Relational Extensions of Learning Vector Quantization

Barbara Hammer, Frank-Michael Schleif, and Xibin Zhu

CITEC center of excellence, Bielefeld University,
33615 Bielefeld, Germany
{bhammer|fschleif|xzhu}@techfak.uni-bielefeld.de

Abstract. Prototype based models offer an intuitive interface to given data sets by means of an inspection of the model prototypes. Supervised classification can be achieved by popular techniques such as learning vector quantization (LVQ) and extensions derived from cost functions such as generalized LVQ (GLVQ) and robust soft LVQ (RSLVQ). These methods, however, are restricted to Euclidean vectors and they cannot be used if data are characterized by a general dissimilarity matrix. In this approach, we propose relational extensions of GLVQ and RSLVQ which can directly be applied to general possibly non-Euclidean data sets characterized by a symmetric dissimilarity matrix.

1 Introduction

Machine learning techniques have revolutionized the possibility to deal with large electronic data sets by offering powerful tools to automatically learn a regularity underlying the data. However, some of the most powerful machine learning tools which are available today such as the support vector machine act as a black box and their decisions cannot easily be inspected by humans. In contrast, prototype-based methods represent their decisions in terms of typical representatives contained in the input space. Since prototypes can directly be inspected by humans in the same way as data points, an intuitive access to the decision becomes possible: the responsible prototype and its similarity to the given data determine the output.

There exist different possibilities to infer appropriate prototypes from data: Unsupervised learning such as simple k-means, fuzzy-k-means, topographic mapping, neural gas, or the self-organizing map, and statistical counterparts such as the generative topographic mapping infer prototypes based on input data only [1–3]. Supervised techniques incorporate class labeling and find decision boundaries which describe priorly known class labels, one of the most popular learning algorithm in this context being learning vector quantization (LVQ) and extensions thereof which are derived from explicit cost functions or statistical models [2, 4, 5]. Besides different mathematical derivations, these learning algorithms share several fundamental aspects: they represent data in a sparse way by means of prototypes, they form decisions based on the similarity of data to prototypes, and training is often very intuitive based on Hebbian principles. In

addition, prototype-based models have excellent generalization ability [6, 7]. Further, prototypes offer a compact representation of data which can be beneficial for life-long learning, see e.g. the approaches proposed in [8–10].

LVQ severely depends on the underlying metric, which is usually chosen as Euclicean metric. Thus, it is unsuitable for complex or heterogeneous data sets where input dimensions have different relevance or a high dimensionality yields to accumulated noise which disrupts the classification. This problem can partially be avoided by appropriate metric learning, see e.g. [7], or by kernel variants, see e.g. [11]. However, if data are inherently non-Euclidean, these techniques cannot be applied. In modern applications, data are often addressed using dedicated non-Euclidean dissimilarities such as dynamic time warping for time series, alignment for symbolic strings, the compression distance to compare sequences based on an information theoretic ground, and similar. These settings do not allow a Euclidean representation of data at all, rather, data are given implicitly in terms of pairwise dissimilarities or relations; we refer to a 'relational data representation' in the following when addressing such settings.

In this contribution, we propose relational extensions of two popular LVQ algorithms derived from cost functions, generalized LVQ (GLVQ) and robust soft LVQ (RSLVQ), respectively [4, 5]. This way, these techniques become directly applicable for relational data sets which are characterized in terms of a symmetric dissimilarity matrix only. The key ingredient is taken from recent approaches for relational data processing in the unsupervised domain [12, 13]: if prototypes are represented implicitly as linear combinations of data in the so-called pseudo-Euclidean embedding, the relevant distances of data and prototypes can be computed without an explicit reference to a vectorial data representation. This principle holds for every symmetric dissimilarity matrix and thus, allows us to formalize a valid objective of RSLVQ and GLVQ for relational data. Based on this observation, optimization can take place using gradient techniques.

In this contribution, we shortly review LVQ techniques derived from a cost function, and we extend these techniques to relational data. We test the technique on several benchmarks, leading to results comparable to SVM while providing prototype based presentations.

