Learning Vector Quantization

Witoelar, Aree Widya; Biehl, Michael; Hammer, Barbara

Published in:
Proceedings

Publication date:
2007

Learning Vector Quantization: generalization ability and dynamics of competing prototypes

Aree Witoelar[1], Michael Biehl[1], and Barbara Hammer[2]

[1] University of Groningen, Mathematics and Computing Science
P.O. Box 800, NL-9700 AV, Groningen, The Netherlands
a.w.witoelar@rug.nl,m.biehl@rug.nl
[2] Clausthal University of Technology, Institute of Computer Science
D-98678 Clausthal-Zellerfeld, Germany
hammer@in.tu-clausthal.de

Abstract. Learning Vector Quantization (LVQ) are popular multi-class classification algorithms. Prototypes in an LVQ system represent the typical features of classes in the data. Frequently multiple prototypes are employed for a class to improve the representation of variations within the class and the generalization ability. In this paper, we investigate the dynamics of LVQ in an exact mathematical way, aiming at understanding the influence of the number of prototypes and their assignment to classes. The theory of on-line learning allows a mathematical description of the learning dynamics in model situations. We demonstrate using a system of three prototypes the different behaviors of LVQ systems of multiple prototype and single prototype class representation.

Keywords. online learning, learning vector quantization

1 Introduction

Learning Vector Quantization (LVQ) is a family of powerful and intuitive classification algorithms. LVQ is used in many applications, including medical data or gene expressions, and handwriting recognition [1]. Prototypes in LVQ algorithms represent typical features within a data set using the same feature space instead of a black-box approach as in many other classification algorithms, e.g. feedforward neural networks or support vector machines. This approach makes them attractive for researchers outside the field of machine learning. Other advantages of LVQ algorithms are (1) they are easy to be implemented for multi-class classification problems and (2) the algorithm complexity can be adjusted during training as needed.

One widely used method in increasing this complexity is by employing multiple prototypes in a class in order to improve representation of variations within the class and generalization ability. However, the effectiveness of this strategy and the influence of multiple prototypes on the learning dynamics and performance of LVQ has not been studied thoroughly.

Dagstuhl Seminar Proceedings 07131
Similarity-based Clustering and its Application to Medicine and Biology
http://drops.dagstuhl.de/opus/volltexte/2007/1131

In general, LVQ algorithms such as Kohonen's original LVQ1 are based on heuristics and many variants are developed without an associated cost function related to generalization ability. There is a lack of theoretical understanding of the learning behavior such as convergence, stability, etc. In this paper we present a mathematical framework to analyse typical LVQ learning behavior on model data. While the model data considered here is certainly simplifying compared to practical situations, it provides an insight on idealized situations and a base for its extension to real-life situations.

2 Algorithms

Let the input data at time step $\mu = 1, 2, \ldots$ be given as $\{\xi^\mu, \sigma^\mu\}$, $\xi^\mu \in I\!R^N$ where N is the potentially high dimension of the data and σ^μ is the class of the data. Here we investigate learning schemes with two classes $\sigma^\mu = \pm 1$ (or $\pm$). An LVQ system aims to represent the data using a set of prototypes $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_S\} \in I\!R^N$ and their class labels $C = \{c_1, \ldots, c_S\} = \pm 1$. To train the system, a single example $\{\xi^\mu, \sigma^\mu\}$ is generated at each learning step according to an input density $P(\xi)$ and presented sequentially to the system. One or more prototypes $\mathbf{w}_i$ are then updated on-line as

$$\mathbf{w}_i^\mu = \mathbf{w}_i^{\mu-1} + \frac{\eta}{N} f_i(d_1^\mu, \ldots, d_S^\mu, \sigma^\mu)(\xi^\mu - \mathbf{w}_i^{\mu-1}) \tag{1}$$

where η is the learning rate and $d_i^\mu = (\xi^\mu - \mathbf{w}_i^{\mu-1})^2/2$ is the Euclidean distance measure. Prototypes are always moved towards or away from the example along the vector $(\xi^\mu - \mathbf{w}_i^{\mu-1})$. The direction and strength of update is specified by the modulation function $f_i(.)$. Here we present two basic LVQ algorithms:

