Local Rejection Strategies for Learning Vector Quantization

Lydia Fischer[1,2], Barbara Hammer[2], and Heiko Wersing[1]

1 – HONDA Research Institute Europe GmbH,
Carl-Legien-Str. 30, 63065 Offenbach, Germany
2 – Bielefeld University, Universitätsstr. 25, 33615 Bielefeld, Germany

Abstract. Classification with rejection is well understood for classifiers which provide explicit class probabilities. The situation is more complicated for popular deterministic classifiers such as learning vector quantisation schemes: albeit reject options using simple distance-based geometric measures were proposed [4], their local scaling behaviour is unclear for complex problems. Here, we propose a local threshold selection strategy which automatically adjusts suitable threshold values for reject options in prototype-based classifiers from given data. We compare this local threshold strategy to a global choice on artificial and benchmark data sets; we show that local thresholds enhance the classification results in comparison to global ones, and they better approximate optimal Bayesian rejection in cases where the latter is available.

Keywords: prototype-based reject, classification, local thresholds

1 Motivation

Learning vector quantisation (LVQ) [9] constitutes a powerful and efficient method for multi-class classification tasks which, due to its representation of models in terms of prototypes, is particularly suited for on-line scenarios or lifelong learning [8]. While classical LVQ models have been introduced on heuristic grounds, modern variants are based on cost-function models like generalized LVQ (GLVQ) [12], or robust soft LVQ (RSLVQ) [15] with guarantees on generalization performance and learning dynamics [2, 13]. One particular success story links LVQ classifiers to simultaneous metric learners which enrich the classifier with interpretable feature weighting terms or a direct classifier visualisation [13, 14]. Still, LVQ classifiers face the problem that real world data do not necessarily allow an unambiguous classification: overlap in the data, outliers, noise, or similar effects can be observed frequently where wrong classifications are unavoidable. A wrong classification can be more costly than postponing a decision and gathering new evidence like in medical diagnostics. Mathematically, such settings can be modelled by introducing a reject option for a classifier: instead of a decision, rejecting is possible for cases with low certainty. This setting has formally been analysed by Chow [3], deriving an optimum decision rule depending on the costs of a reject in comparison to a wrong classification. While this early approach

addresses the setting that reliable class probabilities are available, the approach [7] extends this optimum decision by plug-in rules which rely on empirical estimations of class probabilities only, providing guarantees of the quality in case of a reliable estimator and a suitably low density of data at the reject thresholds.

Still, these schemes rely on the assumption that conditional class probabilities or reliable estimations thereof are available. Albeit there exist few approaches which model LVQ classifiers by means of class probabilities such as RSLVQ [15], it is unclear whether such discriminative models converge to the correct underlying class distributions, and most popular LVQ schemes are based on deterministic decision models only instead of a reference to class probabilities [5, 12, 17]. Recently [4, 5] it has been analysed if alternative real-valued outputs correlated to the deterministic classification model can take the role of a certainty value for a reject option: examples include the distance of a data vector to the closest decision boundary, prototype. Interestingly, using simple thresholds, these measures offer classification schemes with a reject option with the quality close to optimum Bayesian decisions in simple model cases [4, 5].

One drawback of these techniques is that they are based on one global threshold for a reject option, thus relying on the assumption that the considered measures scale independently of the data region. This is usually not the case: measures such as distances, unlike a certainty, are not normalized and scaling varies within a given data set. Hence reject options with a global threshold are restricted to simple models only. In cases where the classes or parts of classes have not the same compactness or where the scaling of the values is unclear, this approach is limited, and it can be an advantage to use local thresholds [6, 17].

For prototype-based classification there exists an intuitive strategy to define regions for the local thresholds: Use the Voronoi-tessellation of the input space provided by the prototypes. Here we present a greedy optimization method to adaptively determine local thresholds for an LVQ classifier based on given data. We compare the resulting local rejection strategies to the global counterparts as proposed in [4, 5] using several benchmarks and one artificial data set. We show that local thresholds outperform their global counterpart, approximating the optimal reject option of Chow [3] in cases where the latter is available.

