RBF Sinir Ağları Sınıflandırıcısının Başlangıç Parametrelerini Bulan Eğitmenli Topaklandırıcı Algoritması

A Supervised Clustering Algorithm for the Initialization of RBF Neural Network Classifiers

Hakan Cevikalp, Diane Larlus, Frederic Jurie

INRIA Rhone-Alpes France
hakan.cevikalp@gmail.com

Özetçe

Bu bildiride RBF sinir ağlarındaki saklı katman öğelerinin sayısını ve başlangıç yerlerini bulmak için tektürel topaklandırma algoritması adını verdiğimiz yeni bir yöntem önerdik. Bu amaçla kullanılan diğer yöntemlerin aksine, önerdiğimiz yöntem sınıfların etiket bilgisini kullanan bir yaklaşım olup, aranan parametreler sınıflar arasındaki örtüşmelere göre belirlenir. Önerilen yöntemin ardındaki temel fikir, içinde sadece tek bir sınıfa ait elemanlar bulunan tektürel topaklar oluşturmaktır. Önerilen yöntemi RBF sinir ağları sınıflandırıcısıyla birlikte Graz02 nesne veri tabanı ve ORL yüz veri tabanı üzerinde test ettik. Deneysel sonuçlar önerdiğimiz yöntemle başlatılan RBF sınıflandırıcının k-en-yakın-merkez topaklandırma yöntemiyle başlatılan RBF sınıflandırıcıdan çok daha iyi sonuçlar verdiğini göstermektedir. Aynı zamanda elde edilen tanıma oranları Graz02 nesne veri tabanı için literatürde yayınlanan en iyi tanıma oranlarını geçmekte ve ORL yüz veri tabanı için de benzer sonuçlar üretmektedir. Bu da bu çalışmada sunulan topaklandırma yönteminin saklı katman öğelerinin sayısını ve başlangıç yerlerini bulmada oldukça başarılı olduğunu göstermektedir.

Abstract

In this paper, we propose a new supervised clustering algorithm, coined as the Homogeneous Clustering (HC), to find the number and initial locations of the hidden units in Radial Basis Function (RBF) neural network classifiers. In contrast to the traditional clustering algorithms introduced for this goal, the proposed algorithm is a supervised procedure where the number and initial locations of the hidden units are determined based on split of the clusters having overlaps among the classes. The basic idea of the proposed approach is to create class specific homogenous clusters where the corresponding samples are closer to their mean than the means of rival clusters belonging to other class categories. We tested the proposed clustering algorithm along with the RBF network classifier on the Graz02 object database and the ORL face database. The experimental results show that the RBF network classifier performs better when it is initialized with the proposed HC algorithm than an unsupervised k-means algorithm. Moreover, our recognition results exceed the best published results on the Graz02 database and they are comparable to the best results on the ORL face database indicating that the proposed clustering algorithm initializes the hidden unit parameters successfully.

1. Introduction

Artificial neural networks are largely used in applications involving classification or function approximation. Among these networks, the Radial Basis Function (RBF) network is a special type with several distinctive features. A typical RBF neural network classifier has three layers as shown in Fig. 1. The input layer of the network is made of source nodes that connect the coordinates of the input vector to the nodes in the second layer. The second layer, the only hidden layer in the network, includes processing units called the hidden basis function units which are located on the centers of well chosen clusters. Each hidden unit implements a special basis function and this process incorporates the nonlinearity into the RBF network. The output layer is linear, and it produces the predicted class labels based on the response of the hidden units.

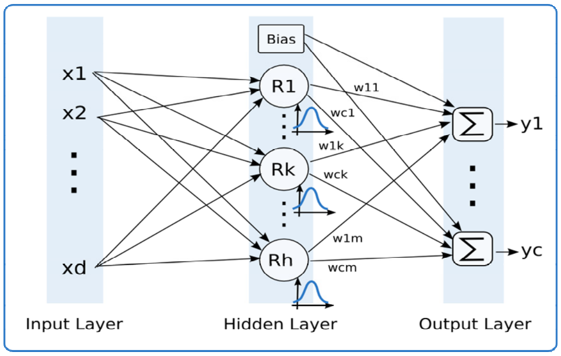

Fig. 1: A Typical RBF Neural Network Classifier System.