1. **LVQ1**: The original formulation of LVQ by Kohonen [8,9] is an intuitive learning scheme that compromises between data representation and finding the decision boundary between classes. The closest prototype to the example is determined. This so-called *winner* is then moved towards the example if it is correct, ie. the winner class label matches the class of the example, or pushed away otherwise. The corresponding modulation function is $f_i(.) = c_i\sigma^\mu$ if w_i is the winner; 0 else.
2. **LVQ+/-**: LVQ+/- aims for a more efficient separation of prototypes with different classes and therefore better generalization ability. This scheme is a simplified version of LVQ2.1 proposed by Kohonen [7], omitting the restriction of selecting only examples close to the current decision boundary by a so-called *window* scheme. The two closest prototypes, say w_J and w_K, are determined. If their class labels are different and one of them is correct, the correct prototype is moved towards the data while the incorrect prototype is pushed away. The modulation function is $f_i(.) = c_i\sigma^\mu$ if $i \in J, K$ and $c_J \neq c_K$; 0 else.

3 Model

We choose the model data as a mixture of two classes $\sigma = \{\pm 1\}$ with the probability density function $P(\xi) = \sum_{\sigma=\pm 1} p_\sigma P(\xi|\sigma)$ with

$$P(\xi|\sigma) = \frac{1}{(\sqrt{2\pi})^N} \exp\left(-\frac{1}{2} \frac{(\xi - \lambda \mathbf{B}_\sigma)^2}{\upsilon_\sigma} \right)$$

where p_σ are the prior probabilities and $p_+ + p_- = 1$. The distribution of each class is a spherical Gaussian cluster. The components of vectors ξ^μ are random numbers with mean vectors $\lambda \mathbf{B}_\sigma$ and variance υ_σ. The parameter λ controls the separation between the mean vectors. $\mathbf{B}_\sigma$ are orthonormal, i.e. $\mathbf{B}_i \cdot \mathbf{B}_j = \delta_{i,j}$ where δ is the Kronecker delta.

Note that data from different classes strongly overlap. They separate only on a two-dimensional space spanned by $\mathbf{B}_+$ and $\mathbf{B}_-$ and completely overlap on other subspaces. The goal is to identify this separation from the N-dimensional data.

4 Analysis

In this section we describe the methods to analyse the learning dynamics in LVQ algorithms. We give a brief description of the theoretical framework and refer to [3,11] for further details. Following the lines of the theory of on-line learning, e.g. [5], the system can be fully described in terms of a few so-called order parameters in the thermodynamic limit $N \to \infty$. A suitable set of characteristic quantities for the considered learning model is:

$$R_{i\sigma}^\mu = \mathbf{w}_i^\mu \cdot \mathbf{B}_\sigma \quad Q_{ij}^\mu = \mathbf{w}_i^\mu \cdot \mathbf{w}_j^\mu. \tag{2}$$

Note that $R_{i\sigma}$ are the projections of prototype vectors $\mathbf{w}_i^\mu$ on the center vectors $\mathbf{B}_\sigma$ and Q_{ij}^μ correspond to the self- and cross- overlaps of the prototype vectors. These quantities are called the *order parameters*.

From the generic update rule defined above, Eq. (1), we can derive the following recursions in terms of the order parameters:

$$\begin{aligned} \frac{R_{i\sigma}^\mu - R_{i\sigma}^{\mu-1}}{1/N} &= \eta f_i(.)\big(b_\sigma^\mu - R_{i\sigma}^{\mu-1}\big) \\ \frac{Q_{ij}^\mu - Q_{ij}^{\mu-1}}{1/N} &= \eta\big[f_j(.)\big(h_i^\mu - Q_{ij}^{\mu-1}\big) + f_i(.)\big(h_j^\mu - Q_{ij}^{\mu-1}\big)\big] + \\ &\qquad \eta^2 f_i(.) f_j(.)(\xi^\mu)^2/N + \mathcal{O}(1/N) \end{aligned} \tag{3}$$

where the input data vectors ξ^μ enters the system as their projections h_i^μ and b_i^μ, defined as

$$h_i^\mu = \mathbf{w}_i^{\mu-1} \cdot \xi^\mu \quad b_\sigma^\mu = \mathbf{B}_\sigma \cdot \xi^\mu. \tag{4}$$

