Online Semi-Supervised Learning with Adaptive Vector Quantization

Yuan-Yuan Shen[1,2] Xu-Yao Zhang[1,2] Cheng-Lin Liu[1,2,3]
[1]National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing, P.R. China
[2]University of Chinese Academy of Sciences, Beijing, P.R. China
[3]CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing, P.R. China
Email: shenyuanyuan2015@ia.ac.cn {xyz, liucl}@nlpr.ia.ac.cn

Abstract—This paper considers the online semi-supervised learning (OSSL) problem in which the data are a mixture of both labeled and unlabeled samples and appear in a sequential (stream) manner. OSSL is very common in real applications and similar to the human-like learning process. Prototype-based classifiers, which represent the data of different classes by some prototypes, are natural in a streaming scenario by updating the prototypes with online (incremental) learning. However, most of previous prototype-based models are either designed for supervised or unsupervised learning separately. In this paper, we propose a novel model called online adaptive vector quantization (OAVQ) aiming at improving the classification performance in case of OSSL. Specially, we use the learning vector quantization (LVQ) criterion for updating the prototypes when the data point is labeled, and the frequency sensitive competitive learning (FSCL) criterion for adjusting the prototypes when the data point is unlabeled. The labeled and unlabeled data are coming randomly in a sequential manner, and these two criteria are used alternatively to learn the positions of prototypes. In this way, we can make full use of both supervised and unsupervised information to further boost the performance. Experiment results on several databases verify the effectiveness and applicability of the proposed method in improving the performance for OSSL.

Index Terms—learning vector quantization, frequency sensitive competitive learning, adaptive vector quantization, online learning, semi-supervised learning

I. Introduction

Pattern recognition methods can usually be partitioned into supervised and unsupervised models. A more general case is semi-supervised learning (SSL) which makes use of both the labeled and unlabeled data to learn the classifier. Since we can easily collect large amount of unlabeled data compared with the complicated process of preparing labeled data, SSL usually achieves much better performance than supervised and unsupervised learning. Moreover, in many real world applications, the training process has to incrementally learn the model on a stream of patterns, which means the models need to be adjusted over time when new patterns (either labeled or unlabeled) appear, and this kind of process is called online semi-supervised learning (OSSL). In this setting, at each time, we may have only one or a small number of data points and the whole training data are usually not stored in the learning process. Furthermore, in some cases, the algorithms are restricted to a situation of single-pass learning where each data point only appear once. OSSL is similar to the human learning process and becomes very common in real application. How to efficiently and effectively make use of labeled and unlabeled information in an online fashion makes OSSL a challenging problem.

Prototype-based methods represent the training data with a set of points in feature space (called prototypes), and have found their utility in wide range of problems. Generally, there are two kinds of prototype-based learning paradigms, namely supervised classification models which learn several prototypes as typical representatives for each class, and unsupervised clustering models in which a set of prototypes are treated as an approximate representation for the whole dataset.

Learning vector quantization (LVQ) [1] as a well known prototype learning algorithm has been widely developed in supervised learning. Kohonen et al. proposed a number of improved versions of LVQ such as LVQ2, LVQ2.1, and LVQ3 [2]. Crammer et al. [3] showed that LVQ falls in a family of maximal margin learning algorithms providing a rigorous upper bound of generalization error. Jin et al. proposed optimizing the log-likelihood of hypothesis margin (LOGM) [4] for improving the convergence of training and generalization performance.

