Generative versus Discriminative Prototype Based Classification

Barbara Hammer[1], David Nebel[2], Martin Riedel[2], and Thomas Villmann[2]

[1] Bielefeld University, Germany
bhammer@techfak.uni-bielefeld.de
[2] University of Applied Sciences Mittweida, Germany
{nebel,riedel,villmann}@hs-mittweida.de

Abstract. Prototype-based models such as learning vector quantization (LVQ) enjoy a wide popularity because they combine excellent classification and generalization ability with an intuitive learning paradigm: models are represented by few characteristic prototypes, the latter often being located at class typical positions in the data space. In this article we investigate inhowfar these expectations are actually met by modern LVQ schemes such as robust soft LVQ and generalized LVQ. We show that the mathematical models do not explicitly optimize the objective to find representative prototypes. We demonstrate this fact in a few benchmarks. Further, we investigate the behavior of the models if this objective is explicitly formalized in the mathematical costs. This way, a smooth transition of the two partially contradictory objectives, discriminative power versus model representativity, can be obtained.

1 Introduction

Since its invention by Kohonen [9], learning vector quantization (LVQ) enjoys a great popularity by practitioners for a number of reasons: the learning rule as well as the classification model are very intuitive and fast; the resulting classifier is interpretable since it represents the model in terms of typical prototypes which can be treated in the same way as data; unlike popular alternatives such as SVM the model can easily deal with an arbitrary number of classes; the representation of data in terms of prototypes lends itself to simple incremental learning strategies by referring to the prototypes as statistics for the already learned data. Due to these properties, LVQ has been successfully applied in diverse areas ranging from telecommunications and robotics to the biomedical domain [9,8].

Despite this success, LVQ has long been thought of as a mere heuristic [2] and some mathematical guarantees concerning its convergence properties or its generalization ability have been investigated more than ten years after its invention only [3,1,13]. Today, LVQ is usually no longer used in its basic form, rather variants which can be derived from mathematical cost functions are used such as generalized LVQ (GLVQ) [12], robust soft LVQ (RSLVQ) [16], or soft nearest prototype classification [15]. Further, one of the success stories of LVQ is linked to its combination with more powerful, possibly adaptive metrics instead

T. Villmann et al. (eds.), *Advances in Self-Organizing Maps and Learning Vector Quantization,* Advances in Intelligent Systems and Computing 295,
DOI: 10.1007/978-3-319-07695-9_12,

of the standard Euclidean one, including, for example, an adaptive quadratic form [7,13], a general kernel [11,6], a functional metric [17], or extensions to discrete data structures [4].

Depending on the application domain, the objective of LVQ to find a highly discriminative classifier is accompanied by additional demands such as sparsity of the models or model interpretability. Modern LVQ techniques such as RSLVQ or GLVQ are explicitly derived from cost functions, such that it is possible to link the objectives of a practitioner to the mathematical objective as modeled in these cost functions. In this contribution, we argue that, while often used as an interpretable model, the objective of arriving at representative prototypes is usually not included in this mathematical objective. We propose an extension of LVQ schemes which explicitly takes this objective into account and which allows a weighting of the two partially contradictory objectives of discriminative power and representativity. We demonstrate the behavior of the resulting models in benchmark data sets where, depending on the setting, models with very different characteristics can be obtained this way.

2 LVQ Schemes

A LVQ classifier is given by a set of prototypes $\boldsymbol{w}_i \in \mathbb{R}^n$, $i = 1, \ldots, k$ together with their labeling $c(\boldsymbol{w}_i) \in \{1, \ldots, C\}$, assuming C classes. Classification of a point $\boldsymbol{x} \in \mathbb{R}^n$ takes place by a winner takes all scheme: $\boldsymbol{x}$ is mapped to the label $c(\boldsymbol{x}) = c(\boldsymbol{w}_i)$ of the prototype $\boldsymbol{w}_i$ which is closest to $\boldsymbol{x}$ as measured in some distance measure, a probability in case of a RSLVQ classifier, respectively. For simplicity, we restrict to the Euclidean metric, even though general metrics could be used.

