}\left(\frac{1}{N}\right) \end{aligned} \quad (3.12)$$

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

$$h_S^\mu = \mathbf{w}_S^{\mu-1} \cdot \xi^\mu \quad b_\sigma^\mu = \mathbf{B}_\sigma \cdot \xi^\mu. \quad (3.13)$$

In the limit  $N \rightarrow \infty$ , the  $\mathcal{O}(1/N)$  term can be neglected and the order parameters *self average* (Reents and 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  $N \rightarrow \infty$ , 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 et al. 2006) after performing an average over the sequence of input data:

$$\begin{aligned}\frac{dR_{S\sigma}}{d\alpha} &= \eta(\langle b_\sigma f_S \rangle - \langle f_S \rangle R_{S\sigma}) \\ \frac{dQ_{ST}}{d\alpha} &= \eta(\langle h_S f_T \rangle - \langle f_T \rangle Q_{ST} + \langle h_T f_S \rangle - \langle f_S \rangle Q_{ST}) + \eta^2 \sum_{\sigma} p_\sigma v_\sigma \langle f_S f_T \rangle\end{aligned}\quad (3.14)$$

where  $\langle \cdot \rangle$  and  $\langle \cdot \rangle_\sigma$  are the averages over the density  $P(\xi)$  and  $P(\xi|\sigma)$ . To simplify the last term of Equation (3.14), we used

$$\lim_{N \rightarrow \infty} \langle f_S f_T \xi^2 \rangle / N = \lim_{N \rightarrow \infty} \sum_{\sigma} p_\sigma (v_\sigma N + \ell^2) \langle f_S f_T \rangle_\sigma / N = \sum_{\sigma} p_\sigma v_\sigma \langle f_S f_T \rangle_\sigma.$$

In various sections in this paper, we investigate learning behaviors using small learning rates  $\eta \rightarrow 0$  and neglect the  $\eta^2$  terms in Equation (3.14). Non trivial behavior is only expected by taking the simultaneous limit  $\eta \rightarrow 0, \alpha \rightarrow \infty$  and rescaling  $\tilde{\alpha} = \eta\alpha$  in Equation (3.14).

Exploiting the limit  $N \rightarrow \infty$  once more, the quantities  $h_S^\mu, b_\sigma^\mu$  become correlated Gaussian quantities by means of the Central Limit Theorem. Therefore, they are fully specified by first and second moments, detailed in Appendix 3.A:

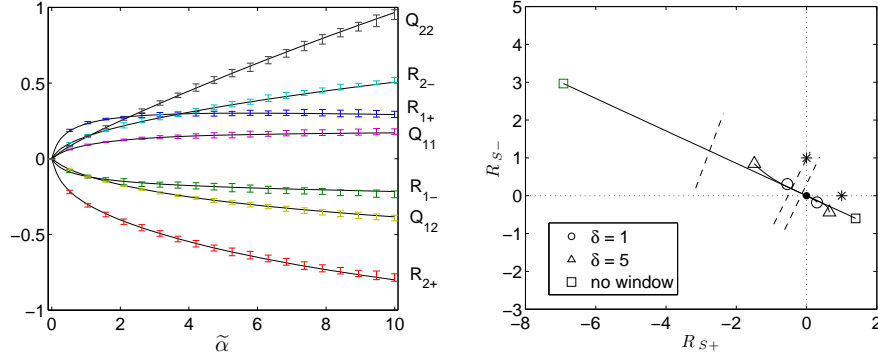
$$\begin{aligned}\langle h_S^\mu \rangle_\sigma &= \ell_\sigma R_{S\sigma}^{\mu-1}, \quad \langle b_\tau^\mu \rangle_\sigma = \ell_\sigma \delta_{\tau\sigma}, \quad \langle h_S^\mu h_T^\mu \rangle_\sigma - \langle h_S^\mu \rangle_\sigma \langle h_T^\mu \rangle_\sigma = v_\sigma Q_{ST}^{\mu-1} \\ \langle b_\tau^\mu b_\rho^\mu \rangle_\sigma - \langle b_\tau^\mu \rangle_\sigma \langle b_\rho^\mu \rangle_\sigma &= v_\sigma \mathbf{T}_{\tau\rho}, \quad \langle h_i^\mu b_\tau^\mu \rangle_\sigma - \langle h_i^\mu \rangle_\sigma \langle b_\tau^\mu \rangle_\sigma = v_\sigma R_{i\tau}^{\mu-1}.\end{aligned}\quad (3.15)$$

where  $S, T$  are prototype indices,  $\tau, \rho, \sigma$  are cluster indices,  $\delta$  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 3.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  $\epsilon_g$ , 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:

$$\epsilon_g = \sum_{\sigma=1}^M p_\sigma \epsilon_{g,\sigma} \quad \text{with} \quad \epsilon_{g,\sigma} = \sum_{S:c_S \neq y_\sigma}^K \left\langle \prod_{T \neq S}^K \Theta_{ST} \right\rangle_\sigma, \quad (3.16)$$

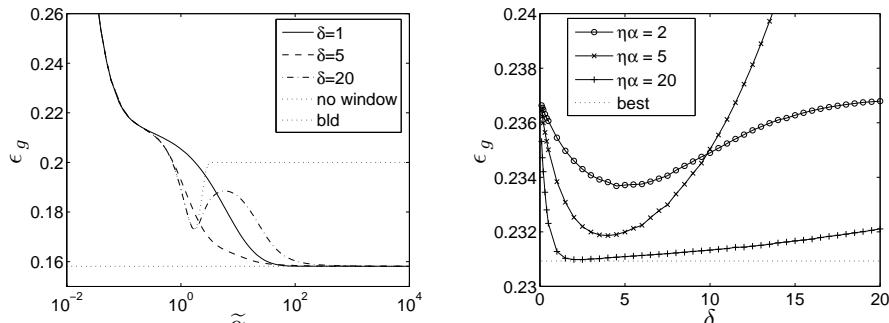


**Figure 3.2:** Left panel: Evolution of the order parameters  $\{R_{S\sigma}, Q_{ST}\}$  for LVQ 2.1 with  $K = 2$ ,  $M = 2$ ,  $\ell_1 = \ell_2 = 1$ ,  $p_1 = 0.7$ ,  $v_1 = v_2 = 1$  and learning parameters  $\eta = 0.1$  and  $\delta = 1$ . Solid lines represent  $\{R_{S\sigma}, Q_{ST}\}$  obtained from the theoretical analysis, while bars represent the variance as produced by Monte Carlo simulations for  $N = 100$  over 100 independent runs. Right panel: Influence of a window on LVQ 2.1 at learning time  $\alpha = 40$ . Prototypes are projected on the  $(\mathbf{B}_+, \mathbf{B}_-)$  subspace for  $\delta = 1$  ( $\circ$ ),  $\delta = 5$  ( $\triangle$ ) and unrestricted LVQ 2.1 ( $\square$ ). In the latter, one prototype strongly diverges. The resulting decision boundaries are indicated by chained lines. The origin is marked by ( $\cdot$ ) and the cluster centers are marked by ( $*$ ).

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

$$\epsilon_{g,\sigma} = \Phi\left(\frac{Q_{\sigma\sigma} - Q_{-\sigma,-\sigma} - 2\ell_\sigma(R_{\sigma,\sigma} - R_{-\sigma,\sigma})}{2\sqrt{v_\sigma}\sqrt{Q_{\sigma\sigma} - 2Q_{\sigma,-\sigma} + Q_{-\sigma,-\sigma}}}\right), \quad (3.17)$$

with  $\Phi(x) = \int_{-\infty}^x dt \frac{1}{\sqrt{2\pi}} \exp(-t^2/2)$ , detailed in Appendix 3.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 3.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 3.2.



**Figure 3.3:** Generalization error  $\epsilon_g$  for LVQ 2.1 with  $K = 2, M = 2, \ell = 1, p_+ = 0.8, p_- = 0.2$  and  $\eta \rightarrow 0$ . Left panel:  $\epsilon_g$  vs  $\tilde{\alpha}$  using  $\delta = 1, 5, 20$  and without a window. Note the logarithmic scaling on the horizontal axis. The asymptotic errors for all settings of  $\delta$  converge at  $\epsilon_g^{bld}$ , indicated by the dotted line. Right panel:  $\epsilon_g$  at fixed learning times  $\tilde{\alpha} = 2, 5$  and 20 as a function of  $\delta$ .

### 3.5 A simple case: two prototypes, two clusters

In this section we discuss in detail the results of the analysis for the simplest non-trivial problem: two-prototype LVQ 2.1, GLVQ, LFM-W and RSLVQ systems and  $M = 2$  with one Gaussian cluster per class. The model data is given in Section 3.2. For simplicity, we denote the two clusters as  $\sigma = \{+, -\}$  and without loss of generality can choose  $\ell_+ = \ell_- = \ell$  and orthonormal  $\mathbf{B}_\sigma$ , i.e.  $\mathbf{B}_i \cdot \mathbf{B}_j = 1$  if  $i = j$ ; 0 else.

We place an emphasis on the asymptotic behavior in the limit  $\alpha \rightarrow \infty$ , i.e. the achieved performance for an arbitrarily large number of examples. The asymptotic generalization error  $\epsilon_g(\infty)$  scales with the learning rate, analogous to minimizing a cost function in stochastic gradient descent procedures. For LVQ 2.1 and RSLVQ, the best achievable generalization error is obtained in the simultaneous limit of small learning rates  $\eta \rightarrow 0, \alpha \rightarrow \infty$  and rescaling  $\tilde{\alpha} = \eta\alpha \rightarrow \infty$ . 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 (BLD) boundaries by linear approximation of the Bayesian optimal decision boundary, see Biehl et al. (2004) for the calculations. We compare the results from each algorithm to the best linearly achievable error  $\epsilon_g^{bld}$ .