2 Prototype-based Clustering and Classification

Assume data $\boldsymbol{x}^i \in \mathbb{R}^n, i = 1, \ldots, m$, are given. Prototypes are elements $\boldsymbol{w}^j \in \mathbb{R}^n, j = 1, \ldots, k$, of the same space. They decompose data into receptive fields $R(\boldsymbol{w}^j) = \{\boldsymbol{x}^i \ : \ \forall k \ d(\boldsymbol{x}^i, \boldsymbol{w}^j) \leq d(\boldsymbol{x}^i, \boldsymbol{w}^k)\}$ based on the squared Euclidean distance $d(\boldsymbol{x}^i, \boldsymbol{w}^j) = \|\boldsymbol{x}^i - \boldsymbol{w}^j\|^2$. The goal of prototype-based machine learning techniques is to find prototypes which represent a given data set as accurately as possible.

In supervised settings, data $\boldsymbol{x}^i$ are equipped with class labels $c(\boldsymbol{x}^i) \in \{1, \ldots, L\}$ in a finite set of known classes. Similarly, every prototype is equipped with a priorly fixed class label $c(\boldsymbol{w}^j)$. A data point is mapped to the class of its closest prototype. The classification error of this mapping is given by the term $\sum_j \sum_{\boldsymbol{x}^i \in R(\boldsymbol{w}^j)} \delta(c(\boldsymbol{x}^i) \neq c(\boldsymbol{w}^j))$ with the delta function δ. This cost function cannot easily be optimized explicitly due to vanishing gradients and discontinuities. Therefore, LVQ relies on a reasonable heuristic by performing Hebbian

and unti-Hebbian updates of the prototypes, given a data point [2]. Extensions of LVQ derive similar update rules from explicit cost functions which are related to the classification error, but display better numerical properties such that optimization algorithms can be derived thereof.

Generalized LVQ (GLVQ) has been proposed in the approach [4]. It is derived from a cost function which can be related to the generalization ability of LVQ classifiers [7]. The cost function of GLVQ is given as

$$E_{\mathrm{GLVQ}} = \sum_i \Phi\left(\frac{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) - d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}\right) \tag{1}$$

where Φ is a differentiable monotonic function such as the hyperbolic tangent, and $\boldsymbol{w}^+(\boldsymbol{x}^i)$ refers to the prototype closest to $\boldsymbol{x}^i$ with the same label as $\boldsymbol{x}^i$, $\boldsymbol{w}^-(\boldsymbol{x}^i)$ refers to the closest prototype with a different label. This way, for every data point, its contribution to the cost function is small if and only if the distance to the closest prototype with a correct label is smaller than the distance to a wrongly labeled prototype, resulting in a correct classification of the point and, at the same time, by optimizing this so-called hypothesis margin of the classifier, aiming at a good generalization ability.

A learning algorithm can be derived thereof by means of a stochastic gradient descent. After a random initialization of prototypes, data $\boldsymbol{x}^i$ are presented in random order. Adaptation of the closest correct and wrong prototype takes place by means of the update rules

$$\Delta\boldsymbol{w}^{\pm}(\boldsymbol{x}^i) \sim \mp\Phi'(\mu(\boldsymbol{x}^i)) \cdot \mu^{\pm}(\boldsymbol{x}^i) \cdot \nabla_{\boldsymbol{w}^{\pm}(\boldsymbol{x}^i)} d(\boldsymbol{x}^i, \boldsymbol{w}^{\pm}(\boldsymbol{x}^i)) \tag{2}$$

where

$$\mu(\boldsymbol{x}^i) = \frac{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) - d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))}{d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))},\ \mu^{\pm}(\boldsymbol{x}^i) = \frac{2 \cdot d(\boldsymbol{x}^i, \boldsymbol{w}^{\mp}(\boldsymbol{x}^i))}{(d(\boldsymbol{x}^i, \boldsymbol{w}^+(\boldsymbol{x}^i)) + d(\boldsymbol{x}^i, \boldsymbol{w}^-(\boldsymbol{x}^i))^2}. \tag{3}$$

For the squared Euclidean norm, the derivative yields $\nabla_{\boldsymbol{w}^j} d(\boldsymbol{x}^i, \boldsymbol{w}^j) = -2(\boldsymbol{x}^i - \boldsymbol{w}^j)$, leading to Hebbian update rules of the prototypes which take into account the priorly known class information, i.e. they adapt the closest prototypes towards / away from a given data point depending on their labels. GLVQ constitutes one particularly efficient method to adapt the prototypes according to a given labeled data sets.