In the limit $N \to \infty$, the $\mathcal{O}(1/N)$ term can be neglected and the order parameters *self average* [10] 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 $t \equiv \mu/N$ can be conceived as a continuous time variable. Accordingly, the dynamics can be described by a set of coupled ODE [3,6] after performing an average over the sequence of input data:

$$\begin{aligned}\frac{dR_{i\sigma}}{dt} &= \eta\big(\langle b_\sigma f_i(.)\rangle - \langle f_i(.)\rangle R_{i\sigma}\big)\\ \frac{dQ_{ij}}{dt} &= \eta\big(\langle h_i f_j(.)\rangle - \langle f_j(.)\rangle Q_{ij} + \langle h_j f_i(.)\rangle - \langle f_i(.)\rangle Q_{ij}\big) + \\ &\qquad \eta^2 \sum_\sigma p_\sigma \upsilon_\sigma \langle f_i(.) f_j(.)\rangle_\sigma \end{aligned} \tag{5}$$

where $\langle . \rangle$ and $\langle . \rangle_\sigma$ are the averages over the density $P(\xi)$ and $P(\xi|\sigma)$, respectively. Here we used the following relation to simplify the last term of Eq. (5):

$$\lim_{N\to\infty} \frac{\langle \xi^2 \rangle}{N} = \lim_{N\to\infty} \frac{1}{N} \sum_\sigma p_\sigma (\upsilon_\sigma N + \lambda^2) = \sum_\sigma p_\sigma \upsilon_\sigma .$$

Exploiting the limit $N \to \infty$ once more, the quantities h_i^μ, b_σ^μ become correlated Gaussian quantities by means of the Central Limit Theorem. Thus, the above averages reduce to Gaussian integrations in $S + 2$ dimensions. In the simplest case of 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. See [3,11] for details of the computations.

Given the averages for a specific modulation function $f(.)$ we obtain a closed set of ODE. Using initial conditions $\{R_{i\sigma}(0), Q_{ij}(0)\}$, we integrate this system for a given algorithm and get the evolution of order parameters in the course of training, $\{R_{i\sigma}(t), Q_{ij}(t)\}$. Also the generalization error ϵ_g is determined from $\{R_{i\sigma}(t), Q_{ij}(t)\}$ as follows:

$$\epsilon_g = \sum_{\sigma=\pm 1} p_\sigma \langle f_i(.)\rangle_{-\sigma} \tag{6}$$

where $f_i(.) = 1$ if w_i is the winner; 0 else. We thus obtain the learning curve $\epsilon_g(t)$ 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 $N = 200$, as demonstrated in [2,6].

5 Results

The dynamics of LVQ algorithms for a system with two prototypes and two classes have been investigated in an earlier publication [2]. Here we discuss an

important extension to three prototypes which allows multiple prototypes to be assigned within one class. It is interesting to observe whether this assignment gives an advantage over the simpler single prototype per class and to search for optimal assignments.

5.1 Competition within classes

The combination of behavior from competing prototypes within a class and between different classes is not straightforward. We begin by investigating the effects of competition between multiple prototypes within a class in the LVQ1 algorithm. Here we introduce a one-class problem (formally $p_+ = 1, p_- = 0$) and two competing prototypes $W = w_1, w_2$ with the same class label $c_1 = c_2 = +1$. Without the presence of other classes, prototypes within the same class behave like in unsupervised vector quantization. The prototypes are initialized randomly.

