

Incremental Figure-Ground Segmentation using localized adaptive metrics in LVQ

Alexander Denecke^{1,2}, Heiko Wersing², Jochen J. Steil¹, and Edgar Körner²

¹ Bielefeld University - CoR-Lab, P.O.-Box 10 01 31, D-33501 Bielefeld, Germany
adenecke@cor-lab.uni-bielefeld.de

<http://www.cor-lab.de>

² Honda Research Institute Europe, Carl-Legien-Str. 30, D-63073 Offenbach/Main, Germany

Abstract. Vector quantization methods are confronted with a model selection problem, namely the number of prototypical feature representatives to model each class. In this paper we present an incremental learning scheme in the context of figure-ground segmentation. In presence of local adaptive metrics and supervised noisy information we use a parallel evaluation scheme combined with a local utility function to organize a learning vector quantization (LVQ) network with an adaptive number of prototypes and verify the capabilities on a real world figure-ground segmentation task.

1 Introduction

The appropriate choice of the number of model neurons is a principle problem in vector quantization networks. In particular incremental learning offers a solution to adjust the amount of resources needed versus classification performance to find a tradeoff between representation quality and the avoidance of over-fitting. Vector quantization methods provide a simple algorithmic yet powerful framework with applications, for example in image processing [1] or life-long learning [2,3]. We investigate such methods in the context of online figure-ground segmentation where homogenous image regions are represented by single feature representatives. Ideally the dimensionality of the network should represent the meaningful entities in the data. As this problem is ill-posed (subjective), several researchers have addressed this problem with heuristics in supervised or unsupervised settings. One main criterion used for unsupervised setups is the distance of the features to their representatives, namely the quantization error. The criterion in Growing Neural Gas [4] (and similar for the Growing Cell structures [2]) aims at a minimization of the quantization error and introduces new prototypes where the quantization error is large, guaranteeing that the introduction of new prototypes reduces this error. Supervised LVQ primarily aims at the minimization of the classification error which offers another source of information. For example Kirstein et al. [3] propose a heuristics to insert new prototypes at the decision boundary using the misclassified data points together with a distance criterion to determine the location for new prototypes.

In this contribution we investigate Generalized Learning Vector Quantization (GLVQ [5]) with adaptive metrics and propose a framework for incremental and online figure-ground segmentation which faces two problems. Firstly the local adaptive metrics complicates distance-based criteria to place new prototypes, where we use the confidence of the classification instead. Secondly the method has to cope with noisy supervised information, that is, the labels to adapt the networks are not fully confident. In particular we address the second problem by using a parallel evaluation method on the basis of a local utility function, which does not rely on global error optimization. After a short problem statement, we describe the overall method to allow incremental learning in the presence of non-confident supervised information together with the criteria to introduce and remove prototypes from the network. Finally we evaluate the method on a real world segmentation task and compare to previous results.

2 Method

2.1 Scenario

Our proposed method addresses the problem of online figure-ground segmentation for object learning and recognition. The application scenario consists of a human presenter showing objects to a pan-tilt stereo camera system, which is controlled by an attention system [6]. Using the concept of peripersonal space, the depth estimation of the region in front of the system is analyzed with a blob-detection within a specified depth interval (50cm-80cm). The most salient object in front of the system is continuously tracked and centered in view by setting the gaze direction. Additionally a square region of interest (ROI) is defined based on a distance estimate of the tracked blob and normalized to a size of $\mathcal{I} \times \mathcal{I}$ pixels, where we use $\mathcal{I} = 144$. Both methods assure an approximate invariance to position and size for the incoming stream of images showing the object in front of cluttered background. A first cue for parts of the scene that belong to the object can be derived from depth estimation which we call *object hypothesis*. Because extracting 3D information from 2D images in general is an ill-posed problem, the resulting hypothesis is characterized by a partially inconsistent overlap with the outline/region of the object. That is, some regions of the background are indicated as foreground and vice versa. Since the learning and recognition can be improved as the quality of the figure-ground segmentation is enhanced, we follow the concept of hypothesis refinement to derive the significant object parts (according to the underlying image features) from this initial guess. In [1] we investigated methods for object segmentation that use prototypical feature representatives to model figure and ground. In particular, we used binarized depth hypotheses as a supervised label for the image features to train a classifier for figure and ground with GLVQ. In our previous setup an empirically predefined number of prototypes were used, while the model selection problem was left open.