However, most existing LVQ methods are usually trained in batch mode. In order to cope with online learning, many algorithms [5]–[7] have been proposed. Online learning vector quantization (OLVQ) as an important type of online prototyped learning algorithms has been researched extensively in [8]–[11]. When a new pattern is arriving, two nearest prototypes from the positive class and the rival class are updated, or a new prototype is added in OLVQ methods. The main difference among those OLVQ methods is the manner of how to introduce new prototypes. In [8] and [9], new prototypes are added according to the increase of some error mechanism. When the number of new samples which are misclassified reaches to the predefined error number, a new prototype is added. In [10] and [11], new prototypes are added by a predefined threshold. When a new sample is arriving, the distance between the new sample and the nearest prototype is

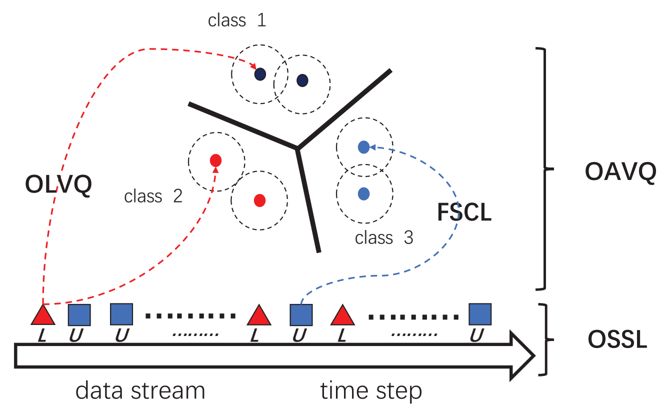


Fig. 1. An illustration of online semi-supervised learning (OSSL). The labeled data (red triangle) and unlabeled data (blue square) are coming randomly in a sequential manner. The proposed online adaptive vector quantization (OAVQ) includes two components: OLVQ and FSCL. When the data point is labeled, online learning vector quantization (OLVQ) criterion is used to update two prototypes (the genuine and rival prototypes). When the data point is unlabeled, the frequency sensitive competitive learning (FSCL) is used to update only one prototype (the nearest prototype weighted by the frequency). In OAVQ, these two processes are repeatedly and adaptively used according to the data labeling type, and in this way, both the supervised and unsupervised information are combined to boost the performance of OSSL.

computed. If the distance is bigger than the predefined threshold, then a new prototype is added. In all those methods, the final classification performance depends on the choose of the prototypes. Obviously, it is more benefit to the procedure of classification if more prototypes are saved. If all the new samples are saved, then the model is reduced to the traditional nearest neighbour classifier. However, due to the constraint on storage space, we assume that the number of prototypes is bounded by some pre-given quantity. In this situation, the choice of prototypes largely depends on the model parameter (i.e., the predefined error number or the predefined threshold) and sequence order of the data. It is also natural that each sample is implicitly assigned to the nearest representative prototype, thereby defining a partition of the whole feature space.

Unsupervised prototype-based learning is a form of unsupervised learning algorithms inspired by biological neural systems. Prominent models are the competitive learning (CL) [12], the frequency sensitive competitive learning (FSCL) [13], the self-organizing map (SOM) [14], and the growing neural gas network (GNG) [15]. In these approaches, only the best-matching prototype or its topological neighbors are updated. Specifically, CL adjusts the best-matching prototype which provides the highest similarity to the given input pattern whereas the rest of the neurons are left unchanged. FSCL is a conscience type competitive learning approach in which the competitive computing units are penalized in proportion to the frequency of their winning. The self-organizing map effectively creates spatially organized internal representations (i.e., prototypes) of various features for input signals. GNG successively learns the important topological relations in a given set of input vectors by means of a simple Hebb-like learning rule.

In this paper, to deal with the challenging OSSL problem, we propose a new model called online adaptive vector quantization (OAVQ) which combines the supervised and unsupervised prototype based learning methods in an unified framework. Specifically, the model is initialized using several initial labeled training samples, then the incremental mixed labeled and unlabeled samples are continually used for updating the prototypes. The prototypes are updated and shared for both supervised and unsupervised processes, and therefore, the class information of labeled data can be transferred to unlabeled data, while the clustering information from unlabeled data can be used to improve the classification performance. If the coming sample is labeled, then prototypes are updated by the mode of online learning vector quantization (OLVQ); otherwise analogy to the self-training, the nearest prototype from the unlabeled sample is being updated based on a modified objective function. Inspired by [13] which verifies the effectiveness of considering the update time of each prototype in unsupervised learning, the frequency-sensitive strategy is exploited in our paper. In this case, update time (frequency) is maintained for each prototype, the distance between sample and prototypes in unsupervised learning is redefined by the original distance between sample and the prototype and the frequency. When the prototype is computed as the nearest prototype, then the frequency of this prototype is increased by one. The less one prototype is updated in unlabeled situation, the easier it is chosen as the winner prototype for unlabeled sample. The flow diagram of our method is illustrated in Fig. 1.