### 3.5.1 LVQ 2.1

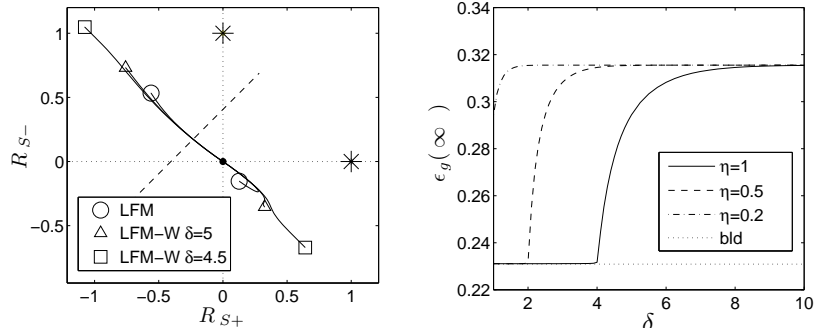
We first examine two-prototype systems, i.e.  $K = 2$ . Figures 3.2 illustrate the evolution of order parameters under the influence of a window and the trajectories of the prototypes projected onto the  $(\mathbf{B}_+, \mathbf{B}_-)$  subspace. Without additional constraints, LVQ 2.1 with two prototypes displays a strong divergent behavior in a system with unbalanced data, i.e.  $p_+ \neq p_-$ . 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  $\tilde{\alpha}$ . As  $\tilde{\alpha} \rightarrow \infty$ ,  $\mathbf{w}_2$  will be arbitrarily far away from the cluster centers and the asymptotic generalization error is trivial,  $\epsilon_g(\infty) = \min(p_+, p_-)$ .

Implementing the window scheme,  $\mathbf{w}_2$  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, viz.  $R_{S\sigma}$  for both prototypes display a linear change with  $\tilde{\alpha}$  at large  $\tilde{\alpha}$ , but the decision boundary remains stable. Trivial classification is prevented, see the generalization error curves  $\epsilon_g$  vs.  $\tilde{\alpha}$  in the left panel of Figure 3.3. Obviously, for smaller  $\delta$  a considerable amount of data is filtered out and the initial learning stages slow down significantly. Meanwhile for large  $\delta$ ,  $\epsilon_g$  becomes non-monotonic and converges more slowly.

Hence the performance at finite  $\tilde{\alpha}$  is dependent on  $\delta$ , displayed in Figure 3.3, and parameter settings are highly critical in practical applications. Given learning time  $\tilde{\alpha}$ , an optimal choice of fixed  $\delta$  exists, which clearly depends on the properties of the data. With larger  $\tilde{\alpha}$ ,  $\epsilon_g$  becomes less sensitive towards  $\delta$  and the optimal setting of  $\delta$  is smaller. Surprisingly,  $\delta$  only influences the convergence speed while the non-trivial asymptotic generalization error  $\epsilon_g(\infty)$  is insensitive to the choice of  $\delta$  and equals the best achievable error  $\epsilon_g^{blid}$  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 (3.11) and the orthonormality of  $\mathbf{B}_+$  and  $\mathbf{B}_-$ , as

$$\varphi = \arccos \left( \frac{R_{1+} + R_{1-} - R_{2+} + R_{2-}}{\sqrt{2}(Q_{11} - 2Q_{12} + Q_{22})} \right), \quad (3.18)$$

which is found to be zero for large  $\tilde{\alpha}$ . Hence, the decision boundary becomes parallel to the BLD and only its offset produces the difference between  $\epsilon_g(\infty)$  and  $\epsilon_g^{blid}$ . 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.



**Figure 3.4:** 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  $\mathbf{w}_1, \mathbf{w}_2$  lie 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  $\delta$ . The lines correspond to learning rates  $\eta = 0.2, 0.5$  and  $1.0$ .

### 3.5.2 LFM-W

The LFM scheme performs updates identical to LVQ 2.1 with the condition that the example is misclassified. A detailed investigation into the characteristics of  $K = 2$  unrestricted LFM has been presented in Biehl et al. (2007). There, it was shown that LFM produces stable prototype configurations for finite learning rates  $\eta$ . The projection of the prototypes lies parallel to the symmetry axis  $\ell(\mathbf{B}_+ - \mathbf{B}_-)$ , displayed in Figure 3.4. 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

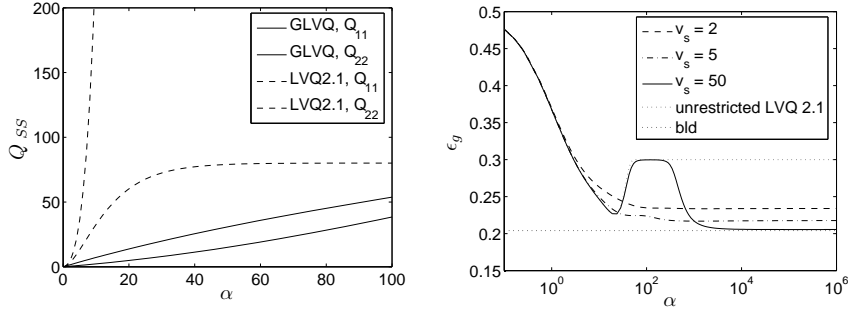
$$|\mathbf{w}_S|^2 > |R_{S+}\mathbf{B}_+ + R_{S-}\mathbf{B}_-|^2.$$

The asymptotic generalization error  $\epsilon_g(\infty)$  is suboptimal and insensitive to  $\eta$ : the asymptotic decision boundary remains at an angle  $\varphi$  from the optimal hyperplane, c.f. Equation (3.18), independent of  $\eta$ . The Euclidean distance between prototypes is given by the quantity

$$\Delta_q = \sqrt{(\mathbf{w}_1 - \mathbf{w}_2)^2} = \sqrt{Q_{11} - 2Q_{12} + Q_{22}}, \quad (3.19)$$

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

In this analysis, we observe that window schemes can dramatically improve performance of LFM. Using a window, the tilt of the decision boundary from the op-



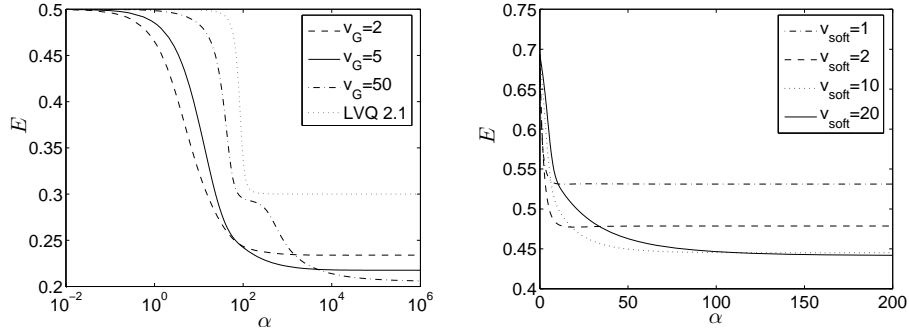
**Figure 3.5:** Left panel:  $Q_{11}$  and  $Q_{22}$  for GLVQ (solid lines), compared to unrestricted LVQ 2.1 (dashed lines). The soft window of GLVQ slows down the repulsion of one prototype, but the prototypes remain divergent. Here  $p_+ = 0.7, \ell = 1, v_+ = 2, v_- = 5, \eta = 0.25$ . Right panel: Learning curves  $\epsilon_g$  vs.  $\alpha$  for softness  $v_G = 2, 5$  and  $50$ , note the logarithmic horizontal axis. The learning rates are maintained at  $\eta/v_G = 0.1$ . Large  $v_G$  produces better asymptotic generalization error, but may exhibit non-monotonic behavior and require very long learning times.

timal hyperplane, i.e  $\varphi$  in (3.18), is reduced, resulting in lower  $\epsilon_g(\infty)$ . We observe that  $\epsilon_g(\infty)$  decreases along with reducing  $\delta$ , displayed in the right panel of Figure 3.4. However, a critical window size  $\delta_c$  exists where the LFM unexpectedly becomes divergent and no stationary state exists. Smaller windows filter examples which produce more repulsion in the orientation of the cluster centers, and we observe asymptotically larger  $\Delta_q$  as  $\delta$  decreases. This is clearly observed in Figure 3.4. Given a sufficiently small  $\delta$ , it is possible that the repulsion factor entirely outweighs the attractive factor. At  $\delta < \delta_c$ , it performs similar to LVQ 2.1: the angle  $\varphi$  becomes zero and  $\epsilon_g(\infty)$  is close to the best achievable error.

Unlike the unrestricted case, the learning rate  $\eta$  can influence the asymptotic performance. The learning rate and window size are indirectly related, as shown in the right panel of Figure 3.4. For example, learning with small learning rates requires smaller windows to achieve optimal asymptotic error. Note that the influence of the window size depends heavily on the structure of the data. For various data models, efficient window settings may only exist on a very limited range and window schemes may be ineffective to improve generalization performance while still maintaining stability.

### 3.5.3 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



**Figure 3.6:**  $p_+ = 0.7, \ell = 1, v_+ = 1, v_- = 1$ . The cost functions for GLVQ with  $\eta \rightarrow 0$  (left panel) and RSLVQ with  $\eta/v_{\text{soft}} = 1$  (right panel) decrease monotonically, corresponding to a stochastic gradient descent.

large window. Figure 3.5 displays the prototype lengths while using GLVQ: the soft window slows down the strong repulsion of the prototype of the weaker cluster, as opposed to unrestricted LVQ 2.1. While both prototypes still diverge because the cost function at  $N \rightarrow \infty$  is not bounded, c.f. (3.6), the asymptotic  $\epsilon_g$  remains non-trivial, see Figure 3.5.