Given a training data set $\boldsymbol{x}_j \in \mathbb{R}^n$, $j = 1, \ldots, m$, together with labels $y_j \in \{1, \ldots, C\}$, LVQ aims at finding prototypes such that the resulting classifier achieves a good classification accuracy, i.e. $y_j = c(\boldsymbol{x}_j)$ for as many j as possible. Classical LVQ schemes such as LVQ 1 or LVQ 2.1 rely on Hebbian learning heuristics, but they do not relate to a valid underlying cost function in the case of a continuous data distribution [2]. A few alternative models have been proposed which are derived from explicit cost functions and which lead to learning rules resembling the update rules of classical LVQ schemes [12,16].

Generalized LVQ (GLVQ) [12] addresses the following cost function

$$E = \sum_j \Phi \left(\frac{d^+(\boldsymbol{x}_j) - d^-(\boldsymbol{x}_j)}{d^+(\boldsymbol{x}_j) + d^-(\boldsymbol{x}_j)} \right) \tag{1}$$

where $d^+(\boldsymbol{x}_j)$ refers to the squared Euclidean distance of $\boldsymbol{x}_j$ to the closest prototype labeled with y_j, and $d^-(\boldsymbol{x}_j)$ refers to the squared Euclidean distance of $\boldsymbol{x}_j$ to the closest prototype labeled with a label different from y_j. Φ refers to a monotonic function such as the identity or the sigmoidal function. Optimization typically takes place using a gradient technique. As argued in [13], the numerator of the summands can be linked to the so-called hypothesis margin of the classifier, such that a large margin and hence good generalization ability is aimed for

while training. The denominator prevents divergence and numerical instabilities by normalizing the costs.

Robust soft LVQ (RSLVQ) [16] yields similar update rules based on the following probabilistic model

$$E = \sum_j \log \frac{p(\boldsymbol{x}_j, y_j|W)}{p(\boldsymbol{x}_j|W)} = \sum_j \log p(y_j|\boldsymbol{x}_j, W) \tag{2}$$

where $p(\boldsymbol{x}_j|W) = \sum_i p(\boldsymbol{w}_i)p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ constitutes a mixture of Gaussians with prior probability $p(\boldsymbol{w}_i)$ (often taken uniformly over all prototypes) and probability $p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ of the point $\boldsymbol{x}_j$ being generated from prototype $\boldsymbol{w}_i$, usually taken as an isotropic Gaussian centered in $\boldsymbol{w}_i$, or a slightly extended version described by a diagonal covariance matrix. The probability $p(\boldsymbol{x}_j, y_j|W) = \sum_i \delta_{y_j}^{c(\boldsymbol{w}_i)} p(\boldsymbol{w}_i)p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ (δ - Kronecker delta) restricts to the mixture components with the correct labeling.This likelihood ratio is optimized using a gradient technique.

When inspecting these cost functions, the question occurs to what extend these LVQ schemes mirror the following objectives:

- **Discriminative Power:** the primary objective of LVQ schemes is to provide a classifier with small classification error on the underlying data distribution. Thus, its objective is to minimize the training error and, more importantly, classification error for new data points.
- **Representativity:** the resulting prototypes should represent the data in an accurate way such that it is possible to interpret the model by inspecting the learned prototypes.

Inhowfar are these objectives accounted for by the GLVQ or RSLVQ costs? Interestingly, RSLVQ aims at a direct optimization of the Bayesian error. Hence, its primary goal is the discriminative power of the model. RSLVQ has no incentive to find representative prototypes unless this fact directly contributes to a good discriminative model. This behavior has been observed in practice [14]: prototypes usually do not lie at class typical positions; they can be located outside the convex hull of the data, for example, provided a better classification accuracy. This behavior has also theoretically been investigated for the limit of small bandwidth in [1]: in the limit of small bandwidth, learning from mistakes takes place, i.e. prototype locations are adapted only if misclassifications are present. We will show one such example for original RSLVQ in the experiments.

