Fuzzy-Kernel Learning Vector Quantization

Daoqiang Zhang[1], Songcan Chen[1] and Zhi-Hua Zhou[2]

[1] Department of Computer Science and Engineering
Nanjing University of Aeronautics and Astronautics
Nanjing 210016, P.R. China
{dqzhang, s.chen}@nuaa.edu.cn
[2] National Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, P.R. China
zhouzh@nju.edu.cn

Abstract. This paper presents an unsupervised fuzzy-kernel learning vector quantization algorithm called FKLVQ. FKLVQ is a batch type of clustering learning network by fusing the batch learning, fuzzy membership functions, and kernel-induced distance measures. We compare FKLVQ with the well-known fuzzy LVQ and the recently proposed fuzzy-soft LVQ on some artificial and real data sets. Experimental results show that FKLVQ is more accurate and needs far fewer iteration steps than the latter two algorithms. Moreover FKLVQ shows good robustness to outliers.

1 Introduction

The self-organizing map (SOM) due to Kohonen [1] is an ingenious neural network and has been widely studied and applied in various areas. The SOM network uses the neighborhood interaction set to approximate lateral neural interaction and discover the topological structure hidden in the data. The unsupervised learning vector quantization (LVQ) [2] can be seen as a special case of the SOM, where the neighborhood set contains only the winner node. Such learning rule is also called the winner-take-all principle.

LVQ has attracted a lot of attentions because of its learning simplicity and efficiency. However, LVQ suffers from several major problems when used for unsupervised clustering. Firstly, LVQ is sequential, so the final result severely depends on the order which the input patterns are presented to the network and usually a lot of numbers of iteration steps are needed for termination. Secondly, LVQ suffers the so-called prototype under-utilization problem, i.e., only the winner is updated for each input. Finally, because of adopting Euclidean distance measure, LVQ can cause bad performance when the data is non-spherical distribution, and especially contains noises or outliers.

To solve the first problem, the batch LVQ is proposed, which comes from the notion of batch SOM [1]. And fuzzy membership functions are introduced to original LVQ to overcome the second problem. For example, Yair et al. [3] proposed a soft competitive learning scheme to LVQ. To simultaneously address the above two problems some batch version of fuzzy LVQs are proposed. Bezdek et al. [2, 4] proposed the well-known fuzzy LVQ (FLVQ). Wu and Yang [5] presented a fuzzy-soft LVQ (FSLVQ). However, both FLVQ and FSLVQ use the Euclidean distance measure, and hence they are effective only when data set is spherical alike distributed. In addition, according to Huber's robust statistics [6], the Euclidean distance measure is not robust, i.e., sensitive to noises and outliers. To solve this problem, a sequential kernel SOM was proposed in one of our recent works, where a kernel-induced distance measures replaces original Euclidean one in order to improve robustness to outliers [7].

In this paper, we advance the sequential kernel SOM to the batch type of clustering learning network and call it a fuzzy-kernel LVQ (FKLVQ). Our goal aims to making FKLVQ simultaneously solve the three problems of LVQ by fusing the batch learning, fuzzy membership functions, and kernel-induced distance measures together. We made comparisons between FKLVQ and other types of batch LVQ on some artificial and real data sets.

2 Fuzzy-Kernel Learning Vector Quantization

As already mentioned in the previous section, FKLVQ consists of three main parts, i.e. batch learning, fuzzy memberships and kernel-induced distance measures. Before presenting FKLVQ algorithm, we first introduce the batch LVQ, fuzzy membership function and kernel-induced distance measures used in FKLVQ.