Note that  $v_G$  directly relates to the overall learning rate  $\eta/v_G$ , refer to Equation (3.9), which influences the level of noise in stochastic gradient procedures. We compare results with respect to  $v_G$ , while maintaining at equal overall learning rate by keeping  $\eta/v_G$  constant, in Figure 3.5. Performance deteriorates at smaller  $v_G$ , where training slows down at intermediate stages and converges at a higher error. However, very large  $v_G$  allows strong repulsion of the weaker prototype which results in non-monotonic  $\epsilon_g$  and long learning convergence times. Surprisingly, the soft GLVQ window is outperformed by the simple hard or crisp window of LVQ 2.1. This is caused by the long tail of the modulation function which sums up into a large repulsion, whereas in the crisp window, only data near the decision boundary are considered.

Figure 3.6 displays the cost function during learning. In the initial learning stages, the minimization of the cost function  $E$  leads to fast decrease of  $\epsilon_g$ . However, while the cost function continues to decrease monotonically,  $\epsilon_g$  behaves non-monotonically. While many techniques are developed to improve minimization procedures of  $E$ , it is important to evaluate the choice of  $E$  and its correlation to the desired generalization performance.

### 3.5.4 RSLVQ

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

We first investigate model scenarios with equal variance clusters  $v_+ = v_-$  and unbalanced data  $p_+ \neq p_-$ . We observe the influence of  $v_{\text{soft}}$  on the learning curves, displayed on the left panel of Figure 3.7. The generalization error curve depends on  $v_{\text{soft}}$ : at large  $v_{\text{soft}}$ ,  $\epsilon_g$  may exhibit non-monotonic behavior, reminiscent of LVQ 2.1. Because of this behavior, the learning process may require long learning times before reaching the asymptotic configuration. This is an important consideration for practical applications which often uses early stopping strategies to avoid overtraining. Meanwhile, the algorithm minimizes the cost function  $E$  in (2.10) monotonically, see Figure 3.6. Thus, the decrease in  $E$  does not always result in a decrease of  $\epsilon_g$ .

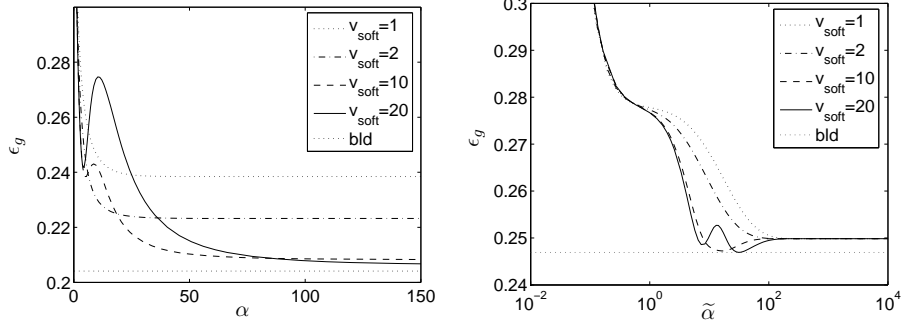
A major advantage of the RSLVQ algorithm is the convergence of prototypes, i.e. a stationary configuration of order parameters exists for finite  $v_{\text{soft}}$ . The asymptotic configuration of prototypes are displayed in Figure 3.8. At  $\tilde{\alpha} \rightarrow \infty$ , the softness parameter controls only the distance between the two prototypes:  $\Delta_q$  as defined in Equation (3.19), decreases linearly with  $v_{\text{soft}}$ . Note that under the conditions  $p_+ = 0.5, v_{\text{soft}} = v_+ = v_-$  and initialization of prototypes on the symmetry axis, each prototype is located at its corresponding cluster center, i.e. the RSLVQ mixture model matches exactly to the actual input density.

Figures 3.7 compare the asymptotic errors in the case of  $\eta/v_{\text{soft}} = 1$  (left panel) and small learning rates  $\eta/v_{\text{soft}} \rightarrow 0$  (right panel). In the former case, performance improves with large  $v_{\text{soft}}$ : at small  $v_{\text{soft}}$ , the system converges at high  $\epsilon_g$  similar to LFM, while at larger  $v_{\text{soft}}$ , it approaches the best linear decision. Meanwhile, at small learning rates, the asymptotic error becomes independent to  $v_{\text{soft}}$ . Therefore, given sufficiently small learning rates, RSLVQ becomes robust wrt. its softness parameter.

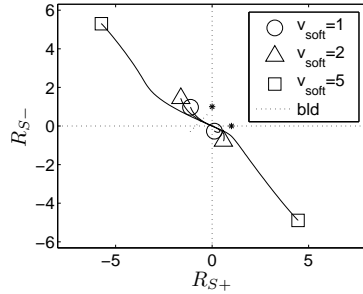
In the equal variance scenario, the asymptotic decision boundary always converges to the best linear decision boundary for all settings of  $\{p_+, p_-\}$  and RSLVQ outperforms both LFM and LVQ 2.1, as it provides robustness, stability and low generalization error.

On the other hand, a scenario with unequal class variances presents an interesting case where RSLVQ with global  $v_{\text{soft}}$  fails to match the model. RSLVQ remains robust, i.e. the decision boundary converges to identical configurations for all settings of  $v_{\text{soft}}$ , see Figure 3.8. However, the asymptotic results are suboptimal. While



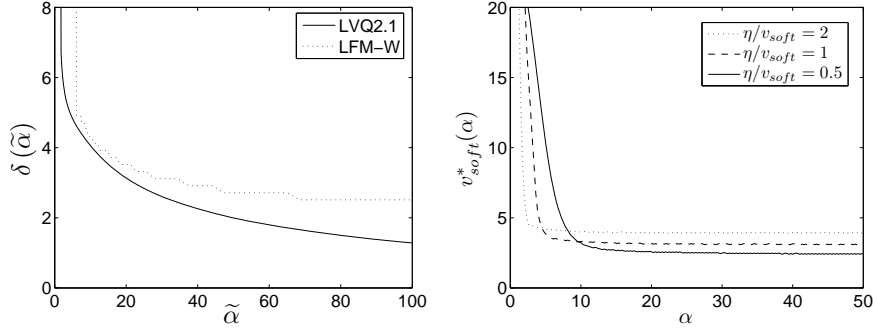


**Figure 3.7:** Learning curves  $\epsilon_g$  for RSLVQ using softness parameter  $v_{\text{soft}} = 1, 2, 10$  and  $20$ . Left:  $p_+ = 0.7$  and equal variance  $v_+ = v_- = 1$  with fixed overall learning rate  $\eta/v_{\text{soft}} = 1$ . Right:  $p_+ = 0.6$  and unequal variance  $v_+ = 1, v_- = 4$  with  $\eta/v_{\text{soft}} \rightarrow 0$ . The asymptotic error is independent of  $v_{\text{soft}}$  at small learning rates, but at a suboptimal value. Note the logarithmic scale of  $\tilde{\alpha}$ .



**Figure 3.8:** Trajectories of prototypes of the system in the left panel of Figure 3.7. Prototypes are projected on the space  $\text{Span}(\mathbf{B}_+, \mathbf{B}_-)$  for  $v_{\text{soft}} = 1$  (circle),  $2$  (triangle) and  $5$  (square).

RSLVQ is insensitive to the priors of the clusters, its performance wrt. the best achievable error is sensitive to the cluster variances, e.g. at highly unbalanced  $\sigma_+/\sigma_-$ , RSLVQ generalizes poorly and is outperformed by the simpler LVQ 2.1. In practical applications,  $v_{\text{soft}}$  may be set locally for each prototype to accommodate such scenarios, but this case cannot be treated along the lines of the present analysis in a straightforward way.



**Figure 3.9:** 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 panel: optimal softness parameter for RSLVQ with fixed  $\eta/v_{\text{soft}} = 2, 1$  and  $0.5$ .

### 3.6 Optimal window schedules

We have observed in Sections 3.5.1 and 3.5.2 the learning curves and asymptotics of LVQ 2.1 and LFM-W wrt. fixed window parameters. In this section we treat the window parameter as dynamic properties during learning, viz.  $\delta(\alpha)$ . Although small windows allow optimal  $\epsilon_g(\alpha \rightarrow \infty)$ , their obvious disadvantage is their slower initial learning and convergence speed. This suggests that dynamic performance can be improved by adjusting the window along with the number of examples presented.