2.2 Generalized Learning Vector Quantization

From the camera system the following data is available for each frame. A stack of $M = 5$ feature maps $\mathcal{F} := \{F_i^{x,y} | i = 1..M\}$ corresponding to the RGB color and position information of the pixels forms the dataset $\mathcal{D} := \{\xi | \xi^{x,y} = (F_1^{x,y} .. F_M^{x,y})^T, 1 \leq x, y \leq \mathcal{I}\}$, where every pixel defines a feature vector. To take advantage from the temporal character of the data the features of $T = 2$ frames are combined to one dataset. Switching to a new frame effectively replaces 50% (or less if T is increased) of the data from one to another adaptation step. Additionally to the data the hypothesis \mathcal{H} is available indicating which pixels belong to foreground $\mathcal{H}^{x,y} = 1$ or background $\mathcal{H}^{x,y} = 0$ which is used as label $c(\xi^{x,y}) := \mathcal{H}^{x,y}$ for the image features. Assume that \mathcal{H} is partially wrong (i.e. only a small portion of the data is wrongly labeled), the goal is to derive a classifier $\mathcal{A}_{\mathcal{F}, \mathcal{H}}^{x,y}(\xi^{x,y})$ for the pixel features that generalizes to the relevant foreground (object) features.

The method of GLVQ is defined by a network of N class-specific prototypical feature representatives $\mathcal{P} := \{\mathbf{w}_p \in \mathbb{R}^M | p = 1..N\}$. For figure-ground segmentation a two class setup is used where $c(\mathbf{w}_p) \in \{0, 1\}$ encodes the user assigned class-membership of every prototype. The goal of the learning dynamics is to find the representatives in feature space to represent the data by minimizing the classification error defined by the functional $E[\mathcal{D}, \mathcal{P}] = \sum_{\xi^{x,y} \in \mathcal{D}} \sigma(\mu(d))$ with $\sigma(x) = \frac{1}{1+e^{-x}}$, $\mu(d) = \frac{d_J - d_K}{d_J + d_K}$. Here the variables $d_J = d(\xi^{x,y}, \mathbf{w}_J)$ and $d_K = d(\xi^{x,y}, \mathbf{w}_K)$ represent the distance between $\xi^{x,y}$ and the most similar prototype \mathbf{w}_J from the correct class with $\mathcal{H}^{x,y} = c(\mathbf{w}_J)$ and the distance to the most similar prototype \mathbf{w}_K from an incorrect class. Since similarity-based clustering and classification crucially depends on the underlying metrics, recently several adaptive metrics were proposed [7]. In the most general case the similarity metrics is extended towards a Mahalanobis metrics $d(\xi, \mathbf{w}_p) = (\xi - \mathbf{w}_p)^T A_p (\xi - \mathbf{w}_p)$, where the distance computation of the features to the representatives is extended towards a prototype specific $M \times M$ matrix A_p of relevance factors (Localized Generalized Matrix LVQ, LGMLVQ). In general, using the kernelized distance computation introduces non-linear decision boundaries. As described in Cramer et al. [8] this allows for a reduced number of prototypes while achieving a comparable performance to standard LVQ with multiple prototypes.