2 Learning Vector Quantization

Assume N training samples $\boldsymbol{x} \in \mathbb{R}^n$ with attached class labels $y \in \{1, \ldots, L\}$, if L classes are considered. An LVQ classifier is represented by a set of prototypes $W = \{\boldsymbol{w}_i \in \mathbb{R}^n\}_{i=1}^k$ which are equipped with class labels $c(\boldsymbol{w}) \in \{1, \ldots, L\}$. Classification takes place by a winner takes all scheme: A data vector $\boldsymbol{x}$ is mapped to the class label $c(\boldsymbol{x}) = c(\boldsymbol{w}_i)$ of the closest prototype $\boldsymbol{w}_i$ according to a distance measure d. Here, we use the squared Euclidean distance $d(\boldsymbol{x}, \boldsymbol{w}) = \|\boldsymbol{x} - \boldsymbol{w}\|^2$.

For a GLVQ model [12], the position of the prototypes W are determined by a stochastic gradient decent on the following cost function:

$$E = \sum_i \Phi((d^+(\boldsymbol{x}_i) - d^-(\boldsymbol{x}_i))/(d^+(\boldsymbol{x}_i) + d^-(\boldsymbol{x}_i))) \; . \tag{1}$$

$\Phi(\cdot)$ is a monotonic increasing function, e. g. the identity. The distances of a data vector $\boldsymbol{x}$ to the closest prototypes with the same/different label are denoted as d^+/d^-. Replacing the distance measure by a general quadratic form $(\boldsymbol{x}-\boldsymbol{w}_i)^T\Lambda(\boldsymbol{x}-\boldsymbol{w}_i)$ with positive semi-definite matrix Λ results in a generalization of GLVQ, generalised matrix LVQ (GMLVQ) [13] whereby matrix parameters can be adapted coevally to the prototypes according to the given data.

The cost E (1) correlates with the classification error because a data vector is classified correctly iff the nominator of (1) is below zero. The nominator can be connected to the hypothesis margin of the classifier which relates to its generalisation ability [13]. Note that the argument of $\Phi(\cdot)$ ranges in $[-1,1]$. A value near -1 indicates a high certainty of the classification because $d^+ \ll d^-$.

3 Reject Option

The aim of a reject option is to identify outliers and data vectors with low certainty of classification [17]. A *rejection measure* refers to a real-valued function $r : \mathbb{R}^n \to \mathbb{R}^+, \boldsymbol{x} \mapsto r(\boldsymbol{x})$ indicating the certainty of the classification. We assume that high values indicate a more certain classification. A vector is rejected iff $r(\boldsymbol{x}) < \theta$, where $\theta \geq 0$ is a threshold. We refer to such strategies as *global* rejection strategies if one global threshold θ is chosen for all inputs $\boldsymbol{x} \in \mathbb{R}^n$.

A *local threshold strategy* where the input space is partitioned into single regions enables a finer control of rejection [17]. Following the suggestion in [17], we use the natural decomposition of the input space into the Voronoi-cells

$$V_j = \{\boldsymbol{x}_i | d(\boldsymbol{x}_i, \boldsymbol{w}_j) \leq d(\boldsymbol{x}_i, \boldsymbol{w}_k), \forall\, k \neq j\}\ ; \tag{2}$$

as induced by the prototypes of an LVQ classifier. For a local threshold strategy based on Voronoi-cells (2) a separate threshold $\theta_j \geq 0$ is chosen for every cell, and the reject strategy is given by a threshold vector of the dimension $|W|$ equal to the number of V_j. A vector $\boldsymbol{x}$ is rejected iff $r(\boldsymbol{x}) < \theta_j$ for $\boldsymbol{x} \in V_j$. In the case of one prototype per class, local thresholds realise a class-wise reject option.