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

$$\delta^*(\alpha) \arg \min_{\delta} \left( \mathbf{u}(\alpha) \cdot \frac{d\mathbf{O}(\alpha)}{d\alpha} \right) = 0 \quad \text{with} \quad \mathbf{u}(\alpha) = \sum_{\sigma=1}^M p_{\sigma} \frac{d\epsilon_{g,\sigma}(\alpha)}{d\mathbf{O}} \quad (3.20)$$

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  $\sigma = \{+, -\}$ ,  $\mathbf{O} = \{R_{++}, R_{+-}, R_{-+}, R_{--}, Q_{++},$

$Q_{+-}, Q_{--}\}^T$  and derivating from (3.17), we obtain

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

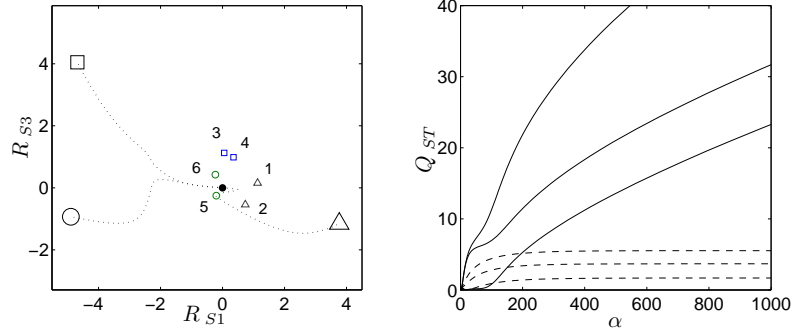
$$\mathbf{A}_+ = \begin{bmatrix} -2\ell \\ 0 \\ +2\ell \\ 0 \\ 1 - Z_+/(2\Delta_q^2) \\ Z_+/\Delta_q^2 \\ -1 - Z_+/(2\Delta_q^2) \end{bmatrix}, \quad \mathbf{A}_- = \begin{bmatrix} 0 \\ +2\ell \\ 0 \\ -2\ell \\ -1 - Z_-/(2\Delta_q^2) \\ Z_-/\Delta_q^2 \\ 1 - Z_-/(2\Delta_q^2) \end{bmatrix}$$

with  $Z_\sigma = Q_{\sigma\sigma} - Q_{-\sigma,-\sigma} - 2\ell(R_{\sigma\sigma} - R_{-\sigma\sigma})$  and  $\Delta_q$  defined in Equation (3.19), see Appendix 3.D for the calculations.

We plug in  $d\mathbf{O}/d\alpha$  for the corresponding algorithm and numerically calculate  $\delta^*(\alpha)$  from Equation (3.20) at each learning step. We find that the learning curve is improved with initially large  $\delta$  which is decreased during training, following the curve in Figure 3.9. This suggests that practical schedules with gradual reduction of window sizes are indeed suitable for this particular learning problem.

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

Figure 3.7 displays that although large  $v_{\text{soft}}$  for RSLVQ allows for a faster initial learning, it also can yield non-monotonic learning curves. We can avoid the non-monotonic behavior and maximize the decrease of  $\epsilon_g$  by applying a variational approach analogous to (3.20) in order to calculate the locally optimal softness parameter schedule  $v_{\text{soft}}^*(\alpha)$ . While fixing the value of  $\eta/v_{\text{soft}}$ , we produce the locally optimal softness schedule  $v_{\text{soft}}^*(\alpha)$  in Figure 3.9, where  $v_{\text{soft}}^*(\alpha)$  is initially large and decreases to saturate at a constant value. Note that this value depends on the learning rate, e.g. it decreases with  $\eta/v_{\text{soft}}$ . In calculations with  $\eta \rightarrow 0$ , we obtain the limit  $v_{\text{soft}}^*(\infty) \rightarrow 0$ , which is the clearly suboptimal LFM. Therefore an analysis of optimal RSLVQ schedule requires  $\eta > 0$ .

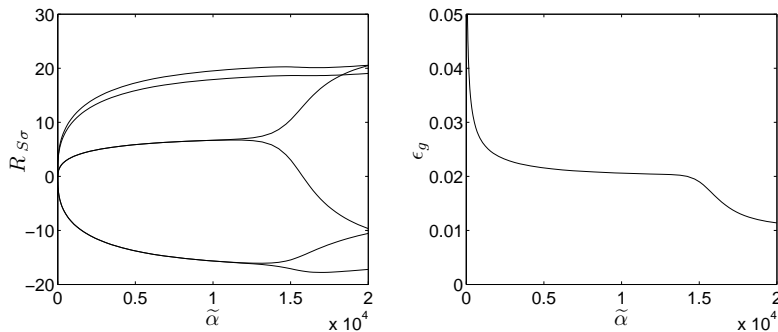


**Figure 3.10:** Left panel: Snapshot at  $\alpha = 50$  of an LVQ 2.1 system,  $\delta = 1$  with  $K = 3$  and  $M = 6$  randomly generated isotropic clusters projected on the  $(\mathbf{B}_1, \mathbf{B}_3)$  subspace. The solid dot marks the initial position of all prototypes and solid lines mark the trajectories of the prototypes. Right panel:  $p_{\Delta} = 0.5, p_{\square} = 0.3, p_{\circ} = 0.2$ . Solid lines represent, from bottom to top, prototype vector lengths  $Q_{11}, Q_{22}, Q_{33}$  for LVQ 2.1  $\delta = 10$ . Dashed lines represent the result for RSLVQ  $v_{soft} = 2$ .

### 3.7 Three-prototype systems

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

We first look at multi-class problems with  $N_c = 3$  classes, an example is shown in Figure 3.10 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 observations, we find that the behaviors of  $K = 3$  systems are qualitatively similar to  $K = 2$  systems. For LVQ 2.1, the learning curves vary according to the window sizes, but its asymptotic generalization error is independent of  $\delta$ . Due to the presence of other prototypes, the repulsion on a weaker class prototype are reduced. However, the prototypes remain divergent, e.g. Figure 3.10. Meanwhile for LFM-W, the asymptotic performance is sensitive to  $\delta$  whose range of effective window sizes depend strongly on the learning parameters. For GLVQ, the prototypes are divergent with a higher asymptotic error than LVQ 2.1, and thus it performs poorly. Finally, for RSLVQ, the prototypes remain stable and the asymptotic generalization performance is robust wrt. settings of  $v_{soft}$ , but it is



**Figure 3.11:** *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 panel: Several order parameters display a specialization phase between prototypes of same class. Right panel: Generalization error.*

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 non-trivial interaction between similarly labeled prototypes, here  $w_1$  and  $w_2$ . While prototypes of different classes immediately separate in the initial training phase, prototypes of the same class remain identical in the  $M$  dimensional space, see Figure 3.11. The latter prototypes differ only in dimensions which are not related for classification and produce a suboptimal decision boundary. This may proceed for a long learning period before these prototypes begin to *specialize*, i.e. each prototype produces a bigger overlap  $R_{S\sigma}$  with a distinct group of clusters. The specialization phase produces a sudden decrease of  $\epsilon_g$ , displayed in the right panel of Figure 3.11. This phenomenon is highly reminiscent of symmetry breaking effects observed in unsupervised learning, such as Winner-Takes-All VQ (Biehl 1994, Witoelar et al. 2008) or multilayer neural networks (Saad and Solla 1995).

Learning parameters highly influence the nature of the transition, e.g. large learning rates and smaller windows prolong the unspecialized phase, and therefore they are critical to the success of learning. Symmetry breaking may require exceedingly long learning times, resulting in learning plateaus which dominate the training process

and present a challenge in practical situations with very high dimensional data. In more extreme circumstances, the system may not escape the unspecialized state at all and the optimal classification cannot be obtained. Details of the symmetry breaking properties wrt. parameters will be investigated in following publications.

### 3.8 Conclusion

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

Our analysis demonstrates the influence of windows on the learning curves and the advantages and drawbacks of each algorithm within the model scenarios. A summary is described in Table 3.1. Asymptotically, LVQ 2.1 achieves optimal performance in all scenarios, but stability remains an issue in terms of diverging prototypes. LFM-W shows a remarkable improvement in performance over LFM. Unfortunately, the introduction of a window may also influence its stability, and therefore it is highly parameter sensitive, i.e., only a narrow range of window size can improve the overall performance. GLVQ behaves similarly to LVQ 2.1. While GLVQ reduces the initial strong overshooting of LVQ 2.1, the prototypes remain divergent and GLVQ produces higher generalization errors or long convergence times. RSLVQ attempts to combine the advantages of both LFM and LVQ 2.1 by providing both stability and optimal performance. However, an important issue of RSLVQ lies on its approximation of the data structure, e.g., it performs well when the actual input density are isotropic Gaussian clusters with equal variance. If the assumptions depart from the input density, the results become suboptimal and RSLVQ can even be outperformed by the simpler LVQ 2.1 and LFM-W. In all scenarios, RSLVQ displays robustness of its classification behavior with respect to the softness parameter, given sufficiently low learning rates.

This analysis also allows a formal optimization of the window size during 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 sche-

**Table 3.1:** Asymptotic properties of LVQ algorithms.

	LVQ 2.1	LFM-W	GLVQ	RSLVQ
Stability	divergent	convergent*	divergent	convergent
Sensitivity wrt. parameters	robust	dependent	dependent	robust
Gen. ability	optimal	suboptimal	suboptimal	suboptimal

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

dules see, for instance, Saad and Rattray (1997). In further work, we will develop efficient dynamic parameter adaptations, i.e., optimal window schedules during on-line training along the lines of variational optimization.