The evolution of the order parameters are depicted in Fig. 1. Although formally the analysis uses the limit $N \to \infty$, we show that they are already in good agreement with Monte Carlo simulations using an artificial data set with dimensionality $N = 50$. The prototypes gradually approach the class center $\lambda \mathbf{B}_+$ and the system reaches a configuration with order parameters $R_{1\sigma} = R_{2\sigma} = \lambda, Q_{11} = Q_{22}$. However, Eq. 2 and $Q_{11} > \lambda^2, Q_{12} \neq Q_{11}$ indicate that the prototypes are not identical vectors located at the exact class center $\lambda \mathbf{B}_+$, but instead they spread out symmetrically in an arbitrary direction. While this behavior produces better data representation [12], its relation to classification ability is not yet clear.

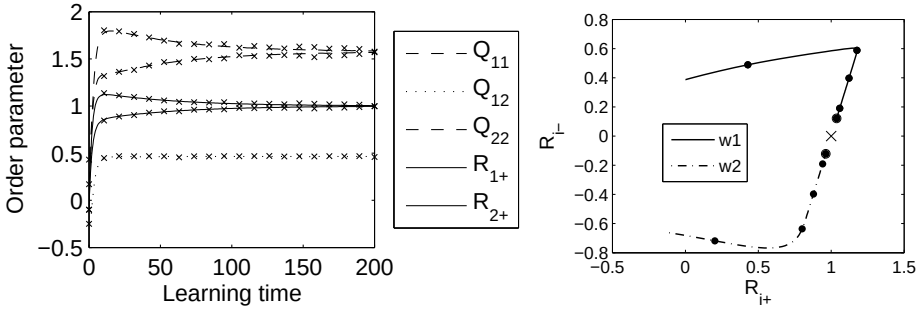

Fig. 1. Left panel: Order parameter $R_{i\sigma}(t), Q_{ij}(t)$ during training using LVQ1 with $p_+ = 1$ (only one class present), $\eta = 0.1$, $\lambda = 1$ and $v_+ = 0.81$. The system is initialized using random prototypes. Crosses mark the corresponding quantities from Monte Carlo simulations using an artificial data set with similar parameters and $N = 50$. **Right panel:** The projection of the prototypes on the plane spanned by $\mathbf{B}_+$, $\mathbf{B}_-$. The cross marks the class center $\lambda \mathbf{B}_+$.

5.2 Optimal class assignment in LVQ1

Now we consider a system with three prototypes with two classes $\sigma = \pm 1$. The prior probabilities p_+, p_- and variance v_+, v_- are set with unequal values in order to break the symmetry between the two classes. As shown in the left panel of Fig. 2, the class labels of the three prototypes can be assigned in two different sets, which are named for shorthand $C_1 = \{+, +, -\}$ or $C_2 = \{+, -, -\}$. Here the parameters are $\lambda = 1, p_+ = 0.6, v_+ = 0.81, v_- = 0.25$.

We compare the generalization errors between the two sets and also to a two prototype system with $C_3 = \{+, -\}$ in the right panel of Fig. 2. In principle, additional prototypes provide the system with more degrees of freedom and allow for more complex decision boundaries. Thus, in an optimal configuration, the generalization error could be lower or at least equal to systems with less prototypes. We indeed observe this result with C_1 where the three-prototype system outperforms the two-prototype system. However, surprisingly, the generalization error with different class assignments C_2 is higher.

Figure 3 shows more general results with the *asymptotic* generalization error $\epsilon_g(t \to \infty)$, i.e. for small learning rates and arbitrary many examples $t \to \infty$, $\eta \to 0$, $\eta t \to \infty$. The performances of different sets are compared as a function of p_+ for unequal variances $v_+ > v_-$. The set C_1 produces the best results while C_2 produces the largest errors. Note that this is valid at all p_+ and therefore the best choice of class labels does not depend on prior probabilities.