The prototypes \mathbf{w}_J and \mathbf{w}_K as well as the corresponding relevance factors A_J and A_K are optimized by means of gradient descent according to E on $10000 \cdot T$ randomly chosen pairs $(\xi^{x,y}, \mathcal{H}^{x,y})$, which is described in more detail in [1]. Since A_p has to be positive semi-definite to yield a valid metrics, i.e. $d(\xi, \mathbf{w}_p) = (\xi - \mathbf{w}_p)^T \Omega_p \Omega_p^T (\xi - \mathbf{w}_p) = (\Omega_p^T (\xi - \mathbf{w}_p))^2 \geq 0$, this is assured by adapting Ω_p , where $A_p = \Omega_p \Omega_p^T$. Additionally, the diagonal elements are normalize by $\sum_{i=1}^M A_{i,i} = 1$. In general the prototypes are kept from one to the consecutive frame and adapted to the new data.

To segment an image on the basis of such a network, it is partitioned into N segments (binary maps) $V_p \in \{0, 1\}$ by assigning all feature vectors $\xi^{x,y}$ (i.e. pixels of a particular frame) independently to the prototype \mathbf{w}_p with the smallest distance $d(\xi^{x,y}, \mathbf{w}_p)$. Using a prototype-based representation, the final

segmentation \mathcal{A} (binary map) is combined by choosing the binary maps from the prototypes assigned to the foreground $\mathcal{A} = \sum_p^N c(\mathbf{w}_p)V_p$. For object learning and recognition now \mathcal{A} is used instead of \mathcal{H} .

2.3 Incremental Framework

We showed [1] that this method is robust in the presence of the noisy \mathcal{H} and the increased model complexity using the adaptive metrics yields an improved segmentation quality in absence of over-fitting effects when an appropriate network dimensionality is chosen. Therefore the number of prototypes is an important parameter which determines the performance of the network with respect to runtime and generalization capability. Our main goal is here to adapt the number of prototypes during online processing of the data to use as many prototypes as necessary for the segmentation.

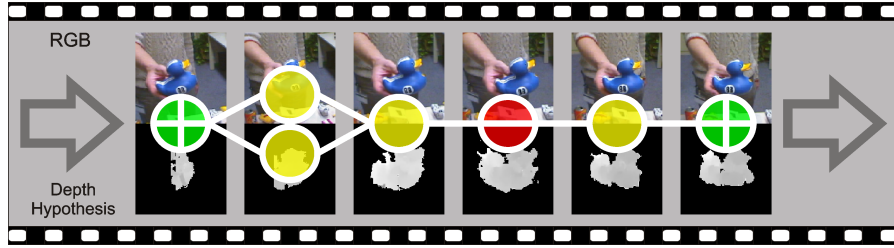


Fig. 1: The general algorithm to adapt the size of the network follows three main parts. Standard adaptation of a network using the LGMLVQ update rules (yellow circles), consisting of step 1 to 3 in Sec. 2.3. The green circles (plus) indicate an additional step to add a new prototype. This step yields two networks which are evaluated in parallel on the consecutive frame. Finally the red circle (minus) indicate an additional contraction step, where one of the prototypes (if appropriate) is removed.

Incremental Online Processing The proposed method consists of three parts, a standard adaptation step, one method to add new prototypes and a local criterion to remove prototypes from the network. To stabilize the incremental learning of the network in presence of the noisy supervised information, we use the temporal aspect of the data for a sequential processing together with a parallel evaluation scheme. That is, to avoid the adaptation to the hypothesis on a particular frame, adding and removing prototypes are applied in a consecutive manner where on a single frame only one prototype is added or removed. Additionally due to the risk of disturbing the network with such operations while online segmenting the image, we use a parallel evaluation scheme Fig. 1. The prototypes are added to a second network, which is an exact copy of the first one. After the evaluation a decision is applied whether the original network or the modified network is kept for the following frame.