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

### 3.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 (3.13) into a  $D$ -dimensional vector, where  $D = K + M$ , as

$$\mathbf{x} = ( h_1^\mu \quad h_2^\mu \quad \dots \quad h_K^\mu \quad b_1^\mu \quad b_2^\mu \quad \dots \quad b_M^\mu )^T \quad (3.21)$$

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

### 3.A.1 First order statistics

We compute the averages of the components of  $\mathbf{x}$  as follows:

$$\langle h_i \rangle_\sigma = \int_{\mathbb{R}^N} \xi \cdot \mathbf{w}_i p(\xi|\sigma) d\xi = \mathbf{w}_i \cdot \int_{\mathbb{R}^N} \xi p(\xi|\sigma) d\xi = \mathbf{w}_i \cdot \ell_\sigma \mathbf{B}_\sigma = \ell_\sigma R_{i\sigma} \quad (3.22)$$

$$\langle b_\tau \rangle_\sigma = \int_{\mathbb{R}^N} \xi \cdot \mathbf{B}_\tau p(\xi|\sigma) d\xi = \mathbf{B}_\tau \cdot \int_{\mathbb{R}^N} \xi p(\xi|\sigma) d\xi = \mathbf{B}_\tau \cdot \ell_\sigma \mathbf{B}_\sigma = \ell_\sigma T_{\tau\sigma} \quad (3.23)$$

with  $T_{\tau\sigma} = \mathbf{B}_\tau \cdot \mathbf{B}_\sigma$ . To a large extent, we utilize orthonormal cluster center vectors, i.e.  $\mathbf{B}_\tau \cdot \mathbf{B}_\sigma = \delta_{\tau\sigma}$  where  $\delta$  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

$$\mu = \ell_\sigma \left( R_{1\sigma} \ R_{2\sigma} \ \dots \ R_{K\sigma} \ T_{1\sigma} \ T_{2\sigma} \ \dots \ T_{M\sigma} \right)^T \quad (3.24)$$

### 3.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

$$\begin{aligned} \langle h_i h_j \rangle_\sigma &= \left\langle \left( \sum_{k=1}^N (\mathbf{w}_i)_k (\xi)_k \right) \left( \sum_{l=1}^N (\mathbf{w}_j)_l (\xi)_l \right) \right\rangle_\sigma \\ &= \left\langle \sum_{k=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_k (\xi)_k (\xi)_k + \sum_{k=1}^N \sum_{l=1, l \neq k}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_l (\xi)_k (\xi)_l \right\rangle_\sigma \\ &= \sum_{k=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_k \langle (\xi)_k (\xi)_k \rangle_\sigma + \sum_{k=1}^N \sum_{l=1, l \neq k}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_l \langle (\xi)_k (\xi)_l \rangle_\sigma \\ &= \sum_{k=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_k (v_\sigma + \ell_\sigma^2 (\mathbf{B}_\sigma)_k (\mathbf{B}_\sigma)_k) + \sum_{k=1}^N \sum_{l=1, l \neq k}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_l \ell_\sigma^2 (\mathbf{B}_\sigma)_k (\mathbf{B}_\sigma)_l \\ &= v_\sigma \sum_{k=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_k + \ell_\sigma^2 \sum_{k=1}^N \sum_{l=1}^N (\mathbf{w}_i)_k (\mathbf{w}_j)_l (\mathbf{B}_\sigma)_k (\mathbf{B}_\sigma)_l \\ &= v_\sigma \mathbf{w}_i \cdot \mathbf{w}_j + \ell_\sigma^2 (\mathbf{w}_i \cdot \mathbf{B}_\sigma) (\mathbf{w}_j \cdot \mathbf{B}_\sigma) = v_\sigma Q_{ij} + \ell_\sigma^2 R_{i\sigma} R_{j\sigma} \end{aligned} \quad (3.25)$$

Here we exploit the following

$$\begin{aligned} \langle (\xi)_k (\xi)_k \rangle_\sigma &= v_\sigma + \langle (\xi)_k \rangle_\sigma \langle (\xi)_k \rangle_\sigma = v_\sigma + \ell_\sigma^2 (\mathbf{B}_\sigma)_k (\mathbf{B}_\sigma)_k \\ \text{and } \langle (\xi)_k (\xi)_l \rangle_\sigma &= \langle (\xi)_k \rangle_\sigma \langle (\xi)_l \rangle_\sigma = \ell_\sigma^2 (\mathbf{B}_\sigma)_k (\mathbf{B}_\sigma)_l \end{aligned}$$

Hence we obtain the conditional second order moment, from Eqs. (3.25) and (3.22),

$$\langle h_i h_j \rangle_\sigma - \langle h_i \rangle_\sigma \langle h_j \rangle_\sigma = v_\sigma Q_{ij} + \ell_\sigma^2 R_{i\sigma} R_{j\sigma} - \ell_\sigma R_{i\sigma} \ell_\sigma R_{j\sigma} = v_\sigma Q_{ij} \quad (3.26)$$



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

$$\langle b_\tau b_\rho \rangle_\sigma - \langle b_\tau \rangle_\sigma \langle b_\rho \rangle_\sigma = v_\sigma T_{\tau\rho} + \ell_\sigma^2 T_{\tau\sigma} T_{\rho\sigma} - \ell_\sigma T_{\tau\sigma} \ell_\sigma T_{\rho\sigma} = v_\sigma T_{\tau\rho} \quad (3.27)$$

$$\langle h_i b_\tau \rangle_\sigma - \langle h_i \rangle_\sigma \langle b_\tau \rangle_\sigma = v_\sigma R_{i\tau} + \ell_\sigma^2 R_{i\sigma} T_{\tau\sigma} - \ell_\sigma R_{i\sigma} \ell_\sigma T_{\tau\sigma} = v_\sigma R_{i\tau} \quad (3.28)$$

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

$$C_\sigma = v_\sigma \begin{pmatrix} Q_{11} & \cdots & Q_{1K} & R_{11} & \cdots & R_{1M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ Q_{1K} & \cdots & Q_{KK} & R_{K1} & \cdots & R_{KM} \\ R_{11} & \cdots & R_{K1} & T_{11} & \cdots & T_{1M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ R_{1M} & \cdots & R_{KM} & T_{M1} & \cdots & T_{MM} \end{pmatrix} \quad (3.29)$$

### 3.B Form of the Differential Equations

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

$$\langle f_S \rangle, \quad \langle \mathbf{x}_n f_S \rangle \quad \text{and} \quad \langle f_S f_T \rangle \quad (3.30)$$

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

#### LVQ 2.1

The general modulation function for LVQ 2.1 is described in Equation (3.4) as

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

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

$$\begin{aligned} \Theta_{ST}^\delta &= \Theta(d_T - d_S - \delta) \\ &= \Theta(-2\mathbf{w}_T \cdot \xi^\mu + \mathbf{w}_T^2 + 2\mathbf{w}_S \cdot \xi^\mu - \mathbf{w}_S^2 - \delta) \\ &= \Theta(-2h_T^\mu + 2h_S^\mu + Q_{TT} - Q_{SS} - \delta) \\ &= \Theta(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}^\delta), \end{aligned} \quad (3.31)$$

with  $\alpha_{ST} = (0, \dots, \underbrace{+2}_{\text{at } S}, \dots, \underbrace{-2}_{\text{at } T}, \dots, 0)$  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

$$f_S = \chi(c_S, y_\sigma^\mu) (\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}). \quad (3.32)$$

And the required averages over the joint density (3.30) are calculated as

$$\begin{aligned} \langle f_S \rangle &= \langle \chi(c_S, y_\sigma^\mu) (\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}) \rangle = \sum_{\sigma=1}^M p_\sigma \chi(c_S, y_\sigma) \langle \Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta} \rangle_\sigma \\ \langle \mathbf{x}_n f_S \rangle &= \sum_{\sigma=1}^M p_\sigma \chi(c_S, y_\sigma) \langle \mathbf{x}_n (\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta}) \rangle_\sigma \\ \langle f_S f_S \rangle &= \langle \chi(c_S, y_\sigma^\mu)^2 (\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta})^2 \rangle = \sum_{\sigma=1}^M p_\sigma \langle \Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta} \rangle_\sigma^2 \\ \langle f_S f_T \rangle &= \langle \chi(c_S, y_\sigma^\mu) \chi(c_T, y_\sigma^\mu) (\Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta})^2 \rangle = - \sum_{\sigma=1}^M p_\sigma \langle \Theta_{ST}^{-\delta} - \Theta_{ST}^{+\delta} \rangle_\sigma \end{aligned} \quad (3.33)$$

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 3.C.

### LFM-W

The general modulation function for LFM-W is described in Equation (3.5) as

$$f_S = \begin{cases} \sum_{K: c_K \neq y_\sigma} (\Theta_{KS} - \Theta_{KS}^\delta) \psi(S, K) & \text{if } c_S = y_\sigma^\mu \\ \sum_{J: c_J = y_\sigma} (\Theta_{SJ} - \Theta_{SJ}^\delta) \psi(J, S) & \text{else.} \end{cases} \quad (3.34)$$

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  $\psi(\cdot) = 1$  and the averages are

$$\begin{aligned} \langle f_S \rangle &= \sum_{\sigma: y_\sigma = c_S}^M p_\sigma \langle \Theta_{TS} - \Theta_{TS}^\delta \rangle_\sigma + \sum_{\sigma: y_\sigma \neq c_S}^M p_\sigma \langle \Theta_{ST} - \Theta_{ST}^\delta \rangle_\sigma \\ \langle \mathbf{x}_n f_S \rangle &= \sum_{\sigma: y_\sigma = c_S}^M p_\sigma \langle \mathbf{x}_n (\Theta_{TS} - \Theta_{TS}^\delta) \rangle_\sigma + \sum_{\sigma: y_\sigma \neq c_S}^M p_\sigma \langle \mathbf{x}_n (\Theta_{ST} - \Theta_{ST}^\delta) \rangle_\sigma \end{aligned} \quad (3.35)$$

### GLVQ

The general modulation function for GLVQ is described in Equation (3.9) as

$$f_S = \begin{cases} \sum_{K:c_K \neq y_\sigma} \left( \frac{2}{v_G} \phi \left( \frac{d_S - d_K}{v_G} \right) \right) \psi(S, K) & \text{if } c_S = y_\sigma^\mu \\ - \sum_{J:c_J = y_\sigma} \left( \frac{2}{v_G} \phi \left( \frac{d_J - d_S}{v_G} \right) \right) \psi(J, S) & \text{else.} \end{cases} \quad (3.36)$$

For two prototypes,

$$\begin{aligned} \langle f_S \rangle &= \sum_{\sigma:y_\sigma=c_S}^M p_\sigma \frac{2}{v_G} \langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma - \sum_{\sigma:y_\sigma \neq c_S}^M p_\sigma \frac{2}{v_G} \langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma \\ \langle \mathbf{x}_n f_S \rangle &= \sum_{\sigma:y_\sigma=c_S}^M p_\sigma \frac{2}{v_G} \langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma - \sum_{\sigma:y_\sigma \neq c_S}^M p_\sigma \frac{2}{v_G} \langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma \end{aligned} \quad (3.37)$$

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

The quantities  $\langle \phi(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST}) \rangle_\sigma$  are found in Equation (3.48) in Appendix 3.C.