After defining local and global threshold strategies, we need to specify the rejection measure and a method for finding suitable local θ_j.

Choice of the rejection measure: In our experiments we use the relative similarity (RelSim) as proposed in [4] as rejection measure:

$$\mathrm{RelSim}(\boldsymbol{x}) = \frac{d^-(\boldsymbol{x}) - d^+(\boldsymbol{x})}{d^-(\boldsymbol{x}) + d^+(\boldsymbol{x})}\ . \tag{3}$$

This measure can be applied for new data vectors after defining their class label with the winner takes all scheme. RelSim is inspired by the cost function of GLVQ (1) [12]. Its values are normalised to $[0,1]$ and 1 indicates a high certainty of the classification with respect to the trained prototypes. It can efficiently be calculated and it combines a reject option for outliers and ambiguous data vectors due to its design (Fig. 1, left). As baseline for an artificial data set the maximum of the class probabilities $\max_y p(y|\boldsymbol{x})$ of the Bayes classifier with known densities [3] is used for rejection (Fig. 1, right).

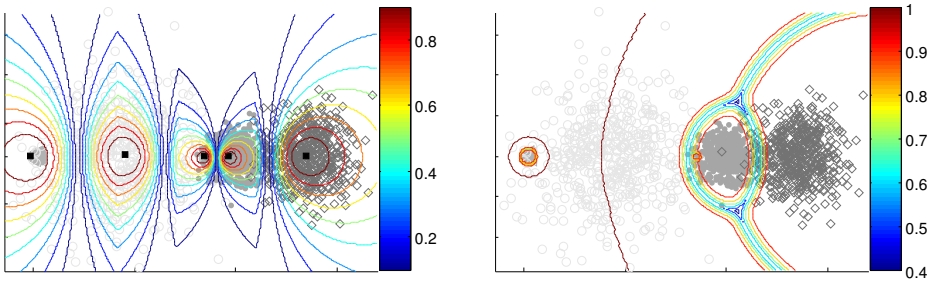

Fig. 1. Level curves of RelSim (3) (left) and Bayes (right) for an artificial five-class problem. The black squares are prototypes. A critical region for a global threshold is between the third and fourth cluster from left. The third cluster needs a high threshold because the data vectors are very compact. Applying the same threshold for the fourth cluster would lead to rejection of the most vectors in this cluster which is undesired.

Adaptation of local thresholds: The baseline case with no rejection is given when all local thresholds are set to $\theta_j = 0$. We propose the following greedy strategy: For rejection increase those θ_j, where most wrongly classified vectors can be rejected while accepting a constant number of rejected correct vectors. We associate the rejection of a correct vector with a constant cost of 1. Starting from 0, the global cost is increased in steps of 1, and the θ_j are adapted accordingly.

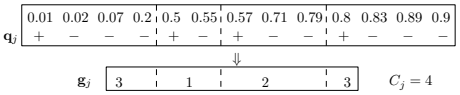

Fig. 2. V_j with 13 vectors. First row implies the sorted $r(\boldsymbol{x}_i)$ values, the second codes if a vector is correct (+)/wrong (−) classified. The third row implies the coding of the gain $\boldsymbol{g}_j$ of rejection steps during the algorithm.