Controlling the Network Size Incremental learning in prototype-based networks needs a mechanism to control the growing process to determine an appropriate number of prototypes. A widely used possibility is a global quality assessment. This information can be used to select the best performing set of prototypes after the network grew until a predefined maximum number of prototypes was reached [9], or to stop if the change in a quality measure does not significantly vary by adding further prototypes. An online scenario as well as noisy supervised information, which corrupt global quality assessments, prohibits such methods. In our approach the network size is controlled by a local utility function without a criterion of global classification performance or measure for model complexity. In comparison to the work of Hamker [2] we avoid to use (non-normalized) distance-based error criteria for the insertion and removal of prototypes from the network which is attributed to the local metrics of the prototypes. We place new prototypes according to a confidence criterion on the decision boundary and rate this placement afterwards by the utility criterion.

Network Expansion Since a confident global measurement of the representation quality is not available to determine when it is necessary to introduce new prototypes the network is expanded in specified time intervals. To decide where a new prototype can be added possible criteria are random insertion, a placement on false classified data or on the decision boundary. In prototype-based networks the decision boundary can be characterized by a similar distance of a feature to two prototypes from different classes. In particular the objective of GLVQ is to minimize an error functional which represents not only the classification error but also introduces an error term for unconfidently classified data points which bases on the difference (the margin) in the nominator of the function $\mu(d)$. Using the margin for learning for example was proposed in the context of active learning by Schleif et al. [10]. Here new data points for learning are acquired on the basis of the margin criterion. But this information was not used in the context of incremental learning before. Since the margin is implicitly optimized by the GLVQ error function, we decide to add new prototypes in these regions of low confidence, respectively directly on the decision boundary. Therefore for each expansion step a new prototype is positioned at the training vector with the minimum normalized margin $m(\xi^i) = \frac{\|d_j^i - d_k^i\|}{d_j^i + d_k^i}$. The label of the new prototype is initialized according to the supervised information, while the relevance matrix is taken from the best matching correct prototype according to this label. Since the network size is not adapted on a single frame and the data is changing from one to another frame, adding prototypes does not affect the optimization of the margin but provides a better initialization of the network for the adaptation on the next frame.

Network Contraction To rate the importance of every single prototype in the network a local utility criterion can be used. In the context of vector quantization Fritzke [11] proposes to rate single neurons according to the quantization error of a prototype by the following utility function $U(w_p) := E[\mathcal{D}, \mathcal{P} \setminus w_p] - E[\mathcal{D}, \mathcal{P}] =$

$\sum_{\xi \in \mathcal{D}} \|\xi - \mathbf{w}_s\|^2 - \|\xi - \mathbf{w}_p\|^2$ where \mathbf{w}_s is the winning prototype from the set $\mathcal{P} \setminus \{\mathbf{w}_p\}$. As the quantization error (which is also exploited by Hamker [2] for a local utility function) is based on a global consistent metrics this method is not appropriate for localized adaptive metrics. Therefore this inspires a utility $u(\mathbf{w}_p)$ function on the basis of the classification error. For a single training example ξ this function is:

$$u(\mathbf{w}_p, \mathbf{w}_s, \xi) = \begin{cases} 1 & c(\mathbf{w}_p) = c(\xi), c(\mathbf{w}_s) \neq c(\xi) \\ 0 & \text{else} \end{cases}$$

Finally the utility of the prototype on the whole dataset is normalized by the number of activations $n(\mathbf{w}_p) = |\{\xi | d(\mathbf{w}_p, \xi) = \min_{q \in \mathcal{P}} d(\mathbf{w}_q, \xi)\}|$ of this prototype: $U(\mathbf{w}_p) = \frac{1}{n(\mathbf{w}_p)} \sum_{\xi \in \mathcal{D}} u(\mathbf{w}_p, \mathbf{w}_s, \xi)$. If the value $U(\mathbf{w}_p)$ falls below a given threshold $t_u = 0.01$ in our experiments, the prototype is regarded as a removal candidate. After an expansion step, the new prototype is kept, if this one and all other current prototypes are useful (i.e. $U(\mathbf{w}_p) > t_u \forall p \in \mathcal{P}$), which assures to avoid unnecessary instabilities of the network. Independent of the utility function to evaluate the success of an expansion step, we use this function for separate contraction steps of the whole network to determine possibly spare prototypes or misplaced prototypes. Spare prototypes can be replaced by another prototype without impairing the performance. Misplaced prototypes can be characterized by an assignment to the wrongly labeled subset of data by the initial hypothesis \mathcal{H} . Usually this causes in the application/segmentation step that more image portions of the background are assigned to the foreground. These badly placed prototypes can be identified to cause a large classification error even on correctly labeled data and therefore reduce the overall segmentation quality. Together with the recorded activation $n(\mathbf{w}_p)$ we use the utility criterion to remove such prototypes. That is, additional to the utility criterion a prototype is removed if $\frac{n(\mathbf{w}_p)}{|\mathcal{D}|} < t_n$, where $t_n = 0.005$.