What about the GLVQ costs? The numerator of GLVQ is negative if and only if the classification of the considered data point is correct. In addition, it resembles the hypothesis margin of the classifier. Due to this fact, one can expect a high correlation of the classification error and the cost function, making GLVQ suitable as a discriminative model. Nevertheless, since this correlation is not an exact equivalence, minima of this cost function do not necessarily correspond to good classifications in all situations: for highly imbalanced data, for example, the GLVQ costs prefer trivial solutions with all data being assigned to the majority class. This observation is also demonstrated by the fact that the classification

accuracy of GLVQ can be inferior as compared to RSLVQ, the latter focussing on discrimination only, see e.g. [14] and our results in the experiments section.

Interestingly, the GLVQ costs have a mild tendency to find representative prototypes due to this form: The term $d^+(\boldsymbol{x})$ in the numerator aims at a small class-wise quantization error of the data. Further, solutions with small denominator are preferred, i.e. there is an emphasis to place all prototypes within the data set. We will see in experiments, that this compromise of representativity and discriminative behavior can yield to classification results inferior to RSLVQ for the sake of more representative models, but still an increase of model representativity is possible by adding a corresponding term to the costs.

3 Extending LVQ Schemes by Generative Modes

We are interested in a model-consistent extension of the RSLVQ and GLVQ costs which explicitly take the goal of representativity into account. Generally, we refer to the cost function of RSLVQ (2) or GLVQ (1) as $E_{\text{discr}}(W)$. The idea is to substitute these costs by the form

$$E = (1-\alpha) \cdot E_{\text{discr}}(W) + \alpha \cdot E_{\text{repr}}(W) \tag{3}$$

where $E_{\text{repr}}(W)$ emphasizes the objective to find representative prototypes $\boldsymbol{w}_j$. The parameter $\alpha \in [0,1]$ weights the influence of both parts for the optimization.

First, we have a look at how to choose $E_{\text{repr}}(W)$ for RSLVQ schemes. The idea is to add a term which maximizes the likelihood of the observed data being generated by the underlying model. Similar to RSLVQ, we can consider a class-wise Gaussian mixture model $p(\boldsymbol{x}_j, y_j|W) = \sum_i \delta_{y_j}^{c(\boldsymbol{w}_i)} p(\boldsymbol{w}_i) p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ with prior probability $p(\boldsymbol{w}_i)$ and Gaussian $p(\boldsymbol{x}_j|\boldsymbol{w}_i)$. The costs aim at a generative model, i.e. we address the class-wise data log likelihood $\log \prod_j \delta_{y_j}^c p(\boldsymbol{x}_j|c,W) = \sum_j \delta_{y_j}^c \log \sum_i \delta_{y_j}^{c(\boldsymbol{w}_i)} p_c(\boldsymbol{w}_i) p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ with prior $p_c(\boldsymbol{w}_i) = p(\boldsymbol{w}_i)/p(c)$ summing to one for every class c. Adding this generative term for all class-wise distributions, we arrive at the form

$$E_{\text{repr}}(W) = \sum_c \sum_j \delta_{y_j}^c \log \sum_i \delta_{y_j}^{c(\boldsymbol{w}_i)} p_c(\boldsymbol{w}_i) p(\boldsymbol{x}_j|\boldsymbol{w}_i) \tag{4}$$

We often assume equal prior for all classes c and prototypes $\boldsymbol{w}_i$ for simplicity. We choose Gaussians of the form

$$p(\boldsymbol{x}_j|\boldsymbol{w}_i) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_i|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{w}_i)^T \Sigma_i^{-1} (\boldsymbol{x}_j - \boldsymbol{w}_i)\right) \tag{5}$$

where Σ_i is taken as diagonal matrix with entries $(\sigma_{i1}^2, \ldots, \sigma_{in}^2)$. Optimization takes place by means of a gradient ascent of these costs. The derivative of $E_{\text{discr}}(W)$ can be found in [16]. See [14] for update rules in case of an adaptive covariance matrix. For $E_{\text{repr}}(W)$ prototypes $\boldsymbol{w}_i$ are adapted according to

$$\frac{\partial E_{\text{repr}}(W)}{\partial \boldsymbol{w}_i} = \sum_j \delta_{y_j}^{c(\boldsymbol{w}_i)} \frac{p_{y_j}(\boldsymbol{w}_i) \cdot p(\boldsymbol{x}_j|\boldsymbol{w}_i)}{\sum_l \delta_{y_j}^{c(\boldsymbol{w}_l)} p_{y_j}(\boldsymbol{w}_l) \cdot p(\boldsymbol{x}_j|\boldsymbol{w}_l)} \cdot \Sigma_i^{-1} \cdot (\boldsymbol{x}_j - \boldsymbol{w}_i), \tag{6}$$