### RSLVQ

With one prototype representing each class, (2.12) become

$$\begin{aligned} P_\sigma(S|\xi^\mu) &= \frac{\exp(-(\xi^\mu - \mathbf{w}_S^\mu)^2/2v_{\text{soft}})}{\exp(-(\xi^\mu - \mathbf{w}_S^\mu)^2/2v_{\text{soft}})} = 1 \\ P(S|\xi^\mu) &= \frac{\exp(-(\xi^\mu - \mathbf{w}_S^\mu)^2/2v_{\text{soft}})}{\sum_{T=1}^K \exp(-(\xi^\mu - \mathbf{w}_T^\mu)^2/2v_{\text{soft}})} \\ &= \frac{1}{1 + \sum_{T \neq S}^K \exp\left(\frac{1}{2v_{\text{soft}}} (-2\xi^\mu \mathbf{w}_S^\mu + (\mathbf{w}_S^\mu)^2 + 2\xi^\mu \mathbf{w}_T^\mu - (\mathbf{w}_T^\mu)^2)\right)} \\ &= \frac{1}{1 + \sum_{T \neq S}^K \exp\left(\frac{1}{2v_{\text{soft}}} (-2h_S + Q_{SS} + 2h_T - Q_{TT})\right)} \\ &= \frac{1}{1 + \sum_{T \neq S}^K \exp(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST})} \end{aligned} \quad (3.38)$$

where we defined

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

Therefore the RSLVQ modulation function becomes

$$f_S = \frac{1}{v_{\text{soft}}} (\delta(c_S, y_\sigma^\mu) - \Omega_S) \quad (3.39)$$

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

$$\Omega_S = \frac{1}{1 + \sum_{T \neq S}^K \exp(\alpha_{ST} \cdot \mathbf{x} - \beta_{ST})} \quad (3.40)$$

We obtain the averages

$$\begin{aligned} \langle f_S \rangle &= \frac{1}{v_{\text{soft}}} \langle \delta(c_S, y_\sigma) - \Omega_S \rangle = \frac{1}{v_{\text{soft}}} \left( \sum_{\sigma: y_\sigma = c_S} p_\sigma - \sum_{\sigma} p_\sigma \langle \Omega_S \rangle_\sigma \right) \\ \langle \mathbf{x}_n f_S \rangle &= \begin{cases} \frac{1}{v_{\text{soft}}} \left( \sum_{\sigma: y_\sigma = c_S} p_\sigma \langle h_n \rangle_\sigma - \sum_{\sigma} p_\sigma \langle \mathbf{x}_n \Omega_S \rangle_\sigma \right) & \text{if } n \leq K \\ \frac{1}{v_{\text{soft}}} \left( \sum_{\sigma: y_\sigma = c_S} p_\sigma \langle b_{n-K} \rangle_\sigma - \sum_{\sigma} p_\sigma \langle \mathbf{x}_n \Omega_S \rangle_\sigma \right) & \text{if } n > K \end{cases} \\ &= \begin{cases} \frac{1}{v_{\text{soft}}} \left( \sum_{\sigma: y_\sigma = c_S} p_\sigma \ell_\sigma R_{n\sigma} - \sum_{\sigma} p_\sigma \langle \mathbf{x}_n \Omega_S \rangle_\sigma \right) & \text{if } n \leq K \\ \frac{1}{v_{\text{soft}}} \left( \sum_{\sigma: y_\sigma = c_S} p_\sigma \ell_\sigma T_{(n-K)\sigma} - \sum_{\sigma} p_\sigma \langle \mathbf{x}_n \Omega_S \rangle_\sigma \right) & \text{if } n > K \end{cases} \quad (3.41) \end{aligned}$$

The required quantities  $\langle \Omega_S \rangle_\sigma$  and  $\langle \mathbf{x}_n \Omega_S \rangle_\sigma$  are supplied in Appendix 3.C.

### 3.C Gaussian Averages

#### 3.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.

$$\begin{aligned}
\langle f_{ab} \rangle_\sigma &= \frac{1}{(2\pi)^{D/2}(\det(C_\sigma))^{1/2}} \int_{\mathbb{R}^D} f(\alpha_{ab} \cdot \mathbf{x} - \beta_{ab}) \\
&\quad \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T C_\sigma^{-1}(\mathbf{x} - \mu)\right) d\mathbf{x} \\
&= \frac{1}{(2\pi)^{D/2}(\det(C_\sigma))^{1/2}} \int_{\mathbb{R}^D} f(\alpha_{ab} \cdot \mathbf{x}' + \alpha_{ab} \cdot \mu - \beta_{ab}) \\
&\quad \exp\left(-\frac{1}{2}(\mathbf{x}')^T C_\sigma^{-1}(\mathbf{x}')\right) d\mathbf{x}' \\
&= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} f\left(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma}\right) \exp\left(-\frac{1}{2} \mathbf{y}^2\right) d\mathbf{y} \quad (3.42)
\end{aligned}$$

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

$$\begin{aligned}
\langle f_{ab} \rangle_\sigma &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f\left(\|\alpha_{ab} C_\sigma^{-1/2}\| \tilde{y} + \tilde{\beta}_{ab,\sigma}\right) \exp\left(-\frac{1}{2} \tilde{y}^2\right) d\tilde{y} \\
&= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f\left(\tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma}\right) \exp\left(-\frac{1}{2} \tilde{y}^2\right) d\tilde{y} \quad (3.43)
\end{aligned}$$

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

$$\begin{aligned}
\langle \mathbf{x}_n f_{ab} \rangle_\sigma &= \frac{1}{(2\pi)^{D/2}(\det(C_\sigma))^{1/2}} \int_{\mathbb{R}^D} \mathbf{x}_n f(\alpha_{ab} \cdot \mathbf{x} - \beta_{ab}) \\
&\quad \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T C_\sigma^{-1}(\mathbf{x} - \mu)\right) d\mathbf{x} \\
&= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \left(C_\sigma^{1/2} \mathbf{y} + \mu\right)_n f\left(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma}\right) \exp\left(-\frac{1}{2} \mathbf{y}^2\right) d\mathbf{y} \\
&= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \left(C_\sigma^{1/2} \mathbf{y}\right)_n f\left(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma}\right) \exp\left(-\frac{1}{2} \mathbf{y}^2\right) d\mathbf{y} \\
&\quad + (\mu)_n \langle f_{ab} \rangle_\sigma \quad (3.44)
\end{aligned}$$

#### LVQ 2.1, LFM-W

The following quantities are required for two prototype LVQ 2.1 and LFM-W:

$$\langle \Theta_{ab}^\delta - \Theta_{ab}^\gamma \rangle_\sigma = \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}^\delta}{\tilde{\alpha}_{ab,\sigma}}\right) - \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}^\gamma}{\tilde{\alpha}_{ab,\sigma}}\right) \quad (3.45)$$

$$\begin{aligned}
\langle \mathbf{x}_n (\Theta_{ab}^\delta - \Theta_{ab}^\gamma) \rangle_\sigma &= \frac{(C_\sigma \alpha_{ab})_n}{\sqrt{2\pi} \tilde{\alpha}_{ab,\sigma}} \left\{ \exp \left[ -\frac{1}{2} \left( \frac{\tilde{\beta}_{ab,\sigma}^\delta}{\tilde{\alpha}_{ab,\sigma}} \right)^2 \right] - \exp \left[ -\frac{1}{2} \left( \frac{\tilde{\beta}_{ab,\sigma}^\gamma}{\tilde{\alpha}_{ab,\sigma}} \right)^2 \right] \right\} \\
&\quad + (\mu_\sigma)_n \left[ \Phi \left( \frac{\tilde{\beta}_{ab,\sigma}^\delta}{\tilde{\alpha}_{ab,\sigma}} \right) - \Phi \left( \frac{\tilde{\beta}_{ab,\sigma}^\gamma}{\tilde{\alpha}_{ab,\sigma}} \right) \right]
\end{aligned} \tag{3.46}$$

### GLVQ

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

$$\begin{aligned}
\langle \phi_{ab} \rangle_\sigma &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \phi \left( \tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma} \right) \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y} \\
&= \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} \tilde{\beta}_{ab,\sigma}^2 \right) \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} (\tilde{\alpha}_{ab,\sigma}^2 + 1) \tilde{y}^2 - \tilde{\alpha}_{ab,\sigma} \tilde{\beta}_{ab,\sigma} \tilde{y} \right) d\tilde{y}
\end{aligned} \tag{3.47}$$