The rest of the paper is organized as follows. Section II describes the proposed OAVQ model, section III reports the experimental results, and the last section concludes this paper.

II. Online adaptive vector quantization (OAVQ)

In this section, firstly we introduce our proposed online adaptive vector quantization (OAVQ) model, then we give detailed discussions on how to learn from labeled and unlabeled samples respectively.

A. OAVQ Framework

Suppose that we observe a sequence of feature vectors $x_1, x_2, ..., x_n$ where $x_t \in \mathbb{R}^d$ is a pre-defined feature representation. The x_t can be either labeled or unlabeled, if it is labeled, a class label y_t is also given. Till time T, here we use N_1 to represent the total number of labeled samples we have seen, and we also use N_2 to represent the total number of unlabeled samples. The model parameters need to be optimized are denoted by

m. Then, the objective function for online semi-supervised learning (OSSL) through time T can be summarized as:

$$F = \min \sum_{i=1}^{N_1} S(x_i, y_i, m) + \lambda \sum_{j=1}^{N_2} U(x_j, m), \tag{1}$$

where the hyperparameter λ controls the trade-off between supervised learning and unsupervised learning, and $S(.)$ and $U(.)$ are the loss functions for labeled samples and unlabeled samples respectively.

Specifically, in our proposed online adaptive vector quantization (OAVQ) model, two prototype-based criteria (i.e., OLVQ criterion for supervised learning and FSCL criterion for unsupervised learning) are combined to solve the OSSL problem. Here the model parameters refer to the prototypes which are shared in supervised and unsupervised learning. Detailed learning procedures are described in section II.B and section II.C, and we summarize the OAVQ process in Algorithm 1.

Algorithm 1 OAVQ

Require: labeled sample (x, y) or unlabeled sample x
prototypes m

```
while receive new pattern x do
    if x is labeled then
        update prototypes based on OLVQ criterion
    else
        update prototypes based on FSCL criterion
    end if
end while
```

B. Supervised Model Learning: OLVQ

In this sub-section, we describe the learning criterion of OLVQ for labeled samples. For a M-class classification problem, prototype learning is to learn a set of prototype vectors m_{ij} ($i = 1, 2, ..., M, j = 1, ..., n_i$) for each class. Here n_i is the number of prototypes in class i. The learning process is usually implemented by minimizing the empirical loss on a training set. An input pattern $x \in \mathbb{R}^d$ is classified to the class of the nearest prototype:

$$k = \arg \min_{i=1}^{M} \min_{j=1}^{n_i} \|x - m_{ij}\|_2^2 = G(x, m). \tag{2}$$

There are many variations of LVQ algorithm [1], [2], [4]. In this paper, we use the minimum classification error (MCE) [4], [16] criterion due to its good performance.

Specifically, given that m_1 and m_2 are two nearest prototypes to pattern x from the positive class and the rival class respectively, the posterior probability of x belonging to genuine class y (i.e., the probability of correct classification) can be approximated by the sigmoid function σ:

$$P(c|x; m) = \sigma(\xi_1 d(x)), \tag{3}$$

where ξ_1 ($\xi_1 > 0$) is a constant for tuning the smoothness of sigmoid function and $d(x)$ is computed as $d(x) = \|x - m_2\|^2 - \|x - m_1\|^2$. Then the conditional log-likelihood loss of pattern x is $S(x, y, m) = 1 - P(c|x; m)$.