We assume that the vectors $\boldsymbol{x}_i \in V_j$ are sorted according to their (RelSim) certainty value $r(\boldsymbol{x}_i)$. Let $\mathbf{q}_j$ denote the vector of classification results for the vectors in V_j with $q_j(\boldsymbol{x}_i) = 1\,(+)$ for correct and $q_j(\boldsymbol{x}_i) = -1\,(-)$ for wrong classification (cp. Fig. 2). Voronoi-cells V_j without errors can be neglected (i. e. $\theta_j = 0$), because they cannot contribute reasonably to rejection. Let $C_j = \sum_{i|q_j(\boldsymbol{x}_i)=1} 1$ be the number of correct vectors in V_j and let $E_j = \sum_{i|q_j(\boldsymbol{x}_i)=-1} 1$ be the number of errors in V_j. The aim of the algorithm 3.1 is to return an accuracy reject curve which consists of two vectors $\mathbf{t}_c$ and $\mathbf{t}_a$. For an iteration step s a single point $(t_c(s), t_a(s))$ reports the relative size $t_c(s)$ of the set of accepted vectors X_θ in comparison to $|X|$ and the accuracy on X_θ which is $t_a(s)$. The original model without rejection initialises these vectors with $t_c(0) = 1$ and $t_a(0) = \sum_j C_j/|X|$. Respectively we define E_R and C_R as counter for rejected errors and rejected correct classified vectors. If $\boldsymbol{x}_i$ with $\max_{\boldsymbol{x}_i \in V_j} r(\boldsymbol{x}_i)$ is correct classified then $\mathbf{a}$ denotes the indexes of the correct classified vectors in V_j. Otherwise the first C_j entries of $\mathbf{a}$ contains indexes of the correct classified vectors and the last entry is given with $|V_j| + 1$ ($\mathbf{a} = (1, 5, 7, 10, 14)$ for example in Fig. 2). If $a_j(1) > 1$ there exist errors in V_j which can be rejected with zero cost. We code the wrongly classified vectors for

constant costs in the gain vector $\boldsymbol{g}_j$ and the accumulated gains $\hat{\boldsymbol{g}}_j$ as follows:

$$g_j(k) = a_j(k+1) - a_j(k) - 1, \quad \hat{g}_j(k) = \sum_{l=1}^{k} g_j(l), \quad k = 1, \ldots, C_j \tag{4}$$

$g_j(k)$ is the local gain for rejecting the next correct vector (cf. Fig. 2) and $\hat{g}_j(k)$ states the accumulated gains for costs of k in V_j. An example for $\boldsymbol{g}_j$ and $\hat{\boldsymbol{g}}_j$ with three Voronoi-cells is given in table 1.

Algorithm 3.1: GREEDY OPTIMIZATON$(\mathbf{a}_j, \boldsymbol{g}_j, \hat{\boldsymbol{g}}_j \forall j)$

$C_R := 1;\ E_R := \sum_j (a_j(1) - 1)$ //errors whose rejection is gratis
$t_c(1) := 1 - E_R/|X|;\ t_a(1) := \sum_j C_j/(|X| - E_R)$
$s := 2;\ k_j := 0\ \forall\, j$
while $E_R \neq \sum_j E_j$
do
- $m := \text{argmax}_j\{g_j(k_j + 1)\}$ //index: most improvement locally
- $\hat{m} := \text{argmax}_j\{\hat{g}_j(C_R)\}$ //index: most improvement globally
- **if** $\max_j\{\hat{g}_j(C_R)\} > \max_j\{g_j(k_j + 1)\}$
  - **then**
    - $k_j := 0,\ \forall\, j;\ k_{\hat{m}} := C_R$ //discard whole solution
    - $C_R := C_R + 1;\ E_R := \hat{g}_{\hat{m}}$
  - **else**
    - **if** $\exists!\, \max_j\{g_j(k_j + 1)$
      - **then**
        - $k_m := k_m + 1;\ C_R := C_R + 1$
        - $E_R := E_R + g_m(k_m)$
      - **else**
        - $o := 1$ //allows increasing > 1 for C_R
        - **while** $\neg(\exists!\, \max_j\{g_j(k_j + 1))$
          - **do** $\{o := o + 1;\ m := \text{argmax}_j\{g_j(k_j + o)\}$
        - $C_R := C_R + o;\ k_m := k_m + o$
        - $E_R := E_R + \sum_{l=1}^{o} g_m(k_m + l)$
- $\mathbf{t}_c(s) := 1 - (C_R + E_R)/|X|$
- $\mathbf{t}_a(s) := (\sum_j C_j - C_R)/(|X| - (C_R + E_R))$
- $s := s + 1$

return $(\mathbf{t}_c, \mathbf{t}_a)$

Table 1. Examples for $\boldsymbol{g}_j$, $\hat{\boldsymbol{g}}_j$ for three V_j.