Robust soft LVQ (RSLVQ) as proposed in [5] constitutes an alternative approach which is based on a statistical model of the data. In the limit of small bandwidth, update rules which are very similar to LVQ result. For non-vanishing bandwidth, soft assignments of data points to prototypes take place. Every prototype induces a probability induced by Gaussians, for example, i.e. $p(\boldsymbol{x}^i|\boldsymbol{w}^j) = K \cdot \exp(-d(\boldsymbol{x}^i, \boldsymbol{w}^j)/2\sigma^2)$ with parameter $\sigma \in \mathbb{R}$ and normalization constant $K = (2\pi\sigma^2)^{-n/2}$. Assuming that every prototype has the same prior, we obtain the overall probability of a data point $p(\boldsymbol{x}^i) = \sum_{\boldsymbol{w}^j} p(\boldsymbol{x}^i|\boldsymbol{w}^j)/k$ and the probability of a point and its corresponding class $p(\boldsymbol{x}^i, c(\boldsymbol{x}^i)) = \sum_{\boldsymbol{w}^j : c(\boldsymbol{w}^j) = c(\boldsymbol{x}^i)} p(\boldsymbol{x}^i|\boldsymbol{w}^j)/k$. The cost function of RSLVQ is given by the quotient

$$E_{\mathrm{RSLVQ}} = \log \prod_i \frac{p(\boldsymbol{x}^i, c(\boldsymbol{x}^i))}{p(\boldsymbol{x}^i)} = \sum_i \log \frac{p(\boldsymbol{x}^i, c(\boldsymbol{x}^i))}{p(\boldsymbol{x}^i)} \tag{4}$$

Considering gradients, we obtain the adaptation rule for every prototype $\boldsymbol{w}^j$ given a training point $\boldsymbol{x}^i$

$$\Delta\boldsymbol{w}^j \sim -\frac{1}{2\sigma^2}\cdot\left(\frac{p(\boldsymbol{x}^i|\boldsymbol{w}^j)}{\sum_{j:c(\boldsymbol{w}^j)=c(\boldsymbol{x}^i)}p(\boldsymbol{x}^i|\boldsymbol{w}^j)}-\frac{p(\boldsymbol{x}^i|\boldsymbol{w}^j)}{\sum_j p(\boldsymbol{x}^i|\boldsymbol{w}^j)}\right)\cdot\nabla_{\boldsymbol{w}^j}d(\boldsymbol{x}^i,\boldsymbol{w}^j) \quad (5)$$

if $c(\boldsymbol{x}^i)=c(\boldsymbol{w}^j)$ and $\Delta\boldsymbol{w}^j \sim \frac{1}{2\sigma^2}\cdot\frac{p(\boldsymbol{x}^i|\boldsymbol{w}^j)}{\sum_j p(\boldsymbol{x}^i|\boldsymbol{w}^j)}\cdot\nabla_{\boldsymbol{w}^j}d(\boldsymbol{x}^i,\boldsymbol{w}^j)$ if $c(\boldsymbol{x}^i)\neq c(\boldsymbol{w}^j)$. Obviously, the scaling factors can be interpreted as soft assignments of the data to corresponding prototypes. The choice of an appropriate parameter σ can critically influence the overall behavior and the quality of the technique, see e.g. [5, 14, 15] for comparisons of GLVQ and RSLVQ and ways to automatically determine σ based on given data.