while the variances σ_{jk} are simultaneously updated referred to

$$\frac{\partial E_{\text{repr}}(W)}{\partial \sigma_{in}} = \sum_j \delta_{y_j}^{c(\boldsymbol{w}_i)} \frac{p(\boldsymbol{w}_i)p(\boldsymbol{x}_j|\boldsymbol{w}_i)}{p(\boldsymbol{x}_j, y_j|W)} \left(\frac{[\boldsymbol{x}_j - \boldsymbol{w}_i]_n^2}{\sigma_{in}^3} - \frac{1}{\sigma_{in}} \right). \tag{7}$$

In the limit of small bandwidth, this amounts to a class wise vector quantization scheme.

In a similar way, we enhance the GLVQ cost function by a term emphasizing the representativity of the prototypes in model consistent way. Here we choose the class-wise quantization error

$$E_{\text{repr}}(W) = \sum_j d^+(\boldsymbol{x}_j), \tag{8}$$

Taking the derivative overlays the update rules with a vector quantization step.

As we will see in experiments, depending on the data set, these two objectives can be contradictory, such that the choice of α can severely influence the outcome. Thereby, the scaling of the two objectives is not clear a priori: while a probabilistic modeling such as RSLVQ places the two objectives into the interval $(-\infty, 0]$ corresponding to a log likelihood, the discriminative part of GLVQ lies in $E_{\text{discr}}(W) \in (-1, 1)$, but $E_{\text{repr}}(W) \in [0, \infty)$ for GLVQ. Hence, without normalizing these terms, the scaling of the parameter α has different meanings in both settings. We will report results for the whole range $\alpha \in [0, 1]$ with step size 0.05 in case of RSLVQ, 0.001 for GLVQ, respectively.

4 Experiments

We test the behavior of the models for different values α in three benchmarks:

- **Gauss:** two two-dimensional Gaussian clusters with different covariance matrices and some degree of overlap are generated.
- **Tecator:** the data set consists of 215 spectra with 100 spectral bands ranging from 850 nm to 1050 nm [10]. The task is to predict the fat content of the probes.

To avoid local optima as much as possible, initial training takes place to distribute the prototypes in the data space, as proposed in [9]. In our experiments we simply start with an initial training phase where $\alpha = 1$ and we anneal the value α afterwards to the desired weighting parameter. For RSLVQ, diagonal entries of the covariance matrix are adapted individually for every mixture component. In all cases, we use one prototype/mixture component per class. Training takes place until convergence. To validate representativity we determine the following ratio for both models:

$$R = \frac{1}{C} \sum_{c_k} \sum_{j:\, c(\boldsymbol{x}_j)=c_k} \frac{d^+(\boldsymbol{x}_j)}{\sum_{i:\, c(\boldsymbol{x}_i)=c_k} d\left(\boldsymbol{x}_i, \boldsymbol{\mu}_{c_k}\right)}, \tag{9}$$

which is the class-wise quantization error according to the class mean $\boldsymbol{\mu}_{c_k}$.

Gauss: Due to the data generation, prototypes lying in the two class centers define a decision boundary which is close to the optimum decision boundary, albeit not being identical due to the non-isotropic Gaussians. This fact is mirrored in the dependency of the classification accuracy in respect to the parameter α as depicted in Fig. 1: the accuracy is widely constant for varying parameter α for both, RSLVQ and GLVQ schemes.