As new pattern x is arriving, OLVQ updates the two prototypes m_1 and m_2 by gradient-based methods [17], [18]:

$$\begin{aligned} m_1 &= m_1 - \eta \frac{\partial S(x, y, m)}{\partial m_1}, \\ m_2 &= m_2 - \eta \frac{\partial S(x, y, m)}{\partial m_2}, \end{aligned} \tag{4}$$

where η is the learning rate.

C. Unsupervised Model Learning: FSCL

In this sub-section, we focus on the problem of how to update the prototypes for unlabeled samples. Based on the FSCL criterion, the prototype that provides the highest similarity to the given input pattern is declared as the winner node and is moved closer to the input pattern, whereas the rest of the prototypes are left unchanged.

Specifically, given that x is the arriving unlabeled sample at current time, then the probability of x belonging to prototype m can be approximated by the sigmoid function σ (similar to OLVQ):

$$P(x \in m) = \sigma(\xi_2 d(x)), \tag{5}$$

where ξ_2 ($\xi_2 > 0$) is a constant for tuning the smoothness of sigmoid function and $d(x)$ is computed as $d(x) = \|x - m_{h^*}\|^2$ (m_{h^*} is the winner prototype where $h^* = \arg\min_h \|x - m_h\|^2$). Based on the winner-take-all strategy, only the nearest prototype is considered in the definition of affiliation function.

When frequency is introduced, suppose that m_{h^*} is the winner prototype, then h^* is computed as:

$$h^* = \arg\min_h n_h \|x - m_h\|^2, \tag{6}$$

where n_h is the frequency of the prototype. Once a prototype is updated by the unlabeled sample, then the relative frequency of this prototype n_h is increased by one. Obviously, this strategy can make a balance in all the prototypes for unsupervised learning. If a prototype is updated for only a few times, then it is easier to be chosen as the winner prototype. In this way, all prototypes can be well activated in the learning process.

Similarly, the loss function for unlabeled sample x can be defined as:

$$U(x, m) = -f * (1 - P(x \in m_{h^*})), \tag{7}$$

where f is the degree of confidence. f is calculated as:

$$f = \sigma(\xi_2(\|x - m_2\|^2 - \|x - m_1\|^2)), \tag{8}$$

here m_1 and m_2 are top two nearest prototypes with the sample x from two different classes. Estimating the uncertainty from two best predicted classes [19] has been proved to be very useful in active learning. If the margin for the distance is larger, the confidence should be larger for this sample.

As unlabeled new pattern x is arriving, the nearest prototypes m is also updated by gradient-based methods [17], [18]:

$$m_{h^*} = m_{h^*} - \eta \frac{\partial U(x, m)}{\partial m_{h^*}}, \tag{9}$$

and here η is the learning rate.

Algorithm 2 Modified FSCL

for next data point x do
  Assign x to the prototype $m_{h^*}^{(t)}$ where $h^* = \arg\min_h n_h^{(t)} \|x - m_h^{(t)}\|^2$
  if $h == h^*$ then
    Set $n_{h^*}^{(t+1)} \leftarrow (1 - 1/L) n_{h^*}^{(t)} + 1$
    Update $m_{h^*}^{(t+1)}$ using Equation (9)
  else
    $n_{h^*}^{(t+1)} \leftarrow n_h^{(t)}$
    Set $m_h^{(t+1)} = m_h^{(t)}$
  end if
end for

Since we have made some modifications to the traditional FSCL, we summarize the learning process in Algorithm 2. Note that instead of increasing by one for frequency in each update, a decreasing sequence parameterized by L is used here to improve the performance. One explanation for this setting is that as the algorithm tends to be stable over time, the effect from the frequency should be decreased.