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

$$\begin{aligned}
\langle \phi_{ab} \rangle_\sigma &= \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} \tilde{\beta}_{ab,\sigma}^2 \right) \frac{1}{\sqrt{\tilde{\alpha}_{ab,\sigma}^2 + 1}} \exp \left( \frac{\tilde{\alpha}_{ab,\sigma}^2 \tilde{\beta}_{ab,\sigma}^2}{2(\tilde{\alpha}_{ab,\sigma}^2 + 1)} \right) \\
&= \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\tilde{\alpha}_{ab,\sigma}^2 + 1}} \exp \left( -\frac{1}{2} \tilde{\beta}_{ab,\sigma}^2 \left( 1 - \frac{\tilde{\alpha}_{ab,\sigma}^2}{(\tilde{\alpha}_{ab,\sigma}^2 + 1)} \right) \right) \\
&= \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\tilde{\alpha}_{ab,\sigma}^2 + 1}} \exp \left( -\frac{1}{2} \frac{\tilde{\beta}_{ab,\sigma}^2}{\tilde{\alpha}_{ab,\sigma}^2 + 1} \right)
\end{aligned} \tag{3.48}$$

### RSLVQ

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

$$\langle \Omega_{ab} \rangle_\sigma = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \frac{1}{1 + \exp(\tilde{\alpha}_{ab,\sigma} \tilde{y} + \tilde{\beta}_{ab,\sigma})} \exp \left( -\frac{1}{2} \tilde{y}^2 \right) d\tilde{y} \tag{3.49}$$

This one-dim. integration has to be solved numerically.

$$\begin{aligned}
\langle \mathbf{x}_n \Omega_{ab} \rangle_\sigma &= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \frac{(C_\sigma^{1/2} \mathbf{y})_n}{1 + \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab, \sigma})} \exp\left(-\frac{1}{2} \mathbf{y}^2\right) d\mathbf{y} \\
&\quad + (\mu)_n \langle \Omega_{ab} \rangle_\sigma \\
&= \frac{1}{(2\pi)^{D/2}} \sum_{j=1}^D I_j + (\mu)_n \langle \Omega_{ab} \rangle_\sigma
\end{aligned} \tag{3.50}$$

where

$$I_j = \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}} \frac{(C_\sigma^{1/2})_{nj}(\mathbf{y})_n}{1 + \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab, \sigma})} \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) d(\mathbf{y})_j \tag{3.51}$$

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

$$\begin{aligned}
u &= \frac{1}{1 + \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab, \sigma})} \\
v &= (C_\sigma^{1/2})_{nj} \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) \\
du &= -\frac{\exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab, \sigma})}{\left(1 + \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab, \sigma})\right)^2} \frac{\partial}{\partial(\mathbf{y})_j} \left(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y}\right) d(\mathbf{y})_j \\
dv &= -(C_\sigma^{1/2})_{nj}(\mathbf{y})_j \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) d(\mathbf{y})_j,
\end{aligned} \tag{3.52}$$

we obtain

$$\begin{aligned}
I_j &= \left[ \frac{1}{1 + \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab, \sigma})} (C_\sigma^{1/2})_{nj} \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) \right]_{-\infty}^{\infty} \\
&\quad - \int_{\mathbb{R}} \frac{(C_\sigma^{1/2})_{nj} \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab, \sigma})}{\left(1 + \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab, \sigma})\right)^2} \frac{\partial}{\partial(\mathbf{y})_j} \left(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y}\right) \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) d(\mathbf{y})_j
\end{aligned} \tag{3.53}$$

$$\begin{aligned}
\langle \mathbf{x}_n \Omega_{ab} \rangle_\sigma &= -\frac{1}{(2\pi)^{D/2}} \sum_{j=1}^D \int_{\mathbb{R}} \frac{(C_\sigma^{1/2})_{nj} \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma})}{\left(1 + \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma})\right)^2} \\
&\quad \frac{\partial}{\partial (\mathbf{y})_j} \left( \alpha_{ab} C_\sigma^{-1/2} \mathbf{y} \right) \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) d(\mathbf{y})_j + (\mu)_n \langle \Omega_{ab} \rangle_\sigma \\
&= -\frac{1}{(2\pi)^{D/2}} (C_k \alpha_{ab})_n \int_{\mathbb{R}} \frac{\exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma})}{\left(1 + \exp(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma})\right)^2} \\
&\quad \exp\left(-\frac{1}{2}(\mathbf{y}_j)^2\right) d(\mathbf{y})_j \tag{3.54}
\end{aligned}$$

After applying rotation,

$$\begin{aligned}
\langle \mathbf{x}_n \Omega_{ab} \rangle_\sigma &= -\frac{(C_k \alpha_{ab})_n}{\sqrt{2\pi}} \int_{\mathbb{R}} \frac{\exp\|\alpha_{ab} C_\sigma^{-1/2} \|\tilde{\mathbf{y}} + \tilde{\beta}_{ab,\sigma}\|}{\left(1 + \exp\left(\|\alpha_{ab} C_\sigma^{-1/2} \|\tilde{\mathbf{y}} + \tilde{\beta}_{ab,\sigma}\|\right)\right)^2} \\
&\quad \times \exp\left(-\frac{1}{2}\tilde{\mathbf{y}}^2\right) d\tilde{\mathbf{y}} + (\mu_k)_n \langle \Omega_{ab} \rangle_\sigma \\
&= -\frac{(C_k \alpha_{ab})_n}{\sqrt{2\pi}} \int_{\mathbb{R}} \frac{\exp(\tilde{\alpha}_{ab,\sigma} \tilde{\mathbf{y}} + \tilde{\beta}_{ab,\sigma})}{\left(1 + \exp(\tilde{\alpha}_{ab,\sigma} \tilde{\mathbf{y}} + \tilde{\beta}_{ab,\sigma})\right)^2} \exp\left(-\frac{1}{2}\tilde{\mathbf{y}}^2\right) d\tilde{\mathbf{y}} \\
&\quad + (\mu_k)_n \langle \Omega_{ab} \rangle_\sigma \tag{3.55}
\end{aligned}$$

which is also solved numerically.

### 3.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.

$$\begin{aligned}
\langle f_{ab} f_{cd} \rangle_\sigma &= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} f(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma}) f(\alpha_{cd} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{cd}) \\
&\quad \times \exp\left(-\frac{1}{2}\mathbf{y}^2\right) d\mathbf{y} \tag{3.56}
\end{aligned}$$

Next we calculate the quantity

$$\begin{aligned}
\langle \mathbf{x}_n f_{ab} f_{cd} \rangle_\sigma &= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} (C_\sigma^{1/2} \mathbf{y})_n f(\alpha_{ab} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{ab,\sigma}) f(\alpha_{cd} C_\sigma^{-1/2} \mathbf{y} + \tilde{\beta}_{cd}) \\
&\quad \times \exp\left(-\frac{1}{2}\mathbf{y}^2\right) d\mathbf{y} + (\mu)_n \langle f_{ab} f_{cd} \rangle_\sigma \tag{3.57}
\end{aligned}$$



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:

$$\langle \Theta_{ab} \Theta_{cd} \rangle_\sigma = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}}^{\infty} \exp\left(-\frac{1}{2}y_1'^2\right) \Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma} + (\alpha_{cd}C_\sigma\alpha_{ab})y_1'}{\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2 - (\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) dy_1' \quad (3.58)$$

$$\begin{aligned} \langle (\mathbf{x})_n \Theta_{ab} \Theta_{cd} \rangle_\sigma &= \frac{(C_\sigma\alpha_{ab})_n}{\sqrt{(2\pi)\tilde{\alpha}_{ab,\sigma}}} \exp\left(-\frac{1}{2}\frac{\tilde{\beta}_{ab,\sigma}^2}{\tilde{\alpha}_{ab,\sigma}^2}\right) \Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma}^2 - \tilde{\beta}_{ab,\sigma}(\alpha_{cd}C_\sigma\alpha_{ab})}{\tilde{\alpha}_{ab,\sigma}\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2 - (\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) \\ &+ \frac{(C_\sigma\alpha_{cd})_n}{\sqrt{(2\pi)\tilde{\alpha}_{cd,\sigma}}} \exp\left(-\frac{1}{2}\frac{\tilde{\beta}_{cd,\sigma}^2}{\tilde{\alpha}_{cd,\sigma}^2}\right) \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}\tilde{\alpha}_{cd,\sigma}^2 - \tilde{\beta}_{cd,\sigma}(\alpha_{ab}C_\sigma\alpha_{cd})}{\tilde{\alpha}_{cd,\sigma}\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2 - (\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) \\ &+ (\mu_\sigma)_n \langle \Theta_{ab} \Theta_{cd} \rangle_\sigma. \end{aligned} \quad (3.59)$$