2.1 Batch LVQ

The LVQ for unsupervised clustering is a special case of the SOM network [5]. Suppose W_i in R^s is the weight vector of the node i and the input sample x_k in R^s is presented online at time t, sequential LVQ updates its neuron i as follows:

$$W_i(t) = W_i(t-1) + \alpha(t) h_{ik} \left(x_k - W_i(t-1) \right). \tag{1}$$

Here $\alpha(t)$ is the scalar-valued learning rate, $0<\alpha(t)<1$, and decreases monotonically with time. h_{ik} is an indicative function whose value is 1 if i is the winner node j, and 0 otherwise, where the winner node j is computed as follows:

$$\forall i, \quad \| x_k - W_j(t-1) \| \leq \| x_k - W_i(t-1) \|. \tag{2}$$

Suppose that the sample set are $X=\{x_1,\ldots,x_n\}$, where n is fixed. the sequential LVQ can be replaced by the following batch version which is significantly faster and does not require specification of any learning rate $\alpha(t)$.

Assume that the online algorithm will converge to a stationary state W_i^*, then the expectation values of $W_i(t)$ and $W_i(t\text{-}1)$ must be equal as t goes to infinity. In other words, in the stationary state we must have

$$E\left[h_{ik}\left(x_k - W_i^*\right)\right] = 0\,. \tag{3}$$

Applying the empirical distribution to solve the above equation, we have the batch learning formula of LVQ with [5]

$$W_i^* = \frac{\sum_k h_{ik} x_k}{\sum_k h_{ik}}\,. \tag{4}$$

Since the determination of h_{ik} still depends on W_i^* according to Eq. (2), an alternate iteration between Eqs. (2) and (4) is used to approximate the explicit solution of W_i^*.

2.2 Fuzzy Membership Function

The above batch LVQ is in fact equivalent to traditional *k*-means (also called hard *c*-means) clustering algorithm, whose fuzzy extension is the widely used fuzzy *c*-means algorithm (FCM) [8]. The key point of FCM is the use of membership function which originally exists in fuzzy sets. Given $X=\{x_1,\ldots,x_n\}$, FCM obtains a fuzzy *c*-partition of X with $\{u_1,\ldots,u_c\}$, where $u_{ik}= u_i(x_k)$ takes value in the interval [0,1] such that $\sum_i u_{ik}=1$ for all k. By optimizing the objective function of FCM, one can obtain the alternate iterative equations. When the Euclidean distance is used, the update equation of the membership u_{ik} is as follows [8]:

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\| x_k - W_i \| / \| x_k - W_j \| \right)^{2/(m-1)} \right)^{-1}. \tag{5}$$

Here $m>1$ denotes the degree of fuzziness, and as m approximate to 1^+, FCM degenerate into crisp k-means algorithm.

Inspired by the success of FCM, many researchers also introduced the fuzzy membership to original batch LVQ in a similar way [2, 5, 9]. That is often achieved by representing h_{ik} with some monotone functions of u_{ik} and afterwards alternately iterating between h_{ik} and W_i, e.g. the FLVQ due to Bezdek et al. [2] and the FSLVQ proposed by Wu and Yang [5].

2.3 Kernel-induced Distance Measures

Given input set X and a nonlinear mapping function Φ, which maps x_k from the input space X to a new space F with higher or even infinite dimensions. The kernel function is defined as the inner product in the new space F with: $K(x,y) = \Phi(x)^{\mathrm{T}}\Phi(y)$, for x, y in input space X.

An important fact about kernel function is that it can be directly constructed in original input space without knowing the concrete form of Φ. That is, a kernel function implicitly defines a nonlinear mapping function. There are several typical kernel functions, e.g. the Gaussian kernel: $K(x,y)=exp(-\|x-y\|^2/\sigma^2)$, and the polynomial kernel: $K(x,y)=(x^{\mathrm{T}}y + 1)^d$. From a kernel K, we have

$$d(x,y) = \| \Phi(x) - \Phi(y) \| = \sqrt{K(x,x) - 2K(x,y) + K(y,y)} \,. \tag{6}$$

Here $d(x, y)$ defines a class of kernel-induced non-Euclidean distance measures with varying kernel functions. In our early works, the kernel-induced distance measures have been adopted in sequential kernel SOM [7], clustering [10, 11] and image denoising [12] respectively. And it has been proved that the measure induced by the Gaussian kernel is more robust to noises and outliers compared with original Euclidean measure [11]. Hereafter, we only discuss the Gaussian kernel in the rest of the paper. For Gaussian kernel, we have $K(x,x)=1$ for all x. Thus the distance measure in Eq. (6) can be simplified as $d(x,y) = sqrt\,(2(1-K(x,y)))$.