Algorithm

1. Input and preprocessing:
 - feature maps and hypothesis from object ROI: $\mathcal{F}^{x,y} := \{F_i^{x,y} | i = 1..M\}$, $\mathcal{H}^{x,y} \in \{0, 1\}$
 - Preprocessing of feature maps \mathcal{F} and hypothesis \mathcal{H} , see Sec. 3
 - Init codebook and metric (on first frame only) $\mathcal{P} = \{\mathbf{w}_p | p = 1, \dots, N\}$ where $N = 2$, $\forall \mathbf{w}_p \in \mathcal{P} : \mathbf{w}_p = \frac{1}{|L|} \sum_L \xi$, $L := \{\xi | c(\xi) = c(\mathbf{w}_p)\}$
 - Replace the data of the oldest frame by the data of current feature maps \mathcal{F} in the short term history \mathcal{D}
2. Adaptation (for T update steps)
 - Find best matching prototypes \mathbf{w}_J for the correct label, \mathbf{w}_K for the incorrect label according to a randomly selected $\xi^i \in \mathcal{D}$.
e.g. $\mathbf{w}_J = \{\mathbf{w}_p \in \mathcal{P} | d(\mathbf{w}_p, \xi^i) = \min_{q, c(\mathbf{w}_q) = \mathcal{H}^i} d(\mathbf{w}_q, \xi^i)\}$

- Update prototypes $\mathbf{w}_{J,K}$ by means of $\mathbf{w}_{J,K} \leftarrow \mathbf{w}_{J,K} + \alpha \cdot \Delta \mathbf{w}_{J,K}$ with learning rate $\alpha = 0.05$ and similar the relevance factors $\Lambda_{J,K}$ with $\alpha = 0.005$
- 3. Evaluation: for all pixels $i \in \mathcal{D}$
 - $\forall \mathbf{w}_p \in \mathcal{P}, V_p^i := \begin{cases} 1 & \text{if } d(\boldsymbol{\xi}^i, \mathbf{w}_p) < d(\boldsymbol{\xi}^i, \mathbf{w}_r), \forall r \neq p, \{r, p\} \in \mathcal{P}, \\ 0 & \text{else} \end{cases}$
 - Determine the binary foreground segmentation $\mathcal{A} = \sum_p^N c(\mathbf{w}_p) \cdot V_p$
 - Compute margin for every feature $m(\boldsymbol{\xi}^i) = \frac{d_J^i - d_K^i}{d_J^i + d_K^i}$
 - Compute utility $U(\mathbf{w}_p)$ and prototype activation $n(\mathbf{w}_p)$, Sec. 2.3
- 4. (Optional) Network Expansion
 - $\mathbf{w}_{new} = \boldsymbol{\xi}^i$ where $i = \arg \min_{\boldsymbol{\xi}^i \in \mathcal{D}} m(\boldsymbol{\xi}^i)$, $c(\mathbf{w}_{new}) = \mathcal{H}^i$, $\Lambda_{new} = \Lambda_J$
 - $\mathcal{P} = \{\mathcal{P}, \mathbf{w}_{new}\}$, $N = N + 1$
- 5. (Optional) Network Contraction
 - select \mathbf{w}_p with the smallest utility $p = \arg \min_{p \in \mathcal{P}} U(\mathbf{w}_p)$
 - remove \mathbf{w}_p if $U(\mathbf{w}_p) < t_u$ or $n(\mathbf{w}_p) < t_n$, $\mathcal{P} = \mathcal{P} \setminus \{\mathbf{w}_p\}$, $N = N - 1$