Theoretical and empirical studies showed that the performance of the RBF network highly depends on the number and initial locations of the hidden units [1], [2]. However, no well defined procedure exists how to tackle this cumbersome step. Generally, the positions of the hidden units are obtained using unsupervised clustering algorithms such as k-means or Expectation Maximization. In [2], the authors use the extracted support vectors for the initialization. Er e*t al*. [3] use a supervised clustering algorithm for the same goal. The authors split the rival clusters to remove overlaps among them. In this paper, we introduce a variant of this clustering algorithm with a couple of modifications. These modifications yield more compact representations of classes with a smaller number of hidden units. Also, the proposed algorithm easily handles data outliers, which is not addressed in [3].

The rest of the paper is organized as follows: Section 2 explains the issues related to the design of the RBF networks.

In Section 3, we introduce our proposed clustering algorithm. Section 4 describes the data sets and experimental results. Finally, our conclusions are given in Section 5.

2. Design Issues

To use the RBF networks, one has to specify the hidden unit function. Among all possible basis functions, typically Gaussian functions are preferred for classification applications defined as

$$R_j(x) = \exp[-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)] \text{ , } j = 1,...,h, \qquad (1)$$

where x is the input vector and μ_j and Σ_j denote the mean vector and covariance matrix associated to the chosen clusters respectively. In the case of general covariance matrices, the input space is partitioned into a number of hyper-ellipsoids by the corresponding hidden units. On the other hand, in most applications, the covariance matrices are taken as diagonal matrices with the same diagonal entries, which are called the widths of the Gaussian basis functions. In that case, the hidden functions define local hyper-spheres.

The second step towards building an RBF network is to determine the number of hidden units and their initial parameters. This step is extremely important since it has a great impact on the classification performance of the overall network. Although the hidden unit parameters are revised in the subsequent learning phase, poorly chosen parameters may easily lead the network to converge to a local minimum. Furthermore, it will take more iterations to reach a solution in such cases making the learning phase less effective in terms of computational efficiency. Generally, the positions of the hidden units are obtained using unsupervised clustering algorithms and the widths of the Gaussian functions are chosen based on the distances between initial cluster centers. Once the cluster centers and associated parameters are determined, one may proceed to the training phase. In the first stage of the training phase, the hidden unit parameters are modified based on the training data. Following that, the basis function parameters are kept fixed and the weights of the networks are computed. The weights are optimized by minimization of the following criterion

$$F = \min(\frac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{C}[y_c(x_n) - t_n^c]^2) \text{ , } \qquad (2)$$

where N is the total number of samples in the training set, C is the number of classes, $y_c(x_n)$ is the network output for the input sample x_n , and t_n^c is the desired output. The cost function can be written as a quadratic function of the weights, thus the weight vector can be found by fast linear matrix inversion techniques.

3. Homogeneous Clustering Algorithm

As mentioned earlier, the initial positions of the hidden units are usually obtained using unsupervised clustering algorithms such as k-means or Expectation Maximization since these clustering algorithms more or less model the local distributions of the data. There are two main limitations of these conventional clustering algorithms: First, one needs to determine the number of clusters in advance. Second, these clustering algorithms do not use the available class information since they are unsupervised. This is particularly restrictive for classification problems where the class labels are available. Learning without class information may lead to the situation illustrated in Fig. 2-(a). There are two classes represented with blue and red colors in the figure. Notice that one RBF unit is sufficient to model homogenous clusters, however unsupervised clustering algorithm may yield two RBF units for those clusters. Furthermore, the heterogeneous cluster (i.e., the cluster in which there are samples from both classes) is represented with only one RBF unit, which may degrade the generalization performance. All these deficiencies can be overcome by a supervised clustering algorithm as illustrated in Fig. 2-(b).

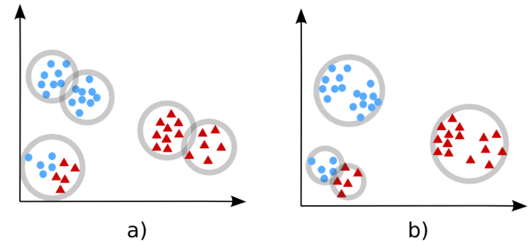

Fig. 2: Clustering algorithms: a) Unsupervised clustering algorithm b) Supervised clustering algorithm.