3 Dissimilarity data

In recent years, data are becoming more and more complex in many application domains e.g. due to improved sensor technology or dedicated data formats. To account for this fact, data are often addressed by means of dedicated dissimilarity measures which account for the structural form of the data such as alignment techniques for bioinformatics sequences, dedicated functional norms for mass spectra, the compression distance for texts, etc. Prototype-based techniques such as GLVQ or RSLVQ are restricted to Euclidean vector spaces. Hence their suitability to deal with complex non-Euclidean data sets is highly limited. Prototype-based techniques such as neural gas have recently been extended towards more general data formats [12]. Here we extend GLVQ and RSLVQ to relational variants in a similar way by means of an implicit reference to a pseudo-Euclidean embedding of data.

We assume that data $\boldsymbol{x}^i$ are given as pairwise dissimilarities $d_{ij}=d(\boldsymbol{x}^i,\boldsymbol{x}^j)$. D refers to the corresponding dissimilarity matrix. Note that it is easily possible to transfer similarities to dissimilarities and vice versa, see [13]. We assume symmetry $d_{ij}=d_{ji}$ and we assume $d_{ii}=0$. However, we do not require that d refers to a Euclidean data space, i.e. D does not need to be embeddable in Euclidean space, nor does it need to fulfill the conditions of a metric.

As argued in [13, 12], every such set of data points can be embedded in a so-called pseudo-Euclidean vector space the dimensionality of which is limited by the number of given points. A pseudo-Euclidean vector space is a real-vector space equipped with the bilinear form $\langle\boldsymbol{x},\boldsymbol{y}\rangle_{p,q}=\boldsymbol{x}^t I_{p,q}\boldsymbol{y}$ where $I_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. The tuple (p,q) is also referred to as the signature of the space, and the value q determines in how far the standard Euclidean norm has to be corrected by negative eigenvalues to arrive at the given dissimilarity measure. The data set is Euclidean if and only if $q=0$. For a given matrix D, the corresponding pseudo-Euclidean embedding can be computed by means of an eigenvalue decomposition of the related Gram matrix, which is an $\mathcal{O}(m^3)$ operation. It yields explicit vectors $\boldsymbol{x}^i$ such that $d_{ij}=\langle\boldsymbol{x}^i-\boldsymbol{x}^j,\boldsymbol{x}^i-\boldsymbol{x}^j\rangle_{p,q}$ holds for every pair of data points.

Note that vector operations can be naturally transferred to pseudo-Euclidean space, i.e. we can define prototypes as linear combinations of data in this space.

Hence we can perform techniques such as GLVQ explicitly in pseudo-Euclidean space since it relies on vector operations only. One problem of this explicit transfer is given by the computational complexity of the initial embedding, on the one hand, and the fact that out-of-sample extensions to new data points characterized by pairwise dissimilarities are not immediate.

Because of this fact, we are interested in efficient techniques which implicitly refer to such embeddings only. As a side product, such algorithms are invariant to coordinate transforms in pseudo-Euclidean space, rather they depend on the pairwise dissimilarities only instead of the chosen embedding. The key assumption is to restrict prototype positions to linear combination of data points of the form

$$\boldsymbol{w}^j = \sum_i \alpha_{ji} \boldsymbol{x}^i \text{ with } \sum_i \alpha_{ji} = 1\,. \tag{6}$$

Since prototypes are located at representative points in the data space, it is a reasonable assumption to restrict prototypes to the affine subspace spanned by the given data points. In this case, dissimilarities can be computed implicitly by means of the formula

$$d(\boldsymbol{x}^i, \boldsymbol{w}^j) = [D \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j \tag{7}$$

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ refers to the vector of coefficients describing the prototype $\boldsymbol{w}^j$ implicitly, as shown in [12].

This observation constitutes the key to transfer GLVQ and RSLVQ to relational data without an explicit embedding in pseudo-Euclidean space. Prototype $\boldsymbol{w}^j$ is represented implicitly by means of the coefficient vectors α_j. Then, we can use the equivalent characterization of distances in the GLVQ and RSVLQ cost function leading to the costs of relational GLVQ (RGLVQ) and relational RSLVG (RSLVQ), respectively:

$$E_{\text{RGLVQ}} = \sum_i \Phi\left(\frac{[D\alpha^+]_i - \frac{1}{2} \cdot (\alpha^+)^t D \alpha^+ - [D\alpha^-]_i + \frac{1}{2} \cdot (\alpha^-)^t D \alpha^-}{[D\alpha^+]_i - \frac{1}{2} \cdot (\alpha^+)^t D \alpha^+ + [D\alpha^-]_i - \frac{1}{2} \cdot (\alpha^-)^t D \alpha^-}\right), \tag{8}$$