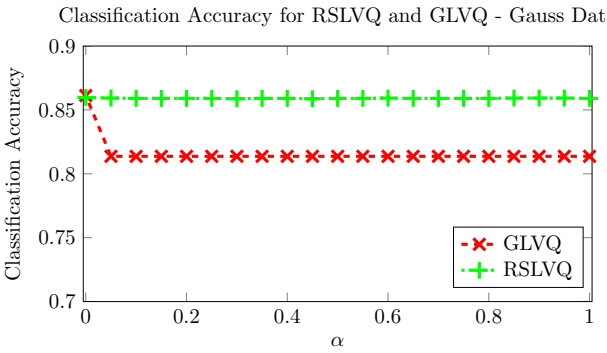

Fig. 1. Classification accuracy for RSLVQ and GLVQ for the Gauss data set varying parameter α

Interestingly, the classification accuracy for RSLVQ is higher than GLVQ which can be attributed to the fact that only the first model explicitly aims at an optimization of the Bayes error and an implicitly fitting of Gaussians, while the GLVQ costs are only correlated to a class discrimination.

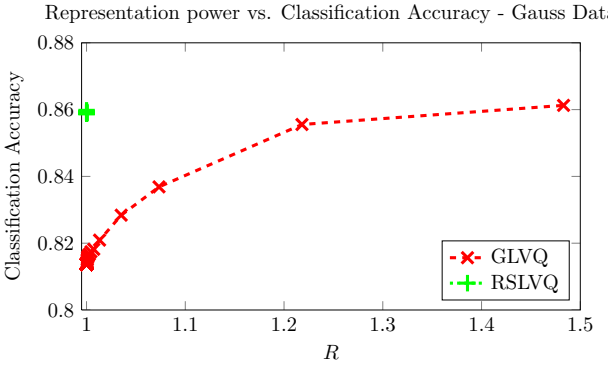

Fig. 2. Class-wise quantization error for the Gauss data set vs. accuracy for varying parameter α

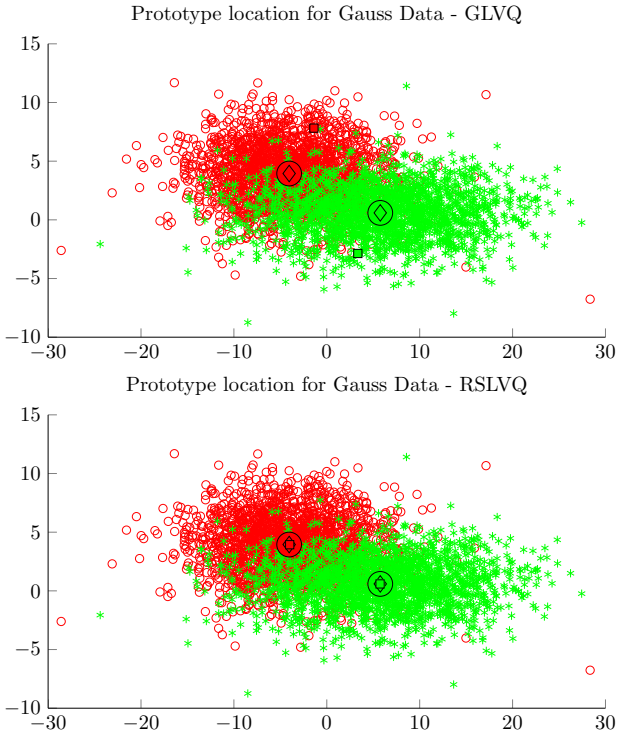

Fig. 3. Prototype location for the Gauss data set for extremal $\alpha \in \{0, 1\}$. squares $\hat{=}$ $\alpha = 1$; diamonds $\hat{=}$ $\alpha = 0$; filled circle $\hat{=}$ class mean.

For both approaches the prototype locations for extremal values $\alpha \in \{0, 1\}$ are depicted in Fig. 3. The prototypes which are obtained with RSLVQ do not change its position, as mirrored in the class-wise quantization error with increasing value α, see Fig. 2. These are at the class centers and obviously do not enormously differ from the respective class means. Unlike GLVQ, where for $\alpha = 1$ the prototypes do not coincide with the class means to better follow the optimum decision boundary for the given case. Contrary to RSLVQ, covariances are not used by standard GLVQ.

Tecator: For the tecator data set, there seems a clear difference between a good generative or good discriminative model as found by LVQ schemes. When varying the parameter α, the classification accuracy decreases (Fig. 4), while the representativity increases, see Fig. 5.