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

$$\langle (\Theta_{ab}^\delta - \Theta_{ab}^\gamma) \Theta_{cd} \rangle_\sigma = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\tilde{\beta}_{ab,\sigma}^\delta}{\tilde{\alpha}_{ab,\sigma}^\delta}}^{\frac{\tilde{\beta}_{ab,\sigma}^\gamma}{\tilde{\alpha}_{ab,\sigma}^\gamma}} \exp\left(-\frac{1}{2}y_1'^2\right) \Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma} + (\alpha_{cd}C_\sigma\alpha_{ab})y_1'}{\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2 - (\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) dy_1'$$

$$\begin{aligned} \langle (\mathbf{x})_n (\Theta_{ab}^\delta - \Theta_{ab}^\gamma) \Theta_{cd} \rangle_\sigma &= \frac{(C_\sigma\alpha_{ab})_n}{\sqrt{(2\pi)\tilde{\alpha}_{ab,\sigma}}} \exp\left(-\frac{1}{2}\frac{(\tilde{\beta}_{ab,\sigma}^\delta)^2}{\tilde{\alpha}_{ab,\sigma}^2}\right) \Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma}^2 - \tilde{\beta}_{ab,\sigma}^\delta(\alpha_{cd}C_\sigma\alpha_{ab})}{\tilde{\alpha}_{ab,\sigma}\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2 - (\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) \\ &+ \frac{(C_\sigma\alpha_{cd})_n}{\sqrt{(2\pi)\tilde{\alpha}_{cd,\sigma}}} \exp\left(-\frac{1}{2}\frac{\tilde{\beta}_{cd,\sigma}^2}{\tilde{\alpha}_{cd,\sigma}^2}\right) \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}^\delta\tilde{\alpha}_{cd,\sigma}^2 - \tilde{\beta}_{cd,\sigma}(\alpha_{ab}C_\sigma\alpha_{cd})}{\tilde{\alpha}_{cd,\sigma}\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2 - (\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) \\ &- \frac{(C_\sigma\alpha_{ab})_n}{\sqrt{(2\pi)\tilde{\alpha}_{ab,\sigma}}} \exp\left(-\frac{1}{2}\frac{(\tilde{\beta}_{ab,\sigma}^\gamma)^2}{\tilde{\alpha}_{ab,\sigma}^2}\right) \Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma}^2 - \tilde{\beta}_{ab,\sigma}^\gamma(\alpha_{cd}C_\sigma\alpha_{ab})}{\tilde{\alpha}_{ab,\sigma}\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2 - (\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) \\ &- \frac{(C_\sigma\alpha_{cd})_n}{\sqrt{(2\pi)\tilde{\alpha}_{cd,\sigma}}} \exp\left(-\frac{1}{2}\frac{\tilde{\beta}_{cd,\sigma}^2}{\tilde{\alpha}_{cd,\sigma}^2}\right) \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}^\gamma\tilde{\alpha}_{cd,\sigma}^2 - \tilde{\beta}_{cd,\sigma}(\alpha_{ab}C_\sigma\alpha_{cd})}{\tilde{\alpha}_{cd,\sigma}\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2 - (\alpha_{cd}C_\sigma\alpha_{ab})^2}}\right) \\ &+ (\mu_\sigma)_n \langle (\Theta_{ab}^\delta - \Theta_{ab}^\gamma) \Theta_{cd} \rangle_\sigma. \end{aligned} \quad (3.60)$$

For LVQ 2.1, the following average is required:

$$\langle (\Theta_{ab}^\delta - \Theta_{ab}^\gamma) \Theta_{ac} \Theta_{bc} \rangle_\sigma = \frac{1}{\sqrt{2\pi}} \int_{y_{1,\min}}^{y_{1,\max}} \exp\left(-\frac{1}{2}y_1'^2\right) \Phi(-y_2^*) dy_1 \quad (3.61)$$

$$\begin{aligned} \text{with } y_{1,\min} &= -\frac{\tilde{\beta}_{ab,\sigma}^\delta}{\tilde{\alpha}_{ab,\sigma}}, \quad y_{1,\max} = -\frac{\tilde{\beta}_{ab,\sigma}^\gamma}{\tilde{\alpha}_{ab,\sigma}} \\ y_2^* &= \min\left(\frac{-\tilde{\beta}_{ac} - (\alpha_{ac}C_\sigma^{1/2}e_1)y_1}{\alpha_{ac}C_\sigma^{1/2}e_2}, \frac{-\tilde{\beta}_{bc} - (\alpha_{bc}C_\sigma^{1/2}e_1)y_1}{\alpha_{bc}C_\sigma^{1/2}e_2}\right) \end{aligned} \quad (3.62)$$

$$\langle \mathbf{x}_n(\Theta_{ab}^\delta - \Theta_{ab}^\gamma)\Theta_{ac}\Theta_{bc} \rangle_\sigma = I_{ab} + I_{ac} + I_{bc} + (\mu)_n \langle \mathbf{x}_n(\Theta_{ab}^\delta - \Theta_{ab}^\gamma)\Theta_{ac}\Theta_{bc} \rangle_\sigma \quad (3.63)$$

where

$$\begin{aligned} I_{ab} &= \frac{(C_\sigma\alpha_{ab})_n}{\sqrt{2\pi}\tilde{\alpha}_{ab,\sigma}} \left[ \exp\left(-\frac{1}{2}\left(-\frac{\tilde{\beta}_{ab,\sigma}^\delta}{\tilde{\alpha}_{ab,\sigma}}\right)^2\right) (\Phi(-y_{2,\min}^\delta) - \Phi(-y_{2,\max}^\delta)) \right. \\ &\quad \left. - \exp\left(-\frac{1}{2}\left(-\frac{\tilde{\beta}_{ab,\sigma}^\gamma}{\tilde{\alpha}_{ab,\sigma}}\right)^2\right) (\Phi(-y_{2,\min}^\gamma) - \Phi(-y_{2,\max}^\gamma)) \right] \end{aligned} \quad (3.64)$$

$$\begin{aligned} I_{ac} &= \frac{(C_\sigma\alpha_{ac})_n}{\sqrt{2\pi}\tilde{\alpha}_{ac,\sigma}} \left[ \exp\left(-\frac{1}{2}(z)^2\right) (\Phi(-y_{2,\min}) - \Phi(-y_{2,\max})) \right. \\ &\quad \left. - \exp\left(-\frac{1}{2}(z)^2\right) (\Phi(-y_{2,\min}) - \Phi(-y_{2,\max})) \right] \end{aligned} \quad (3.65)$$

$$\begin{aligned} \text{with } y_{1,\min} &= -\frac{\tilde{\beta}_{ab,\sigma}^\delta}{\tilde{\alpha}_{ab,\sigma}}, \quad y_{1,\max} = -\frac{\tilde{\beta}_{ab,\sigma}^\gamma}{\tilde{\alpha}_{ab,\sigma}} \\ y_2^* &= \min\left(\frac{-\tilde{\beta}_{ac} - (\alpha_{ac}C_\sigma^{1/2}e_1)y_1}{\alpha_{ac}C_\sigma^{1/2}e_2}, \frac{-\tilde{\beta}_{bc} - (\alpha_{bc}C_\sigma^{1/2}e_1)y_1}{\alpha_{bc}C_\sigma^{1/2}e_2}\right) \end{aligned} \quad (3.66)$$

### 3.D Generalization error

#### Two prototypes

We compute the generalization error from Equation (3.16) as follows. For two prototypes  $w_+$  and  $w_-$ , we calculate  $\epsilon_g = \sum p_\sigma \epsilon_{g,\sigma}$  with

$$\epsilon_{g,\sigma} = \langle \Theta_{-\sigma\sigma} \rangle_+ = \Phi\left(\frac{\tilde{\beta}_{-\sigma\sigma,\sigma}}{\tilde{\alpha}_{-\sigma\sigma,\sigma}}\right) \quad (3.67)$$

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

$$\epsilon_{g,\sigma} = \Phi \left( \frac{Q_{\sigma\sigma} - Q_{-\sigma,-\sigma} - 2\ell_{\sigma}(R_{\sigma\sigma} - R_{-\sigma,\sigma})}{2\sqrt{v_{\sigma}} \sqrt{Q_{\sigma\sigma} - 2Q_{\sigma,-\sigma} + Q_{-\sigma,-\sigma}}} \right) \quad (3.68)$$

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:

$$\frac{d\epsilon_{g\sigma}}{d\mathbf{O}} = \frac{1}{\sqrt{2\pi}2\sqrt{v_{\sigma}}} \exp \left( -\frac{1}{2} \left[ \frac{Z_{\sigma}}{2\sqrt{v_{\sigma}}\Delta_q} \right]^2 \right) \frac{d}{d\mathbf{O}} \frac{Z_{\sigma}}{\Delta_q} \quad (3.69)$$

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

$$\frac{d}{d\mathbf{O}} \frac{Z_{+}}{\Delta_q} = \begin{bmatrix} -2\ell/\Delta_q \\ 0 \\ +2\ell/\Delta_q \\ 0 \\ 1/\Delta_q - Z_{+}/(2\Delta_q^3) \\ Z_{+}/\Delta_q^3 \\ -1/\Delta_q - Z_{+}/(2\Delta_q^3) \end{bmatrix}, \quad \frac{d}{d\mathbf{O}} \frac{Z_{-}}{\Delta_q} = \begin{bmatrix} 0 \\ +2\ell/\Delta_q \\ 0 \\ -2\ell/\Delta_q \\ -1/\Delta_q - Z_{-}/(2\Delta_q^3) \\ Z_{-}/\Delta_q^3 \\ 1/\Delta_q - Z_{-}/(2\Delta_q^3) \end{bmatrix} \quad (3.70)$$

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

$$\frac{d\epsilon_{g\sigma}}{d\mathbf{O}} = \sum_{\sigma} \frac{d\epsilon_{g\sigma}}{d\mathbf{O}} = \frac{1}{2\sqrt{2\pi}\sqrt{v}} \exp \left( -\frac{1}{2} \left[ \frac{Z}{2\sqrt{v}\Delta_q} \right]^2 \right) \begin{bmatrix} -\ell/\Delta_q \\ +\ell/\Delta_q \\ +\ell/\Delta_q \\ -\ell/\Delta_q \\ -Z/(2\Delta_q^3) \\ Z/\Delta_q^3 \\ -Z/(2\Delta_q^3) \end{bmatrix}. \quad (3.71)$$

### 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

$$\epsilon_{g,\sigma} = \sum_{S:c_S \neq y_{\sigma}}^K \langle \Theta_{ST} \Theta_{SU} \rangle_{\sigma}, \quad (3.72)$$

where the averages are written in Equation (3.58).