Also in Fig. 3, The performances are compared to the best achievable generalization error. For bimodal Gaussian distributions, this optimal decision boundary is hyperquadric [4] where the condition $p_+ P(\xi|+) = p_- P(\xi|-)$ is satisfied. The

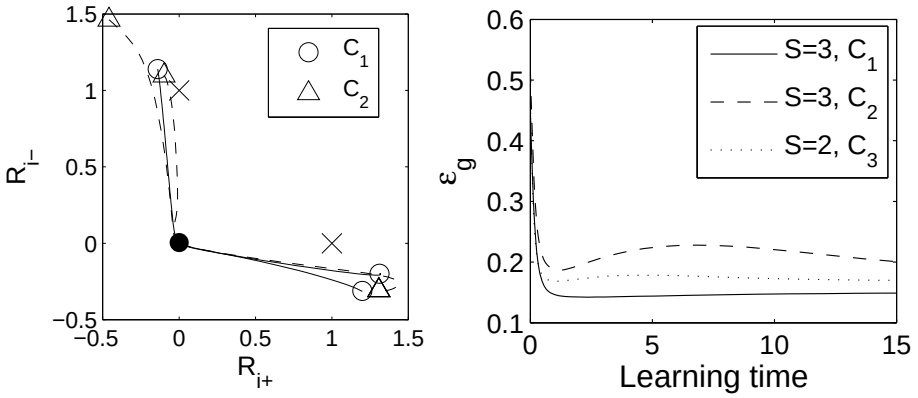

Fig. 2. LVQ1 with $\eta = 0.01, \lambda = 1, p_+ = 0.6, v_+ = 0.81, v_- = 0.25$, $\tilde{t} = \eta t = 15$. **Left panel:** Trajectories of the three prototypes with $C_1 = \{+, +, -\}$ (solid line, ◯) and $C_2 = \{+, -, -\}$ (dashed line, △). **Right panel:** The corresponding learning curves $\epsilon_g(t)$ with C_1, C_2 and two prototypes with $C_3 = \{+, -\}$ (dotted lines).

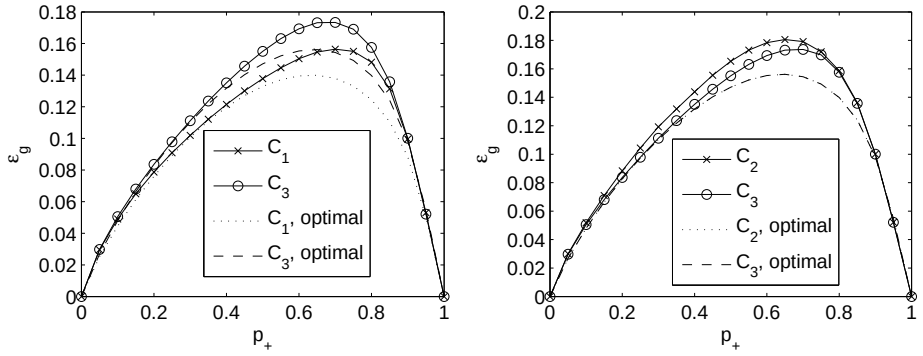

Fig. 3. The asymptotic generalization error as a function of p_+ with other parameters similar as in Fig. 2 for sets C_1, C_2 ($\times$) and a two-prototype system $S = 2, C_3$ ($\bigcirc$). The lowest achievable error for the respective sets are shown by the dotted line and chain line.

shape depends on the variance of both classes, viz. v_+, v_-. In the case $v_+ = v_-$ the decision boundary is a hyperplane, which can be reproduced exactly using two prototypes and so two-prototype systems are already optimal for classification. For unequal variances, e.g. $v_+ > v_-$ it is a concave subspace from $\lambda \mathbf{B}_+$ and convex subspace from $\lambda \mathbf{B}_-$. Multiple prototypes in one class allow for a piecewise decision boundary and a better approximation of the optimal decision boundary. However in a three-prototype system, the decision boundary can only form a convex, wedge-shaped subspace in the class with the single prototype. Therefore the performance can be improved only if two prototypes are assigned to the class with larger variance.