# (+)	1	2	3	4	1	2	3	4
	$\boldsymbol{g}_j$				$\hat{\boldsymbol{g}}_j$			
V_1	3	1	2	3	3	4	6	9
V_2	2	1	3	-	2	3	6	-
V_3	1	1	8	10	1	2	10	20

The greedy algorithm for the local threshold adaptation (Alg. 3.1) operates mainly on $\boldsymbol{g}_j$ and $\hat{\boldsymbol{g}}_j$. Using the example (Tab. 1) one obtains the steps in table 2. First the algorithm checks if errors can be rejected with no cost, then checks for the cell with highest gain. E. g. the gains $g_1(1) = 3$, $g_2(1) = 2$, $g_3(1)$=1 are possible and the algorithm picks $g_1(1)$=3 which results in $k_1 = 1$ $(\hat{g}_j(1) = g_j(1), \forall\, j)$. Then $g_1(2) = 1$, $g_2(1) = 2$, $g_3(1) = 1$ are possible and it picks $g_2(1) = 2$, raising θ_j in this cell because $\max_j\{\hat{g}_j(C_R)\}$ is lower. Now we have 2 correct data vectors and 5 errors rejected. In the next step, $g_1(2)$=1, $g_2(2)$=1, $g_3(1)$=1 are possible and we choose a gain of 1 and 3 correct data vectors would be rejected. The overall gain of this solution is $3 + 2 + 1 = 6$.

Checking table. 1 we see for 3 correct data vectors the gains $\hat{g}_1(3)$=6, $\hat{g}_2(3)$=6, $\hat{g}_3(3)$=10. Then it is better to discard the previous solution and reject only in V_3 because of a gain of 10 instead of 6. An exception rule comes into operation when there are more than one optima in the gains and discarding the whole solution is no option. This means there is no single maximum in $\boldsymbol{g}_j$ and $\hat{\boldsymbol{g}}$. In this case the algorithm increases the costs till one local optimum remains.

4 Experiments

We evaluate the benefit of a local threshold strategy as compared to the global counterpart in a variety of experiments. Thereby, we train all models using GLVQ and GMLVQ with one prototype per class. Evaluations are obtained as results of a repeated 10-fold cross-validation with ten repetitions. For one artificial data set, an optimum Bayesian reject option based on the quantity $\max_y p(y|\boldsymbol{x})$ can be evaluated as proposed in [3]. We will use this ground truth as baseline where it is possible. The data sets which we will consider include the following:

Table 2. Iterations of the algorithm. Shows how the costs are split to V_j.

costs	1	2	3	4	5	6	8	9	10
$V_1 : k_1$	1	1	0	0	1	1	1	2	3
$V_2 : k_2$	0	1	0	0	0	1	3	3	3
$V_3 : k_3$	0	0	3	4	4	4	4	4	4