where as before the closest correct and wrong prototype are referred to, corresponding to the coefficients α^+ and α^-, respectively. A stochastic gradient descent leads to adaptation rules for the coefficients α^+ and α^- in relational GLVQ: component k of these vectors is adapted as

$$\Delta\alpha_k^{\pm} \sim \mp \Phi'(\mu(\boldsymbol{x}^i)) \cdot \mu^{\pm}(\boldsymbol{x}^i) \cdot \frac{\partial\left([D\alpha^{\pm}]_i - \frac{1}{2} \cdot (\alpha^{\pm})^t D \alpha^{\pm}\right)}{\partial \alpha_k^{\pm}} \tag{9}$$

where $\mu(\boldsymbol{x}^i)$, $\mu^+(\boldsymbol{x}^i)$, and $\mu^-(\boldsymbol{x}^i)$ are as above. The partial derivative yields

$$\frac{\partial\left([D\alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j\right)}{\partial \alpha_{jk}} = d_{ik} - \sum_l d_{lk} \alpha_{jl} \tag{10}$$

Similarly,

$$E_{\text{RRSLVQ}} = \sum_i \log \frac{\sum_{\alpha_j : c(\alpha_j) = c(\boldsymbol{x}^i)} p(\boldsymbol{x}^i | \alpha_j)/k}{\sum_{\alpha_j} p(\boldsymbol{x}^i | \alpha_j)/k} \tag{11}$$

where $p(\boldsymbol{x}^i|\alpha_j) = K \cdot \exp\left(-\left([D\alpha_j]_i - \frac{1}{2}\cdot\alpha_j^t D\alpha_j\right)/2\sigma^2\right)$. A stochastic gradient descent leads to the adaptation rule

$$\Delta\alpha_{jk} \sim -\frac{1}{2\sigma^2}\cdot\left(\frac{p(\boldsymbol{x}^i|\alpha_j)}{\sum_{j:c(\alpha_j)=c(\boldsymbol{x}^i)} p(\boldsymbol{x}^i|\alpha_j)} - \frac{p(\boldsymbol{x}^i|\alpha_j)}{\sum_j p(\boldsymbol{x}^i|\alpha_j)}\right)\cdot\frac{\partial\left([D\alpha_j]_i - \frac{1}{2}\alpha_j^t D\alpha_j\right)}{\partial\alpha_{jk}} \tag{12}$$

if $c(\boldsymbol{x}^i) = c(\alpha_j)$ and $\Delta\alpha_{jk} \sim \frac{1}{2\sigma^2}\cdot\frac{p(\boldsymbol{x}^i|\alpha_j)}{\sum_j p(\boldsymbol{x}^i|\alpha_j)}\cdot\frac{\partial\left([D\alpha_j]_i - \frac{1}{2}\alpha_j^t D\alpha_j\right)}{\partial\alpha_{jk}}$ if $c(\boldsymbol{x}^i) \neq c(\alpha_j)$. After every adaptation step, normalization takes place to guarantee $\sum_i \alpha_{ji} = 1$.

The prototypes are initialized as random vectors, i.e we initialize α_{ij} with small random values such that the sum is one. It is possible to take class information into account by setting all α_{ij} to zero which do not correspond to the class of the prototype. The prototype labels can then be determined based on their receptive fields before adapting the initial decision boundaries by means of supervised learning vector quantization.

An extension of the classification to new data is immediate based on an observation made in [12]: given a novel data point $\boldsymbol{x}$ characterized by its pairwise dissimilarities $D(\boldsymbol{x})$ to the data used for training, the dissimilarity of $\boldsymbol{x}$ to a prototype represented by α_j is $d(\boldsymbol{x}, \boldsymbol{w}^j) = D(\boldsymbol{x})^t \cdot \alpha_j - \frac{1}{2}\cdot\alpha_j^t D\alpha_j$.