Another observation is shown in Fig. 4, where $v_- = 1$ and the priors are fixed as equal $p_+ = p_-$ (left panel) or $p_+ > p_-$ (right panel). Based on the asymptotic generalization errors, the optimal choice of class assignments depends on v_+, which are divided in three stages: $S = 3, C_2$ (in the left panel, at $v_+ < 0.8$), $S = 2$ (at $0.8 < v_+ < 1.3$) and $S = 3, C_1$ (at $v_+ > 1.3$). Note that the lines of C_1 and C_2 intersect at equal variance $v_+ = v_- = 1$. The behavior is the same for the case of classes with unequal priors, with the three stages shifting towards lower v_+: $S = 3, C_2$ ($v_+ < 0.45$), $S = 2$ (at $0.45 < v_+ < 0.8$) and $S = 3, C_1$ (at $v_+ > 0.8$). These results can be interpreted as (1) multiple prototypes should be assigned to the class with higher variance, which becomes more apparent for largely unequal variance and (2) near equal variances, the two prototype system is already optimal.

To summarize, using the LVQ1 algorithm, multiple prototype systems do not always perform better than simpler single prototype systems, i.e. does not always reduce the generalization error. The LVQ1 algorithm does not explicitly find the minimum generalization error. Also, it has a data representation part as explained in section 5.1. This part however can have negative contribution

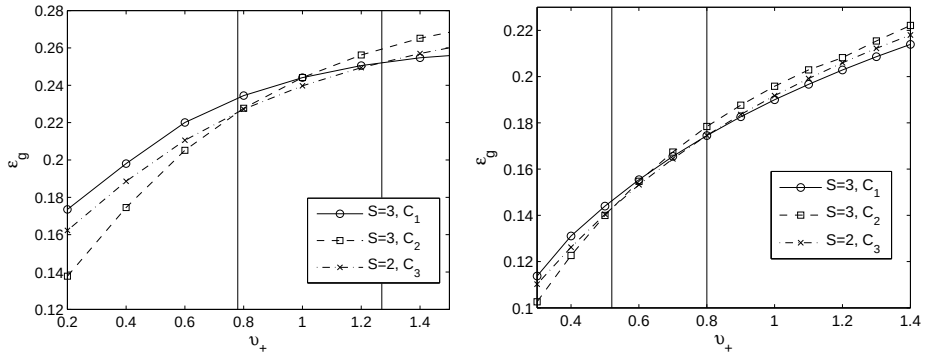

Fig. 4. Asymptotic generalization error as a function of υ_+ where $\upsilon_- = 1$. The prior probabilities are set equal $p_+ = 0.5$ in the left panel and $p_+ = 0.75$ in the right panel The two prototype system is the optimal choice when υ_+ is close to υ_-. C_1 is best when $\upsilon_+ > \upsilon_-$ and C_2 is best when $\upsilon_+ < \upsilon_-$. Vertical lines indicate regions where, respectively from left to right, C_2, C_3 and C_1 produce the best performance.

to classification purposes, e.g. in cases of assigning more prototypes to the class with equal or less or similar variance in the model scenario.

5.3 Stability in LVQ+/-

The LVQ+/- is known to be subject to divergence and stability problems for classes with unequal weights [2]. Prototypes representing the weak classes, ie. the class with lower prior probability, are pushed away frequently by examples of the stronger classes. Without any modifications, the attraction from its own class is outweighed by this repulsion and the system diverges exponentially.

In a two-prototype system, the system becomes highly divergent at all priors except for the singular case of balanced priors [2]. The condition that two nearest prototypes have different classes is always met, and so both prototypes are always updated for each example. On the other hand, in a system with multiple prototypes per class, it is possible that the two nearest prototypes belong to the same class and no update is performed. The result is that the system is more stable, although problems still exist.