2.4 The Proposed FKLVQ

We are in position to propose the FKLVQ algorithm now. Define the objective function between the weight vector W_i and the input sample x_k as follows:

$$J(W_i, x_k) = K(W_i, W_i) - 2K(W_i, x_k) + K(x_k, x_k) \,. \tag{7}$$

Minimizing Eq. (7) by gradient descent, we obtain the update equation for W_i as:

$$W_i(t) = W_i(t-1) - \alpha(t) h_{ik} \left(\frac{\partial J(W_i, x_k)}{\partial W_i} \right). \tag{8}$$

Especially for the Gaussian kernel $K(x,y)=exp(-\|x-y\|^2/\sigma^2)$, $K(W_i,W_i)=1$, so we have

$$W_i(t) = W_i(t-1) - \alpha(t) h_{ik} K\left(x_k, W_i(t-1)\right)\left(x_k - W_i(t-1)\right) \cdot 2/\sigma^2 \,. \tag{9}$$

Here $\alpha(t)$ is defined before, but h_{ik} is calculated as follows:

$$h_{ik} = \left(\frac{u_{ik}}{\max_{1 \le i \le c} \{u_{ik}\}} \right)^{(1+\sqrt{t}/c)} , \tag{10}$$

where u_{ik} is the fuzzy membership under kernel-induced measures. In a similar way to FCM, for the Gaussian kernel, u_{ik} can be derived as follows [10]:

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\left(1 - K(x_k, W_i)\right) \Big/ \left(1 - K(x_k, W_j)\right) \right)^{1/(m-1)} \right)^{-1} . \tag{11}$$

So far, we have derived the sequential fuzzy-kernel LVQ algorithm. Its batch version can be constructed in a similar way to that of batch LVQ. From Eq. (9), in the stationary state we must have

$$E\left[h_{ik} K(x_k, W_i) \left(x_k - W_i\right) \right] = 0 \,. \tag{12}$$

Applying the empirical distribution to solve the above equation, we have the batch learning formula of fuzzy-kernel LVQ with

$$W_i = \frac{\sum_k h_{ik} K(W_i, x_k) x_k}{\sum_k h_{ik} K(W_i, x_k)} . \quad (13)$$

Here h_{ik} is determined by Eq. (10), and by alternately iterating between Eqs. (11), (10) and (13), we get the FKLVQ algorithm.

3 Experimental Results

In this section, we make numerical comparison between the proposed FKLVQ and other batch algorithms such as FCM, FLVQ and FSLVQ on some artificial and real data sets. The Gaussian kernel is used for FKLVQ.

The fist example is an artificial data set which contains two clusters. Two clusters contain respectively 50 and 49 sample patterns and are separately centered at the points (0,0) and (3,0) with Gaussian distributions. Besides the two clusters, there exists an outlier at (200, 0). In this experiment, the parameters used in the algorithms are set to m=2, c=2, σ =20, and the maximum number of iterations is 50. Fig. 1 shows the comparison of four algorithms. From Fig. 1, results of FCM, FLVQ and FSLVQ are severely affected by the outlier, and the numbers of misclassified samples are all 49. However, FKLVQ successfully avoids the disturbance of the outlier and correctly classified the two clusters.