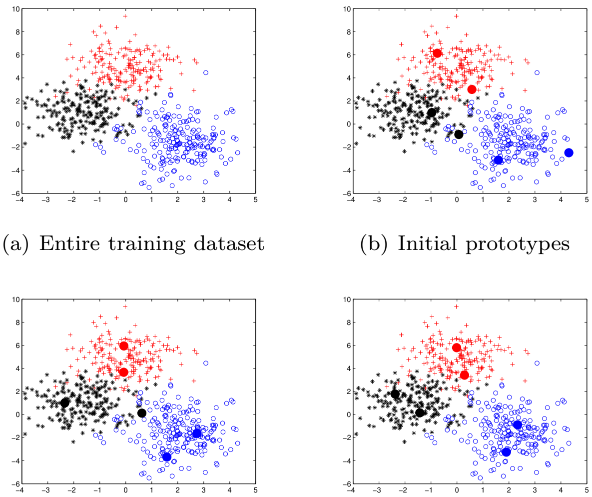
III. Experiments

In this section, we conduct experiments on both artificial dataset and real-world datasets to evaluate the model. An artificially generated dataset is used firstly to illustrate the effectiveness of incorporating unlabeled data into the learning process. The proposed algorithm is then applied to some real-world datasets to further verify its performance.

A. Artificial dataset

The artificial dataset consists of three Gaussian distributions, each representing a class. The dataset consists of 300 points in total with 100 points for each class. The distributions are centred at [0, 5], [-1.5, 1] and [2, -1.5] respectively, with the covariance matrix [1 0; 0 2], [1 0; 0 2] and [1 0; 0 2]. The dataset and the learned prototypes for different methods are figured in Fig. 2. Solid bold circles represent the learned prototypes in Fig. 2b to Fig. 2d.

In order to illustrate the benefits of unsupervised learning, we omit the supervised process in this experiment. In the above four figures, Fig. 2a shows scatter plots of all training data. Then we randomly choose two initial prototypes for each class in Fig. 2b. By continuing to learn models from unlabeled samples in the manner of CL and FSCL respectively, Fig. 2c and Fig. 2d are produced. Here we take the learning process of the two prototypes from the class which are painted black as an example.



(a) Entire training dataset (b) Initial prototypes

(c) Learn with CL (d) Learn with FSCL

Fig. 2. (a) Scatter plots of all training data. (b) Scatter plots of random initial prototypes. (c) Scatter plots of learning with online competitive learning strategy. (d) Scatter plots of learning with online frequency sensitive competitive learning strategy.

TABLE I
Real-world datasets

Dataset	Size	Dim	Class
DNA	2,000/1,186	180	3
Pendigits	7,494/3,498	16	10
USPS	7,291/ 2,007	256	10
MNIST	60,000/ 10,000	784	10

From Fig. 2b we observe that the initial two prototypes are too close to the right and are not benefit to the process of the classification. By learning the distribution information from unlabeled samples with CL criterion, in Fig. 2c we can see that the left one of the two prototypes are shift to the left. Thus the two prototypes become a better representation of the class which are painted black. Furthermore, online learning vector quantization (OLVQ) is sensitive to the initial prototypes. A typical case in our data is the right prototype of the labeled black. In Fig. 2c, it rarely can be updated or wrongly updated by the pattern from the class which are painted blue. In Fig. 2d FSCL strategy is considered and we can obtain more reasonable prototypes by using this approach.

B. Real-world data set

To test our proposed method on realistic data, we conduct experiments with the UCI datasets including: (a) dna, (b) pendigits, (c) usps, (d) mnist. The statistics of the above datasets are listed in Table I. As discussed in previous sections, the results of online learning are usually influenced by the order of the samples. In order to yield stable results, the experiment for each dataset is implemented for 20 times, and for each time the appearing of the samples are arranged randomly. In our experiments,

TABLE II
Parameter setting in our experiment.

Dataset	Number of Prototypes	Initial Learning Rate
DNA	3	0.1
Pendigits	10	0.01
USPS	10	0.2
MNIST	8	0.1

the first 20 patterns from each class are used to train the initial prototypes. Specifically, we predefine the number of the prototypes as P. The first 20 patterns from each class are clustered into P clusters and the cluster centers are viewed as the initial prototypes. For the same dataset, the number of prototypes for different methods is identical. For all the remaining data, we randomly choose a certain percentage of samples as labeled data. Then the mixed data with labeled and unlabeled samples are arriving one-by-one randomly. Adagrad algorithm [18] is exploited for updating model in our method.