Note that, for GLVQ, a kernelized version has been proposed in [11]. However, this refers to a kernel matrix only, i.e. it requires Euclidean similarities instead of general symmetric dissimilarities. In particular, it must be possible to embed data in a possibly high dimensional Euclidean feature space. Here we extended GLVQ and RSLVQ to relational data characterized by a general symmetric dissimilarities which might be induced by strictly non-Euclidean data.

4 Experiments

We evaluate the algorithms for several benchmark data sets where data are characterized by pairwise dissimilarities. On the one hand, we consider six data sets used also in [16]: Amazon47, Aural-Sonar, Face Recognition, Patrol, Protein and Voting. In additional we consider the Cat Cortex from [18], the Copenhagen Chromosomes data [17] and one own data set, the Vibrio data, which consists of 1,100 samples of vibrio bacteria populations characterized by mass spectra. The spectra contain approx. 42,000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The preprocessing of the Vibrio data is described in [20] and the underlying similarity measures in [21, 20]. The article [16] investigates the possibility to deal with similarity/dissimilarity data which is non-Euclidean with the SVM. Since the corresponding Gram matrix is not positive semidefinite, according preprocessing steps have to be done which make the SVM well defined. These steps can change the spectrum of the Gram matrix or they can treat the dissimilarity values as feature vectors which can be processed by means of a standard kernel.

Since some of these matrices correspond to similarities rather than dissimilarities, we use standard preprocessing as presented in [13]. For every data set, a number of prototypes which mirrors the number of classes was used, representing every class by only few prototypes relating to the choices as taken in [12],

	#Data Points	#Labels	**RGLVQ**	**RRSLVQ**	**best SVM**	**#Proto.**
Amazon47	204	47	0.81(0.01)	**0.83**(0.02)	0.82*	94
Aural Sonar	100	2	**0.88**(0.02)	0.85(0.02)	0.87*	10
Face Rec.	945	139	**0.96**(0.00)	**0.96**(0.00)	**0.96***	139
Patrol	241	8	0.84(0.01)	0.85(0.01)	**0.88***	24
Protein	213	4	0.92(0.02)	0.53(0.01)	**0.97***	20
Voting	435	2	**0.95**(0.01)	0.62(0.01)	**0.95***	20
Cat Cortex	65	5	0.93(0.01)	0.94(0.01)	**0.95**	12
Vibrio	4200	22	**1.00**(0.00)	0.94(0.08)	**1.00**	49
Chromosome	1100	49	0.93(0.00)	0.80(0.01)	**0.95**	63

Table 1. Results of prototype based classification in comparison to SVM for diverse dissimilarity data sets. The classification accuracy obtained in a repeated cross-validation is reported, the standard deviation is given in parenthesis. SVM results marked with * are taken from [16]. For Cat Cortex, Vibrio, Chromosome, the respective best SVM result is reported by using different preprocessing mechanisms clip, flip, shift, and similarities as features with linear and Gaussian kernel.

see Tab. 1. The evaluation of the results is done by means of the classification accuracy as evaluated on the test set in a ten fold repeated cross-validation (nine tenths of date set for training, one tenth for testing) with ten repeats. The results are reported in Tab. 1. In addition, we report the best results obtained by SVM after diverse preprocessing techniques [16].

Interestingly, in most cases, results which are comparable to the best SVM as reported in [16] can be found, whereby making preprocessing as done in [16] superfluous. Further, unlike for SVM which is based on support vectors in the data set, solutions are represented as typical prototypes.

5 Conclusions

We have presented an extension of prototype-based techniques to general possibly non-Euclidean data sets by means of an implicit embedding in pseudo-Euclidean data space and a corresponding extension of the cost function of GLVQ and RSLVQ to this setting. As a result, a very powerful learning algorithm can be derived which, in most cases, achieves results which are comparable to SVM but without the necessity of according preprocessing since relational LVQ can directly deal with possibly non-Euclidean data whereas SVM requires a positive semidefinite Gram matrix. Similar to SVM, relational LVQ has quadratic complexity due to its dependency on the full dissimilarity matrix. A speed-up to linear techniques e.g. by means of the Nyström approximation for dissimilarity data similar to [22] is the subject of ongoing research.