Here we introduce a variant of the clustering algorithm proposed in [3] to determine the number and initial locations of the hidden units. We use the spherical Gaussian functions since there are not sufficient samples to estimate full covariance matrices in our work. The proposed approach is a supervised procedure where the number and locations of the clusters are determined based on the split of the rival clusters having overlaps among classes. The basic idea of the proposed approach is to create homogenous clusters, thus we will call this approach the homogenous clustering (HC) algorithm. The HC algorithm can be summarized as follows:

i) Initially choose the number of clusters as the number of total classes in our database, i.e., h=C, and set each cluster center to the center of the corresponding class.

ii) For any cluster $H_{m_i}^i$ belonging to the i-th class, compute the Euclidean distance between the furthest point and the mean of that class and assign it to $d_{m_i}^i$. We treat this computed distance as the width of the corresponding hyper-sphere.

iii) Compute the distances between the cluster centers belonging to different classes, that is

$$d_{j,m_j}^{i,m_i} = \| \mu_{m_i}^i - \mu_{m_j}^j \| \text{ , } i,j = 1,...,C, i \neq j \text{ , } \qquad (3)$$

where $\mu_{m_i}^i$ is the mean of $H_{m_i}^i$.

iv) Check the relation between the widths and computed distances among clusters. There are 3 basic scenarios:

1) $d_{j,m_j}^{i,m_i} \geq d_{m_i}^i + d_{m_j}^j$: There is no intersection between the hyper-spheres as shown in Fig. 3-(a). Thus no split is necessary.

2) $\{ d_{j,m_j}^{i,m_i} < d_{m_i}^i + d_{m_j}^j \}$ & $\{ \| d_{m_i}^i - d_{m_j}^j \| < d_{j,m_j}^{i,m_i} \}$: There is intersection between the hyper-spheres and there are 3 possible situations in this case:

a) There are no samples in the intersection of the hyper-spheres as shown in Fig. 3-(b). More formally,

$\{ \| x_{m_i}^i - \mu_{m_i}^i \| < \| x_{m_i}^i - \mu_{m_j}^j \| \}$ & $\{ \| x_{m_j}^j - \mu_{m_j}^j \| < \| x_{mj}^j - \mu_{m_i}^i \| \}$, where $x^i_{m_i} \in H^i_{m_i}$. Thus, no split is necessary since all samples can be classified correctly.

b) $\{ \| x^i_{m_i} - \mu^i_{m_i} \| > \| x^i_{m_i} - \mu^j_{m_j} \| \}$ & $\{ \| x^j_{m_j} - \mu^j_{m_j} \| < \| x^j_{mj} - \mu^i_{m_i} \| \}$: There are samples in the intersection belonging to $H^i_{m_i}$ and they are closer to the center of rival cluster $H^j_{m_j}$. This case is illustrated in Fig. 3-(c). In this case, the samples closer to rival cluster centers in the intersection will be misclassified, thus the cluster $H^i_{m_i}$ must be split into two clusters if the number of the samples in the intersection is bigger than a selected threshold. There may be data outliers in the training data and the threshold is introduced to prevent over-fitting in such cases.

c) $\{ \| x^i_{m_i} - \mu^i_{m_i} \| > \| x^i_{m_i} - \mu^j_{m_j} \| \}$ & $\{ \| x^j_{m_j} - \mu^j_{m_j} \| > \| x^j_{mj} - \mu^i_{m_i} \| \}$: In this case, both clusters have samples in the intersection where the distances from those samples to rival cluster centers are closer than the distances to their own cluster centers. This case is illustrated in Fig. 3-(d). Therefore, both clusters must be split if the number of samples in the intersections is bigger than the selected threshold.

3) $\{ d^{i,m_i}_{j,m_j} < d^i_{m_i} + d^j_{m_j} \}$ & $\{ \| d^i_{m_i} - d^j_{m_j} \| \geq d^{i,m_i}_{j,m_j} \}$: In this case one of the hyper-sphere is completely enclosed in the other as shown in Fig. 3-(e). Thus the bigger hyper-sphere must be split.

After each split, cluster centers in the vicinity of the split cluster (including the center of the split cluster), associated to the class category of the split cluster, must be modified. In the case of data outliers, those outliers must be removed from the training set and corresponding cluster centers must be updated.

v) Repeat (i)-(iv) until no more splits are necessary.

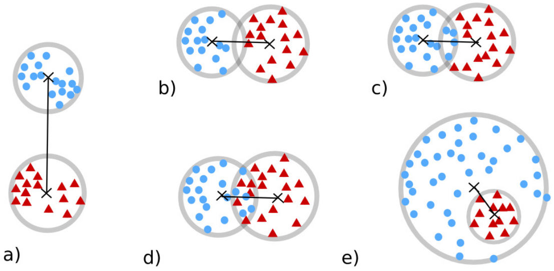

Fig. 3: Possible intersection scenarios between clusters.