3 Results

Data Finally we evaluate the capabilities of this approach on challenging real world image data and investigate the effort of the derived object segmentations in the context of online object learning and recognition. Here we are using the data from [6] consisting of 50 natural, view centered objects with 300 training and 100 testing images. After the acquisition of the feature maps \mathcal{F} and the hypothesis \mathcal{H} a pre-processing $F_i^{x,y} \leftarrow T_F(F_i^{x,y})$ of the feature maps $F_i^{x,y}$ (a gamma correction and white balancing on the maps representing the RGB image data) is performed first. From the available depth and skin information the hypothesis \mathcal{H} is computed where all skin-colored areas \mathcal{S} , $\mathcal{S}^{x,y} \in \{0, 1\}$ are removed from the hypothesis $\mathcal{H} \leftarrow T_H(\mathcal{H})$, where $(T_H(\mathcal{H}) := \mathcal{H} \leftarrow \mathcal{H} - (\mathcal{H} \cap \mathcal{S}))$. This is necessary because the hand is strongly connected to every object/hypothesis and can be regarded as systematic noise violating our assumptions. To compare the results with previous work, the image regions defined by the foreground classification (i.e. the presented objects) are fed into a hierarchical feature processing stage [6]. For object learning and recognition a separate nearest neighbor classifier is applied to the derived high dimensional shape features. The separation of training and test data is used for the object classifier, while the incremental segmentation is adapted on a subset of the pixel data for every single frame.

Network Dimensionality First the behavior of the algorithm is analyzed on an example of the training-dataset in Fig. 2. The change in object identity yields an adaptation of the number of prototypes in particular for the foreground, which shows significant differences for some of the objects dependent on their subjective visual complexity. To avoid an influence from the sequence of the presented objects, the order of the 50 objects was randomly rearranged for the eight repetitions of the experiment. In contrast to previous work a reduced complexity of the representation finally allows for a more efficient processing of single frames.

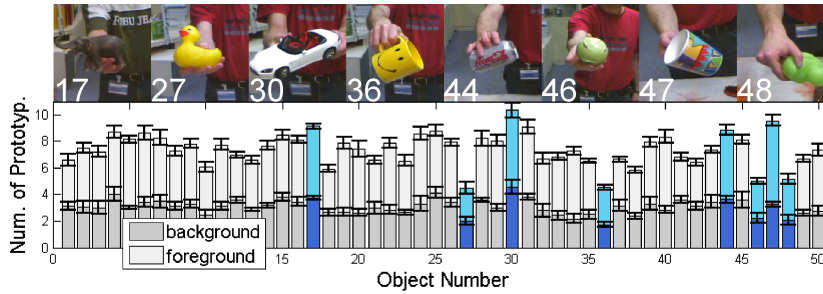


Fig. 2: Number of prototypes for an application to the training-dataset (50 objects with 300 views, each bar is the average of 8 repetitions and 300 views each). On average over all objects 4.35 prototypes are used for foreground and 3.07 are used for background. Additionally the object specific std. dev. of the average number of prototypes for multiple repetitions is drawn, which shows that this number for a particular object is consistent over multiple repetitions of the experiment. On top, examples for eight objects with the highest and lowest number of prototypes are shown.