Acknowledgement Financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative and from the "German Science Foundation (DFG)" under grant number HA-2719/4-1 is gratefully acknowledged.

References

1. Martinetz, T.M., Berkovich S.G., Schulten, K.J.: 'Neural-gas' Network for Vector Quantization and Its Application to Time-series Prediction. IEEE Trans. on Neural Networks 4(4), 558–569 (1993)
2. Kohonen T.: Self-Organizing Maps. Springer-Verlag New York, Inc., 3rd edition (2001)
3. Bishop C., Svensen M., and Williams C.: The Generative Topographic Mapping. Neural Computation 10(1), 215-234 (1998)
4. Sato A. and Yamada K.: Generalized Learning Vector Quantization. Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference, pages 423–9, Cambridge, MA, USA, 1996. MIT Press.
5. Seo S. and Obermayer K.: Soft Learning Vector Quantization. Neural Computation 15(7), 1589–1604 (2003)
6. Hammer B. and Villmann T.: Generalized Relevance Learning Vector Quantization. Neural Networks 15(8-9), 1059–1068 (2002)
7. Schneider P., Biehl M., and Hammer B.: Adaptive Relevance Matrices in Learning Vector Quantization. Neural Computation 21(12), 3532–3561 (2009)
8. Denecke A., Wersing H., Steil J.J., and Koerner E.: Online Figure-Ground Segmentation with Adaptive Metrics in Generalized LVQ. Neurocomputing 72(7-9), 1470-1482 (2009)
9. Kietzmann T., Lange S. and Riedmiller M.: Incremental GRLVQ: Learning Relevant Features for 3D Object Recognition. Neurocomputing 71 (13-15), 2868-2879 (2008)
10. Alex N., Hasenfuss A., Hammer B.: Patch Clustering for Massive Data Sets. Neurocomputing 72(7-9), 1455-1469 (2009)
11. Qin A.K., Suganthan P.N.: A Novel Kernel Prototype-based Learning Algorithm. Proc. of ICPR'04, 621–624 (2004)
12. Hammer B. and Hasenfuss A.: Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9), 2229-2284 (2010)
13. E. Pekalska and R.P.W. Duin The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore, December 2005.
14. Schneider P., Biehl M., Hammer B.: Hyperparameter Learning in Probabilistic Prototype-based Models. Neurocomputing 73(7-9): 1117-1124 (2010)
15. Seo S. and Obermayer K.: Dynamic Hyperparameter Scaling Method for LVQ Algorithms. IJCNN:3196-3203 (2006)
16. Chen Y., Eric K.G., Maya R.G., Ali R.L.C.: Similarity-based Classification: Concepts and Algorithms, Journal of Machine Learning Research 10(Mar), 747–776 (2009)
17. Neuhaus M. and Bunke H.: Edit Distance Based Kernel functions for Structural Pattern Classification. Pattern Recognition 39(10): 1852–1863 (2006)
18. Haasdonk B. and Bahlmann C.: Learning with Distance Substitution Kernels, Pattern Recognition - Proc. of the 26th DAGM Symposium (2004)
19. Lundsteen C., Phillip J., and Granum E.: Quantitative Analysis of 6985 Digitized Trypsin g-banded Human Metaphase Chromosomes. Clinical Genetics 18: 355–370 (1980)
20. Maier T., Klebel S., Renner U., and Kostrzewa M.: Fast and Reliable maldi-tof ms–based Microorganism Identification. Nature Methods 3 (2006)
21. Barbuddhe S.B. , Maier T., Schwarz G., Kostrzewa M., Hof H., Domann E., Chakraborty T., and Hain T.: Rapid Identification and Typing of Listeria Species by Matrix-assisted Laser Desorption Ionization-time of Flight Mass Spectrometry. Applied and Environmental Microbiology 74(17), 5402–5407 (2008)
22. Gisbrecht A., Hammer B., Schleif F.-M. and Zhu X.: Accelerating Dissimilarity Clustering for Biomedical Data Analysis. Proceedings of SSCI (2011)