Interestingly, the prototypes lie at atypical positions for the purely discriminative models in this case, making their interpretability problematic: as depicted in Fig. 6, the spectral curves display a very characteristic shape which has no resemblance to spectra as observed in the data. These forms facilitate the class discrimination while interpretability is questionable. This setting also demonstrates

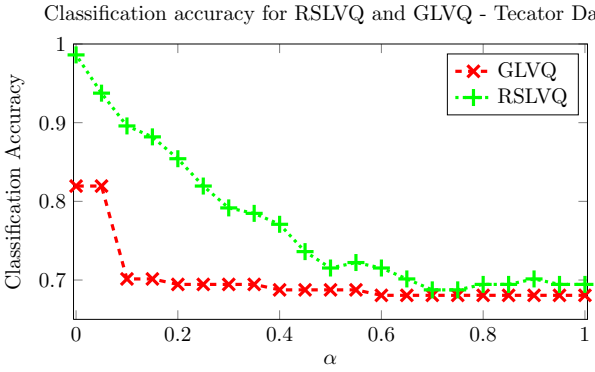

Fig. 4. Classification accuracy for RSLVQ and GLVQ for the Tecator data set varying parameter α

the partially problematic choice of an appropriate parameter α in particular for the GLVQ model. In this case, due to the inherent scaling, already small values of α have a dramatic effect on the classification accuracy of the result.

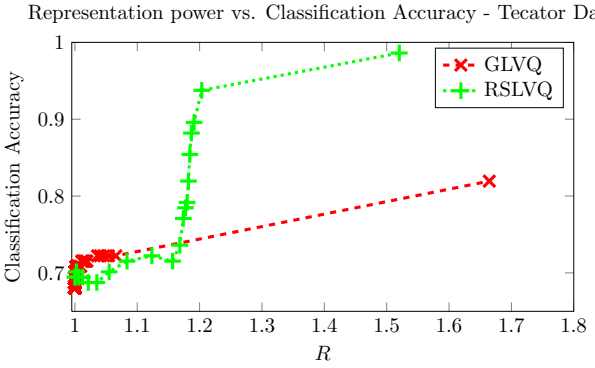

Fig. 5. Class-wise quantization error for RSLVQ and GLVQ for the Tecator data set vs. Classification accuracy for varying parameter α

5 Discussions

We have discussed the correlation of popular LVQ cost functions to the two aims, to obtain a small classification error and to obtain a representative model where prototypes are interpretable. By means of examples, we have seen that LVQ usually models the former objective, but the latter is only implicitly taken into account. An explicit integration of this objective enables enhanced models where the discriminative power versus the representativity of the prototypes can be controlled by the user, leading to better interpretable models in case the two

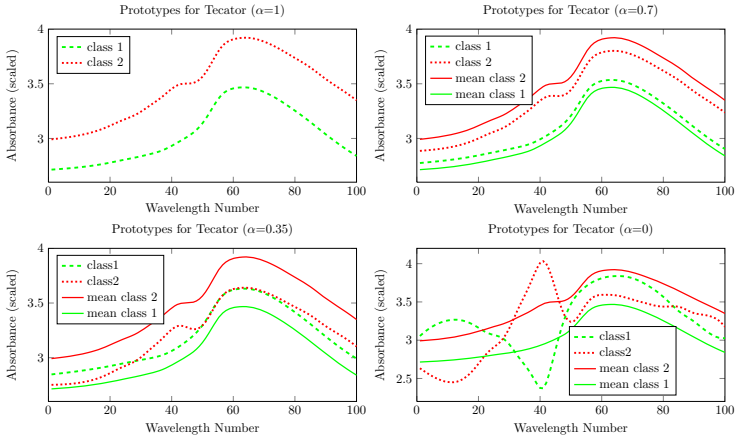

Fig. 6. Prototype locations for the Tecator data set and different choices of the parameter α. Interestingly, for the discriminative case $\alpha = 0$, atypical shapes with little resemblance of the class averages are obtained, while $\alpha = 1$ boosts class averages.

objectives are contradictory for the given data. We have shown the effect of such a control on the form of the prototypes in a few benchmarks.