In [3], the authors attempt to remove intersections without taking the data outliers into consideration, and they do not revise the cluster centers in the vicinity of the split cluster. Thus, their proposed algorithm tends to yield additional unnecessary clusters.

To determine the initial widths of the basis functions, we considered that they are equal for all Gaussian functions. Then, the widths are fixed to $\sigma_j = 1.5 d_{\min}$, $j = 1,\ldots,h$, where $d_{\min}$ is the minimum distance among all clusters belonging to different classes.

4. Experiments

We tested our proposed clustering approach on the classification of two real data sets, namely the Graz02 object database [4] and the ORL face database [5]. For all experiments we initialized the RBF network classifier with the HC algorithm and the unsupervised k-means clustering algorithm. The number of hidden units is firstly determined by the HC algorithm and then it is used for the initialization of k-means clustering algorithm. Then, recognition rates were obtained for each initialization. Data are standardized before the experiments since we utilized hyper-sphere Gaussian basis functions $R_j(x) = \exp(-\| x - \mu_j \|^2 / \sigma_j^2)$ in this work.

4.1. Experiments on the Graz02 Database

The Graz02 database contains three object categories (bikes, cars, and persons) and counter examples. The goal is to label the test images with either one of the object categories or as being a counter example. Each object class includes samples from various viewpoints under different lighting conditions and scales. In addition, the objects are not always fully visible and there is a large intra-class variation, which makes the classification task even more challenging. Some chosen images from the Graz02 database are shown in Fig. 4.

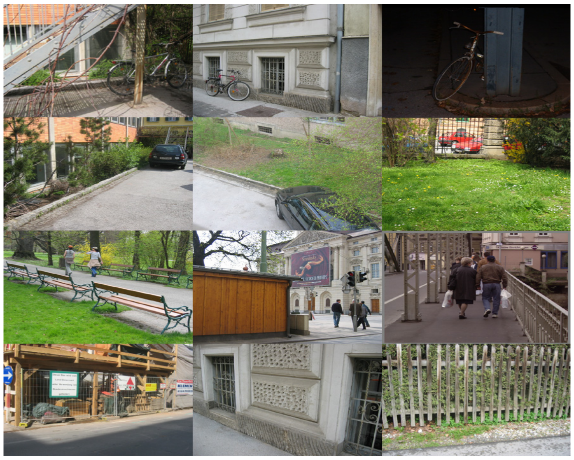

Fig 4: Some image samples from the Graz02 database: The images in the first, second, third, and fourth rows are respectively chosen from the bikes, cars, persons, and counter example categories.

In order to represent the image samples, we used "the bag of features" representation. Introduced by Csurka *et al.* [6], it has been applied widely for both object classification and localization tasks. In this approach, firstly small image patches are chosen at different positions and different scales. Interest point detectors can be employed to choose the patches. In our case, we randomly selected a large number (in the order of thousands) of image patches at different positions and scales. Then, the chosen patches are described by the scale invariant feature transform (SIFT) descriptor, which was reported to yield the best results in the bag of features scheme [6]. Following this process, all descriptors extracted from images are quantized in a discrete set of so called visual words forming a vocabulary. We set the size of the visual vocabulary to 2000. To build image representation, each extracted descriptor is compared to the visual words and associated to the closest word. Based on these assignments, we build histograms, which were used as image feature vectors.

Recently significant improvements were achieved by using the Chi-Square (χ^2) distance (CSD) during classification of image histograms. Therefore, in this study, we also used the generalized Gaussian kernel $R_j(x) = \exp(-\chi^2(x, \mu_j) / \sigma_j^2)$ incorporating χ^2 distance between histograms.

We performed three binary classification tests for each object category where the goal is to decide whether an image includes the object of interest or not. For all tests, we used 150 image histograms of the object category as positive samples and 150 of the counter examples as negative samples following the same experimental setup described in [7]. The Receiver Operating Characteristic (ROC) curves belonging to each classification problem are shown in Fig. 5. We obtained Equal Error Rates (EERs) from these curves and they are given in Table I. In terms of recognition accuracy, the basis functions utilizing the χ^2 distance yielded better results than the classical Gaussian function in the RBF classifier network. For all cases, the RBF classifier initialized with the proposed HC algorithm achieved the best recognition rates, and these rates exceed the best published recognition rates for this experimental setup.