An example of a three-prototype system is shown in Fig. 5 for $\lambda = 1$, $\upsilon_+ = \upsilon_- = 1.0$ and unbalanced priors $p_+ = 0.6$, $p_- = 0.4$. In the left panel, the set of class label is $C_1 = \{+, +, -\}$, ie. two prototypes are assigned to the stronger class. The characteristic quantities increase linearly for $R_{i\sigma}$ and quadratically for Q_{ij} with the learning time t. The prototype move toward infinity as $t \to \infty$. Alternately if the labels are $C_2 = \{+, -, -\}$, the system is highly divergent. The characteristic quantities for the two prototype increase exponentially as in the

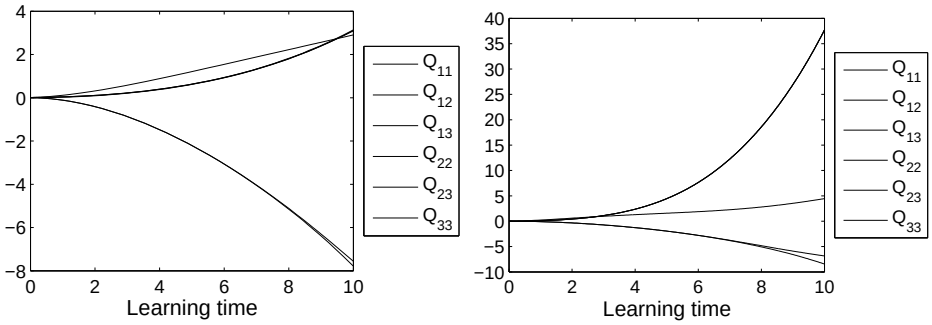

Fig. 5. Evolution of Q_{ij} for LVQ+/- with $\eta = 0.05, \lambda = 1$, $v_+ = v_- = 1.0$ and $p_+ = 0.6$. The classes are C_1 (left panel) and C_2 (right panel). The system displays highly divergent behavior in the case of C_2.

two-prototype system. It is better to assign more prototypes to the stronger class if stability is the main concern.

Several methods have been developed to counter this diverging behavior, e.g. *window* schemes [9]. One conceptually simple approach is an early stopping scheme. Here the learning process is terminated when the system reaches its lowest generalization error and before the performance deteriorates. However, the achievable generalization ability is highly dependent on the initial conditions. The training process becomes a race between the system finding the optimal set before the instability problems occur.

5.4 Integrated algorithms

While the LVQ1 algorithm is intuitive and fast in finding the structure of the data, LVQ+/- is specifically designed for classification purposes. However, the performance of LVQ+/- varies depending on the initial conditions and the training process can have instability problems. Here we combine the advantages of each algorithm by initially using LVQ1 to its asymptotic configuration. Afterwards we switch to LVQ+/- with early stopping using the same configuration to fine tune the decision boundary. The performance after the LVQ+/- can only be better or at least equal than that of LVQ1 because of the early stopping method.

Figure 6 shows the achievable generalization error as a function of the prior probabilities. The variance of each class is $v_+ = 0.81, v_- = 0.25$. In the left panel, the LVQ+/- does not improve the performance of LVQ1. The LVQ1 already performs very well because the class assignments are already optimal, ie. assigning more prototypes on the class with larger variance. On the contrary, the LVQ+/- produces significantly lower generalization errors than LVQ1 when $v_+ < v_-$. LVQ+/- is not as dependent on correct class assignments as LVQ1. Note that for small p_+, LVQ+/- with C_2 rapidly diverges and does not provide advantage over the LVQ1.