For all datasets, the balance parameter λ is set as 0.01. The hyperparameters ξ_1 and ξ_2 are initialized from the training samples. Suppose that the number of the initial training samples is K, then ξ_1 is estimated as $\frac{2}{\frac{1}{K}\sum_{i=1}^{K}|(\|x_i-m_2\|^2-\|x_i-m_1\|^2)|}$ (m_1 and m_2 are two nearest prototypes to pattern x_i from the positive class and the rival class respectively) and ξ_2 is estimated as $\frac{2}{\frac{1}{K}\sum_{i=1}^{K}(\|x_i-m_1\|^2)}$.

The number P of prototypes is chosen from 3 to 10 and the initial learning rate of gradient descent is chosen from {0.01, 0.1, 0.2}. The number of prototypes and the initial learning rate are set as Table II.

Table III shows the results of OLVQ, OLVQ+CL and OAVQ respectively. OLVQ is a purely online supervised classifier, while OLVQ+CL and OAVQ are the online semi-supervised learning models. Learning from unlabeled samples is based on CL strategy and FSCL strategy in OLVQ+CL and OAVQ respectively. Different percentages (from 1% to 100%) of samples are labeled to evaluate the effectiveness of the proposed method. Best performance of the compared methods are given in bold. From Table III, we can observe that when the percentage of labeled samples is less than 10%, OLVQ+CL and OAVQ have a clear advantage over OLVQ. Usually the unlabeled data we use is sampled from the same distribution with the labeled data, then the data distribution information contained in unlabeled data is also helpful to the classification. OLVQ+CL/OAVQ introduces the learning of unlabeled samples in OLVQ and the better classification performance is obtained than OLVQ. This is useful in the situation that the training set is the combination of a small portion of labeled samples and a large portion of unlabeled samples. However, as the labeled data being increased continually, the performance promotion of classification is less or even the classification performance is degraded slightly. This is caused mainly by the accumulative error of learning

TABLE III
Error rates on four UCI dataset.

Dataset	Labeled percentage	OLVQ	OLVQ+CL	OAVQ
DNA	1%	25.41	24.88	24.25
	2%	24.00	23.46	22.73
	4%	21.13	20.93	20.65
	8%	17.62	17.28	17.27
	16%	15.37	14.74	14.98
	32%	11.51	11.64	11.68
	64%	9.24	9.23	9.22
	100%	8.04	8.04	8.04
PENDIGITS	1%	10.17	9.8	9.71
	2%	9.80	9.43	9.42
	4%	9.68	9.47	9.40
	8%	9.51	9.39	9.37
	16%	9.15	9.06	9.04
	32%	8.19	8.20	8.20
	64%	7.82	7.84	7.81
	100%	7.16	7.16	7.16
USPS	1%	14.22	12.81	12.99
	2%	13.49	12.59	12.51
	4%	12.39	11.35	11.40
	8%	11.06	10.54	10.53
	16%	9.47	9.48	9.36
	32%	8.35	8.33	8.15
	64%	7.20	7.36	7.36
	100%	6.63	6.63	6.63
MNIST	1%	12.77	11.31	11.19
	2%	10.56	10.02	9.52
	4%	9.11	8.86	8.41
	8%	7.47	7.45	7.26
	16%	6.41	6.51	6.35
	32%	5.59	5.71	5.58
	64%	4.99	4.97	4.98
	100%	4.58	4.58	4.58

unsupervised samples. For unlabeled data, the winner prototype is possible to come from the different class with the arriving sample (i.e., the prototype should be updated is not been adjusted, and meanwhile the prototype should not be updated is adjusted). Semi-supervised models balance the advantage of distribution information and the disadvantage of the wrongly updating. When the labeled samples are in the majority, the distribution information contained in the labeled samples is enough. Then the advantage of distribution information is weakened and the accumulative error from unlabeled samples become the main factor. Finally, considering the updating frequency of each prototype from unlabeled samples, OAVQ tends to choose prototypes which has the lower frequency in unsupervised model learning and always has better or comparable performance than OLVQ+CL.