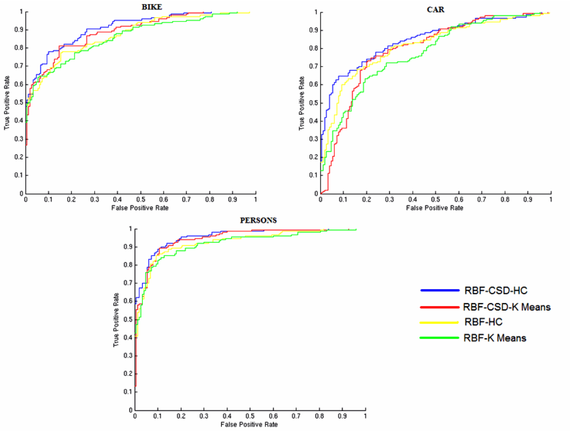

Fig 5: ROC curves belonging to each classification problem.

Table I
Recognition Results (%) in terms of EERs for the Graz02 database: KM - k-means

	RBF		RBF-CSD		Moosman *et al.* [7]	Opelt *et al.* [4]
	HC	KM	HC	KM		
Bike	78	77.3	**82.6**	81.3	79.9	76.5
Cars	74	72.6	**76.6**	75.3	71.7	67
Persons	87.3	85.3	**89.3**	**89.3**	-	81

4.2. Experiments on the ORL Face Database

The ORL face database contains C=40 individuals with 10 images per person. The images are taken at different time instances with different facial expressions, and facial details. All individuals are in upright, frontal position (with tolerance for some side movement). The size of the each image is 92x112 and we used gray scale values as feature vectors representing face images.

We first randomized the samples in the database and then selected 5 samples from each class for training and the remaining 5 samples were used for testing. We did not apply any preprocessing to the images. The dimensionality of the sample space is too high compared to the training set size thus the RBF network classifiers are not suitable for such cases. Therefore we decreased the dimensionality of the sample space to 70 by employing the Orthonormal Discriminant Vector [8] dimensionality reduction technique. This technique is an iterative method where each projection direction is chosen such that it optimizes the well known Fisher's Linear Discriminant analysis criterion in each step. Following the dimensionality reduction, we applied the RBF networks initialized with the HC and k-means clustering methods. This process was repeated 10 times and the final recognition rates were obtained averaging the error rate of each run. The computed recognition rates are given in Table II. To assess the performance, we compared the computed recognition rates to the DCV method which can be considered as the state-of-the-art.

Table II
Recognition Rates and Standard Deviations on the ORL Faces

RBF		**DCV**
HC	k-means	
95.45%, ±1.44	91.90%, ±2.19	95.20%, ±1.49

As can be seen in the table, for the same number of hidden units, the RBF network classifier initialized with our proposed HC algorithm yields significantly better recognition rates than the RBF classifier initialized with the k-means clustering algorithm. Note that the recognition rate obtained by the proposed clustering algorithm is slightly better than the recognition of the DCV method.

5. Conclusion

In this paper, we proposed a new supervised clustering algorithm for finding the number and initial locations of the hidden layer units in the RBF network classifier. We tested the proposed scheme along with the RBF network classifier on two databases. Experimental results show that the RBF network classifier performs better when it is initialized with the proposed HC algorithm rather than an unsupervised k-means algorithm. Moreover, our recognition rates significantly outperform the best published results on the Graz02 object database and produces similar results to the DCV method on the ORL face database indicating that that the proposed clustering algorithm yields a superior initialization for the RBF network classifier.

6. References

[1] Z. Uykan, C. Guzelis, M. E. Celebi, and H. N. Koivo, "Analysis of input-output clustering for determining centers of RBFN," *IEEE Trans. on Neural Networks*, vol. 11, 2000.

[2] B. Scholkopf, K. K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. on Signal Processing*, vol. 45, 1997.

[3] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *IEEE Trans. on Neural Networks*, vol. 13, 2002.

[4] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Trans. on PAMI*, vol. 28, 2006.

[5] The ORL Database of Faces, AT&T Laboratories_Cambridge. Available: http://www.uk.research.att.com/face data-base.html.

[6] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray, "Visual categorization with bag of keypoints," *in EECV'04 Workshop on Statistical Learning in Computer Vision*, pp. 59-74, 2004.

[7] F. Moosmann, D. Larlus, and F. Jurie, "Learning saliency maps for object categorization," *in ECCV International Workshop on The Representation and Use of Prior Knowledge in Vision*, 2006.

[8] T. Okada and S. Tomita, "An optimal orthonormal system for discriminant analysis," *Journal of Pattern Recognition,* 18, pp. 139-144, 1985.