- *Pearl necklace*: This data set consists of five artificially generated Gaussian clusters in two dimensions with overlap. (parameters: $\mu_{y_i} = 3\ \forall i, \mu_x = (2, 44, 85, 100, 136), \sigma_x = (1, 20, 0.5, 7, 11), \sigma_x = \sigma_y$)
- *Image Segmentation*: The image segmentation data set consists of 2310 data vectors which contain 19 real-valued image descriptors. The data vectors represent small patches from outdoor images with 7 different classes with equal distribution such as grass, cement, etc. [1].
- *Tecator data*: The Tecator data set [16] consists of 215 spectra of meat probes. The 100 spectral bands ranging from 850 nm to 1050 nm. The task is to predict the fat content (high/low) of the probes, which is turned into a two class classification problem. Both classes have the same size.
- *Haberman*: The Haberman survival data set includes 306 instances from two classes indicating being alive for more than 5 years after breast cancer surgery [1]. One instance is represented by three attributes linked to the age, the year, and the number of positive axillary nodes detected.
- *Coil*: The Columbia Object Image Database Library (COIL-20) contains gray scaled images of twenty objects [11]. Each object is rotated in 5° steps, resulting in 72 images per object. The data set contains 1440 vectors with a dimension of 16384. We reduce the dimensionality to 30 with PCA.

Figure 3 displays the classification accuracy obtained by local and global reject options for these data sets and the GLVQ and GMLVQ classifier on the set of classified vectors versus the percentage of vectors which are not rejected for a given threshold [10]. More precisely, assume X_θ denotes the set of data vectors which are not rejected using the respective threshold strategy. Then the graphs

display the relative size $|X_\theta|/|X|$ against the classification accuracy on X_θ. The local thresholds are determined with algorithm 3.1 on the train sets and then applied on the test sets. Figure 3 shows that local thresholds are slightly better than a global threshold in almost all cases. The local thresholds seem to be more efficient for GLVQ than for GMLVQ (Image Segmentation, Coil). The biggest improvement of the local strategy can be seen for the pearl necklace data set which was designed to show its advantage. Because of the huge differences in the standard deviations of the single classes/clusters one global threshold would be very ineffective. For low standard deviations a higher threshold than for a cluster with a huge standard deviation is needed. In most cases it is obvious that one needs a very high threshold for the global strategy to reject all errors. For the local strategy all errors can be rejected without rejecting all correctly classified data vectors in some cases (both models GLVQ and GMLVQ: Pearl necklace, Image Segmentation).

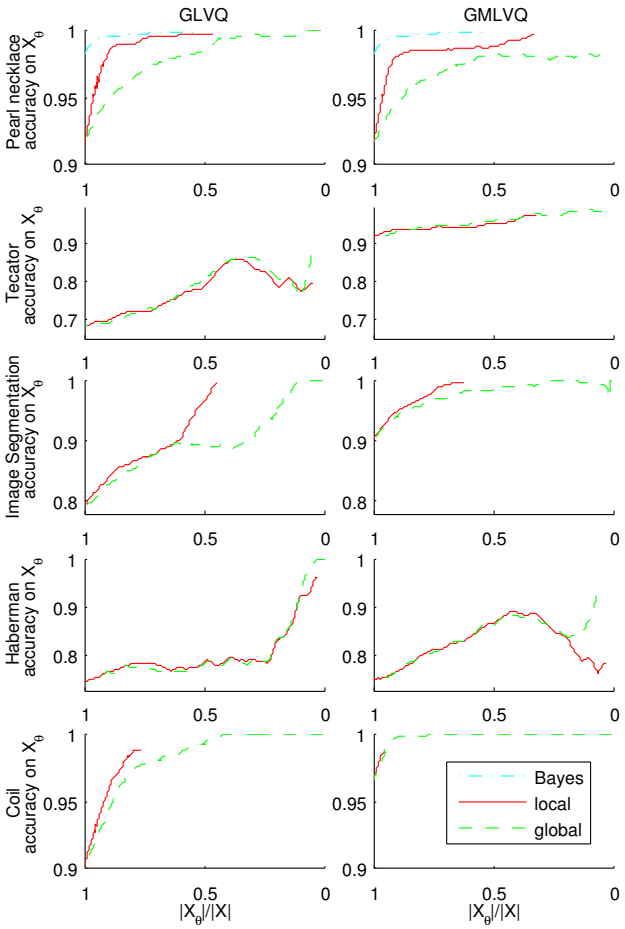

Fig. 3. Results of the global and local reject option with RelSim when applied to G(M)LVQ models trained for different data sets (test sets). We report accuracy reject curves [10]. The averaged curve is plotted, where at least 80 % of the single runs deliver a value.