IV. Conclusion

In this paper, we consider an important problem of online semi-supervised learning (OSSL). OSSL is very common in real applications, where potentially unlimited data arrive sequentially, which cannot be entirely stored and only a small fraction of them are labeled. We propose a new model called online adaptive vector quantization (OAVQ) to solve this problem which includes two basic components: the OLVQ criterion for dealing with labeled samples and the FSCL criterion for handing unlabeled samples. Experimental results show the effectiveness of OAVQ on both artificial and real datasets. Our future work will consider extending OAVQ to other challenging problems such as class-incremental learning.

Acknowledgment

This work has been supported by the National Natural Science Foundation of China (NSFC) Grants 61411136002, 61633021 and 61721004.

References

[1] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola, “LVQ PAK: the learning vector quantization program package,” in Technical Report, Helsinki University of Technology, 1995.

[2] T. Kohonen, “Improved versions of learning vector quantization,” Neural Networks, vol. 1, no. 17-21, pp. 545–550, 1990.

[3] K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby, “Margin Analysis of the LVQ Algorithm,” in Advances in Neural Information Processing Systems NIPS, 2002, pp. 462–469.

[4] X. Jin, C.-L. Liu, and X. Hou, “Regularized margin-based conditional log-likelihood loss for prototype learning,” Pattern Recognition, vol. 43, no. 7, pp. 2428–2438, 2010.

[5] N. A. Syed, S. Huan, L. Kah, and K. Sung, “Incremental learning with support vector machines,” in International Joint Conference on Artificial Intelligence, 1999, pp. 352–356.

[6] S. C. H. Hoi, J. Wang, and P. Zhao, “LIBOL: a library for online learning algorithms,” Journal of Machine Learning Research, vol. 15, no. 1, pp. 495–499, 2014.

[7] P. Zhao and S. C. H. Hoi, “OTL: A framework of online transfer learning,” in International Conference on Machine Learning (ICML), 2010, pp. 1231–1238.

[8] T. C. Kietzmann, S. Lange, and M. A. Riedmiller, “Incremental GRLVQ: Learning relevant features for 3D object recognition,” Neurocomputing, vol. 71, no. 13-15, pp. 2868–2879, 2008.

[9] S. Kirstein, H. Wersing, and E. Körner, “A biologically motivated visual memory architecture for online learning of objects,” Neural Networks, vol. 21, no. 1, pp. 65–77, 2008.

[10] E. Lughofer, “Evolving vector quantization for classification of on-line data streams,” in International Conferences on Computational Intelligence for Modelling Control and Automation (CIMCA), 2008, pp. 779–784.

[11] Y. Xu, F. Shen, and J. Zhao, “An incremental learning vector quantization algorithm for pattern classification,” Neural Computing and Applications, vol. 21, no. 6, pp. 1205–1215, 2012.

[12] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, “Competitive learning algorithms for vector quantization,” Neural Networks, vol. 3, no. 3, pp. 277–290, 1990.

[13] A. Banerjee and J. Ghosh, “Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres,” IEEE Trans. Neural Networks, vol. 15, no. 3, pp. 702–719, 2004.

[14] T. Kohonen, “The self-organizing map,” Neurocomputing, vol. 21, no. 1-3, pp. 1–6, 1998.

[15] B. Fritzke, “A growing neural gas network learns topologies,” in Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994], 1994, pp. 625–632.

[16] B.-H. Juang and S. Katagiri, “Discriminative learning for minimum error classification,” IEEE Trans. Signal Processing, vol. 40, no. 12, pp. 3043–3054, 1992.

[17] C. M. Bishop, Pattern recognition and machine learning, ser. Information science and statistics. Springer, 2007.

[18] J. C. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” Journal of Machine Learning Research, vol. 12, pp. 2121–2159, 2011.

[19] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, “Multi-class active learning for image classification,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1063–6919.