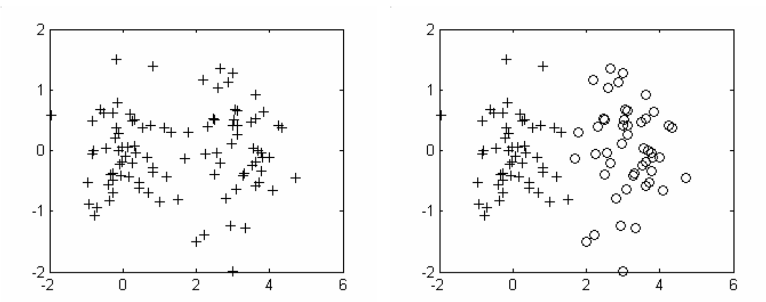

Fig. 1. Comparisons of performances of the four algorithms on artificial data set with outlier (not plotted in the figure): Left is the result by FCM, FLVQ and FSLVQ, where samples from both real clusters are classified to one group and the outlier is classified to the other group; Right is the result of FKLVQ where disturbance of the outlier is completely avoided.

The second example is the well-known Iris data set. It contains 3 clusters with 50 samples each. In this experiment, the parameters are set to m=2, c=3, σ =10, and the maximum number of iterations is 50. Table 1 gives comparison of accuracies and numbers of iterations of the four algorithms. From Table 1, FKLVQ achieves better accuracy and needs much less iteration than FLVQ and FSLVQ. For clustering the Iris data, FKLVQ only needs few tens of iterations. However, it was reported in [7] that the sequential kernel SOM typically needs hundreds to thousands of iterations for classifying the Iris data. Thus FKLVQ is much superior to kernel SOM as far as the computation efficiency is concerned.

Table 1. Comparison of accuracy and number of iterations of the four algorithms

Algo-rithms	Number of misclassified samples				Number of iterations
	Cluster 1	Cluster 2	Cluster 3	Total	
FCM	0	3	13	16	14
FLVQ	0	3	14	17	50
FSLVQ	0	3	14	17	41
FKLVQ	0	6	5	11	13

Acknowledgements

This work was supported by the National Outstanding Youth Foundation of China under the Grant No. 60325207 and the Natural Science Foundation of Jiangsu Province under the Grant No. BK2002092.

References

1. Kohonen, T.: The Self-Organizing Map. Neurocomputing 21 (1998) 1-6
2. Bezdek, J.C., Pal, N.R.: Two Soft Relatives of Learning Vector Quantization. Neural Networks 8 (1995) 729-743
3. Yair, E., Zeger, K., Gersho, A.: Competitive Learning and Soft Competition for Vector Quantization Desigh. IEEE Trans. Signal Process. 40 (1992) 294-309
4. Tsao, E.C.K., Bezdek, J.C., Pal, N.R.: Fuzzy Kohonen Clustering Networks. Pattern Recognition 27 (1994) 757-764
5. Wu, K.L., Yang, M.S.: A Fuzzy-Soft Learning Vector Quantization. Neurocomputing 55 (2003) 681-697
6. Huber, P.J.: Robust statistics. Wiley, New York (1981)
7. Pan, Z.S., Chen, S.C., Zhang, D.Q.: A Kernel-based SOM classifier in Input Space. ACTA ELECTRONICA SINICA 32 (2004) 227-231
8. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
9. Huntsberger, T., Ajjimarangsee, P.: Parallel Self-Organizing Feature Maps for Unsupervised Pattern Recognition. Intern. J. Gen. Systerms 16 (1990) 357-372
10. Zhang, D.Q., Chen, S.C.: Clustering Incomplete Data Using Kernel-based Fuzzy C-Means Algorithm. Neural Processing Letters 18 (2003) 155-162
11. Zhang, D.Q., Chen, S.C.: A Novel Kernelised Fuzzy C-Means Algorithm with Application in Medical Image Segmentation. Artificial Intelligence in Medicine, in press (2004)
12. Tan, K.R., Chen, S.C., Zhang, D.Q.: Robust Image Denoising Using Kernel-induced Measures. Accepted for publication in International Conference on Pattern Recognition (2004)