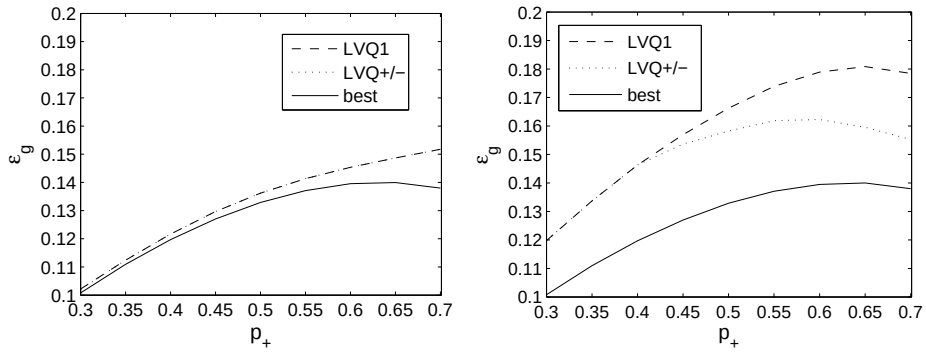

Fig. 6. Generalization error achieved using LVQ1 and then LVQ+/- with early stopping for $\upsilon_+ = 0.81, \upsilon_- = 0.25$, $\eta = 0.1$, after $\tilde{t} = 500$ for LVQ1 and $\tilde{t} = 5$ for the adjustments by LVQ+/-. **Left panel:** $C_1 = \{+, +, -\}$. The plots of LVQ1 and LVQ+/- coincide ie. LVQ+/- does not provide any advantage. **Right panel:** $C_2 = \{+, -, -\}$. LVQ+/- improves the performance because it is less sensitive to suboptimal class assignments.

6 Summary

We investigated the learning dynamics of LVQ algorithms, including LVQ1 and LVQ+/-, for high dimensional data and multiple prototypes within a class. While LVQ1 aims to both represent the data well and good generalization ability, the two goals do not always agree with one another. Introducing multiple prototypes to the system improves representation but may have positive or negative effect on the generalization ability.

The class assignments are vital in the performance of LVQ1 and it is closely related to the structure of the data. The important feature to improve generalization is by assigning the additional prototype(s) to classes with larger variance, and not related explicitly to the prior probabilities. This feature of LVQ1 is useful when there is prior knowledge or hypothesis on the relative variations of different classes.

LVQ+/- with early stopping is less sensitive to suboptimal prototype class assignments and can achieve better performance than LVQ1 in these cases. However, instability remains the main problem in this learning method. Assigning multiple prototypes on a strong class slows down but does not solve the issue of divergence.

Further research is directed amongst others toward more complex and realistic data structures. Also, one should investigate probabilistic labeling where the classes are learned during the course of training. This would avoid the problem of incorrectly adding complexity with minimal benefits or even lower performance.

References

1. Bibliography on the Self Organising Map (SOM) and Learning Vector Quantization (LVQ), Neural Networks Research Centre, Helsinki University of Technology, 2002.
2. M. Biehl, A. Ghosh, and B. Hammer, "Dynamics and Generalization Ability of LVQ Algorithms", *Journal of Machine Learning Research* (8): 323-360 (2007).
3. M. Biehl, A. Freking, A. Ghosh and G. Reents, *A theoretical framework for analysing the dynamics of LVQ*, Technical Report, Technical Report 2004-09-02, Mathematics and Computing Science, University Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands, December 2004, available from *www.cs.rug.nl/∼biehl*.
4. R. Duda, P. Hart and D. Stork, editors. *Pattern Classification*, John Wiley & Sons Inc., 2001.
5. A. Engel and C. van Broeck, editors. *The Statistical Mechanics of Learning*, Cambridge University Press, 2001.
6. A. Ghosh, M. Biehl and B. Hammer, Performance Analysis of LVQ Algorithms: A Statistical Physics Approach, *Neural Networks*, special issue on Advances in Self-Organizing Maps. Vol. 19:817-829, 2006.
7. T. Kohonen, Improved versions of learning vector quantization, *Proc. of the International Joint conference on Neural Networks (San Diego, 1990)*, 1:545-550, 1990.
8. T. Kohonen, Learning vector quantization. In M. Arbib, editor, *The handbook of brain theory and neural networks*, pages 537-540. MIT Press, Cambridge, MA, 1995.
9. T. Kohonen, *Self organising maps*. Springer, Berlin, 1997.
10. G. Reents and R. Urbanczik, Self Averaging and On-line Learning, *Phys. Rev. Letter*, 80:5445-5448, 1998.
11. A. Witoelar and M. Biehl, *Dynamics of multiple prototype LVQ*, Technical Report, in progress, Mathematics and Computing Science, University Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands.
12. A. Witoelar, M. Biehl, A. Ghosh, and B. Hammer, On the Dynamics of Vector Quantization and Neural Gas. In M. Verleysen, editor, *Proc. of European Symposium on Neural Networks (ESANN) 2007*, d-side, Evere, Belgium, 127-132, 2007.