So far, the two objectives are combined in one cost function and an appropriate balance parameter α has to be set. To make both algorithms comparable according to the used distance a localized relevance GLVQ approach [5] is mandatory. In this contribution our focus is on pointing out that both LVQ variants can be extended to make their results more interpretable. As an alternative, one can consider formulations which emphasize the primary aim of correct classification as a hard constraint, but integrate representativity as a soft constraint. This way, one can aim for the most representative solutions among a set of possible solutions which are invariant with respect to the classification error. Such an approach would result in formalizations of the form

$$\min \sum_j d^+(\boldsymbol{x}_j)$$
$$\text{such that } \; d^+(\boldsymbol{x}_j) \leq d^-(\boldsymbol{x}_j) + \epsilon \quad \forall j$$

for GLVQ, incorporating slack variables if no feasible solution exists, or

$$\max \sum_c \sum_j \delta^c_{y_j} \log \sum_i \delta^{c(\boldsymbol{w}_i)}_{y_j} p_c(\boldsymbol{w}_i) p(\boldsymbol{x}_j|\boldsymbol{w}_i)$$
$$\text{such that } \; p(y_j|\boldsymbol{x}_j, W) \geq p(c|\boldsymbol{x}_j, W) + \epsilon \quad \forall j \quad \forall c \neq y_j$$

for RSLVQ, again incorporating slack variables if necessary. The investigation of these alternatives will be the subject of future work.

Acknowledgement. BH has been supported by the CITEC center of excellence. DN and MR acknowledge funding by ESF.

References

1. Biehl, M., Ghosh, A., Hammer, B.: Dynamics and generalization ability of LVQ algorithms. Journal of Machine Learning Research 8, 323–360 (2007)
2. Biehl, M., Hammer, B., Schneider, P., Villmann, T.: Metric learning for prototype based classification. In: Bianchini, M., Maggini, M., Scarselli, F. (eds.) Innovations in Neural Information – Paradigms and Applications. SCI, vol. 247, pp. 183–199. Springer, Heidelberg (2009)
3. Crammer, K., Gilad-Bachrach, R., Navot, A., Tishby, A.: Margin analysis of the lvq algorithm. In: Advances in Neural Information Processing Systems, vol. 15, pp. 462–469. MIT Press, Cambridge (2003)
4. Hammer, B., Mokbel, B., Schleif, F.-M., Zhu, X.: White box classification of dissimilarity data. In: Corchado, E., Snáv̊sel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part III. LNCS, vol. 7208, pp. 309–321. Springer, Heidelberg (2012)
5. Hammer, B., Schleif, F.-M., Villmann, T.: On the generalization ability of prototype-based classifiers with local relevance determination (2005)
6. Hammer, B., Strickert, M., Villmann, T.: Supervised neural gas with general similarity measure. Neural Processing Letters 21(1), 21–44 (2005)
7. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Networks 15(8-9), 1059–1068 (2002)
8. Kirstein, S., Wersing, H., Körner, E.: A biologically motivated visual memory architecture for online learning of objects. Neural Networks 21(1), 65–77 (2008)
9. Kohonen, T.: The self-organizing map. Proc. of the IEEE 78(9), 1464–1480 (1990)
10. D. of Statistics at Carnegie Mellon University, `http://lib.stat.cmu.edu/datasets/`
11. Qin, A.K., Suganthan, P.N.: A novel kernel prototype-based learning algorithm. In: ICPR (4), pp. 621–624 (2004)
12. Sato, A., Yamada, K.: Generalized learning vector quantization. In: Touretzky, M.C.M.D.S., Hasselmo, M.E. (eds.) Proceedings of the 1995 Conference on Advances in Neural Information Processing Systems 8, pp. 423–429. MIT Press, Cambridge (1996)
13. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in learning vector quantization. Neural Computation 21, 3532–3561 (2009)
14. Schneider, P., Biehl, M., Hammer, B.: Distance learning in discriminative vector quantization. Neural Computation 21, 2942–2969 (2009)
15. Seo, S., Bode, M., Obermayer, K.: Soft nearest prototype classification. IEEE Transactions on Neural Networks 14, 390–398 (2003)
16. Seo, S., Obermayer, K.: Soft learning vector quantization. Neural Computation 15(7), 1589–1604 (2003)
17. Villmann, T., Haase, S.: Divergence-based vector quantization. Neural Computation 23(5), 1343–1392 (2011)