Classification Performance Compared to a predefined number of prototypes in previous results (see Table 1) three aspects are important: i) the general performance of the object classifier on the basis of the segmentation (an indirect quality assessment, verified in [1]), ii) the used resources to derive the results and iii) the variance in the results. Therefore we compare the incremental method to the results derived by predefined number of prototypes (chosen according to the average number of the incremental method). On the basis of the same resources, a comparable performance can be achieved. Remarkably the variance of the results is significantly decreased which indicates a higher robustness to the noisy supervised data by discarding misplaced prototypes. Together with a faster adaptation to the changing image data the incremental method also reduce the dependency on the initialization of the prototypes. Since the initialization for the fixed prototype setup was purely random this can explain the beneficial effect. Compared to an offline parameter search the incremental segmentation might not be able to reach the potentially maximum performance (for 20 prototypes, 15 background - 5 foreground on this dataset), but offers an application to data with unknown “optimal” number of prototypes.

4 Conclusion

In this paper we present an incremental learning scheme for GLVQ in the context of figure-ground segmentation. In presence of local adaptive metrics and supervised noisy information we use a parallel evaluation scheme combined with a local utility function to organize a learning vector quantization with an adaptive number of prototypes. On our real world benchmark dataset we show, that the incremental network is capable to achieve a comparable (to the results from [1]) performance in hypothesis refinement while maintaining a significantly smaller

variance of the results, thus is more robust. Due to the parallel evaluation scheme the expansion of the network is free of additional computational load and does not impair the current performance of the network.

N (#bg/#fg)	2(1/1)	7(3/4)	20(15/5) [1]	adaptive	hypothesis[1]
mean	0.7442	0.8715	0.8828	0.8742	0.755
std. dev.	0.0132	0.0110	0.0252	0.0036	n.a.

Table 1: Results of the incremental segmentation scheme compared to previous results (average of 8 repetitions, except the last column). Dependent on the derived number of prototypes (on average 3 for background and 4 for foreground) the proposed method achieves a comparable performance to a predefined prototype setup, whereby the variance of the results is significantly reduced.

References

1. Alexander Denecke, Heiko Wersing, Jochen J. Steil, and Edgar Körner. Online figure-ground segmentation with adaptive metrics in generalized LVQ. *Neurocomputing*, 72(7-9):1470 – 1482, 2009.
2. Fred H. Hamker. Life-long learning cell structures —continuously learning without catastrophic interference. *Neural Networks*, 14(4-5):551–573, 2001.
3. S. Kirstein, H. Wersing, and E. Körner. A biologically motivated visual memory architecture for online learning of objects. *Neural Networks*, 1:65–77, 2008.
4. Bernd Fritzke. A growing neural gas network learns topologies. In G. Tesauero, D. S. Touretzky, and T. K. Leen, editors, *NIPS*, pages 625–632. MIT Press, 1994.
5. A. Sato and K. Yamada. Generalized learning vector quantization. In *Advances in Neural Information Processing Systems*, volume 7, pages 423–429, 1995.
6. H. Wersing, S. Kirstein, M. Götting, H. Brandl, M. Dunn, I. Mikhailova, C. Goerick, J. J. Steil, H. Ritter, and E. Körner. Online learning of objects in a biologically motivated visual architecture. *Int. J. Neur. Syst.*, 17(4):219–230, 2007.
7. P. Schneider, M. Biehl, F.-M. Schleif, and B. Hammer. Advanced metric adaptation in Generalized LVQ for classification of mass spectrometry data. In *Proceedings of 6th International Workshop on Self-Organizing Maps (WSOM)*, 2007.
8. K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin analysis of the LVQ algorithm. In *NIPS*, 2002.
9. A. Jirayusakul and S. Auwatanamongkol. A supervised growing neural gas algorithm for cluster analysis. *Int. J. Hybrid Intell. Syst.*, 4(4):217–229, 2007.
10. F. M. Schleif, B. Hammer, and T. Villmann. Margin-based active learning for LVQ networks. *Neurocomput.*, 70(7-9):1215–1224, 2007.
11. B. Fritzke. The LBG-U method for vector quantization – an improvement over LBG inspired from neural networks. *Neural Process. Lett.*, 5(1):35–45, 1997.