5 Conclusion

We analysed the performance of a global and a local threshold strategy of a reject option. The results of artificial and benchmark data sets show that the local strategy delivers better accuracy values than the global counterpart in most cases. We showed a way of evaluating the local threshold strategy for different

rejection rates obtaining also the local thresholds. Applying a local threshold strategy costs a bit more than a global one but it has the advantage that one can fit the local thresholds to the data. This improves the accuracy and enhances the classification model especially for simple models like GLVQ.

Acknowledgments. The authors thank Stephan Hasler for very helpful debates on the threshold algorithm. BH gratefully acknowledges funding by the CITEC center of excellence. LF acknowledges funding by the CoR-Lab Research Institute for Cognition and Robotics and support from Honda Research Institute Europe.

References

1. K. Bache and M. Lichman. UCI machine learning repository, 2013.
2. M. Biehl, A. Ghosh, and B. Hammer. Dynamics and generalization ability of LVQ algorithms. *The Journal of Machine Learning Research*, 8:323–360, 2007.
3. C. K. Chow. On Optimum Recognition Error and Reject Tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.
4. L. Fischer, B. Hammer, and H. Wersing. Rejection Strategies for Learning Vector Quantization. In *Proc. European Symposium on Artificial Neural Networks (ESANN), Bruges*, pages 41–46, 2014.
5. L. Fischer, D. Nebel, T. Villmann, B. Hammer, and H. Wersing. Rejection Strategies for Learning Vector Quantization – a Comparison of Probabilistic and Deterministic Approaches. In *Proc. Workshop on Self-Organizing Maps (WSOM)*, 2014 accepted.
6. G. Fumera, F. Roli, and G. Giacinto. Reject option with multiple thresholds. *Pattern Recognition*, 33(12):2099–2101, Dec. 2000.
7. R. Herbei and M. H. Wegkamp. Classification with reject option. *Canadian Journal of Statistics*, 34(4):709–721, 2006.
8. S. Kirstein, H. Wersing, H.-M. Gross, and E. Körner. A Life-Long Learning Vector Quantization Approach for Interactive Learning of Multiple Categories. *Neural Networks*, 28:90–105, 2012.
9. T. Kohonen. *Self-Organization and Associative Memory*. Springer Series in Information Sciences, Springer-Verlag, third edition, 1989.
10. M. S. A. Nadeem, J.-D. Zucker, and B. Hanczar. Accuracy-Rejection Curves (ARCs) for Comparing Classification Methods with a Reject Option. In *International Workshop on Machine Learning in Systems Biology*, pages 65–81, 2010.
11. S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). *Technical Report CUCS-005-96*, February 1996.
12. A. Sato and K. Yamada. Generalized Learning Vector Quantization. In *Advances in Neural Information Processing Systems*, volume 7, pages 423–429, 1995.
13. P. Schneider, M. Biehl, and B. Hammer. Adaptive Relevance Matrices in Learning Vector Quantization. *Neural Computation*, 21(12):3532–3561, 2009.
14. P. Schneider, K. Bunte, H. Stiekema, B. Hammer, T. Villmann, and M. Biehl. Regularization in matrix relevance learning. *IEEE Transactions on Neural Networks*, 21(5):831–840, 2010.
15. S. Seo and K. Obermayer. Soft Learning Lector Quantization. *Neural Computation*, 15(7):1589–1604, Jul 2003.
16. H. H. Thodberg. Tecator data set, contained in StatLib Datasets Archive, 1995.
17. A. Vailaya and A. K. Jain. Reject Option for VQ-Based Bayesian Classification. In *International Conference on Pattern Recognition (ICPR)*, pages 2048–2051, 2000.