International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

Locating Center Points for Radial Basis Function Networks Using Instance Reduction Techniques

Rana Yousef[1] and Khalil el Hindi[2]

Abstract**—**The behavior of Radial Basis Function (RBF) Networks greatly depends on how the center points of the basis functions are selected. In this work we investigate the use of instance reduction techniques, originally developed to reduce the storage requirements of instance based learners, for this purpose. Five Instance-Based Reduction Techniques were used to determine the set of center points, and RBF networks were trained using these sets of centers. The performance of the RBF networks is studied in terms of classification accuracy and training time. The results obtained were compared with two Radial Basis Function Networks: RBF networks that use all instances of the training set as center points (RBF-ALL) and Probabilistic Neural Networks (PNN). The former achieves high classification accuracies and the latter requires smaller training time. Results showed that RBF networks trained using sets of centers located by noise-filtering techniques (ALLKNN and ENN) rather than pure reduction techniques produce the best results in terms of classification accuracy. The results show that these networks require smaller training time than that of RBF-ALL and higher classification accuracy than that of PNN. Thus, using ALLKNN and ENN to select center points gives better combination of classification accuracy and training time. Our experiments also show that using the reduced sets to train the networks is beneficial especially in the presence of noise in the original training sets.

Keywords**—**Radial basis function networks, Instance-based reduction, PNN.

I. Introduction

The goal of this work is to study the effect of using instance reduction techniques to choose center points for RBF networks with respect to classification accuracy and training time. Several Instance-Based Reduction Techniques [5] will be used to find a subset of the training set to act as center points of RBF networks. By doing so, we hope to achieve good classification accuracies and, at the same time, reduce the time needed to train RBF networks. We compared our technique with two RBF networks: RBF networks that use all instances of the training set as center points (RBF-ALL) and Probabilistic Neural Network (PNN). In another session of experiments we also used the reduced sets to train the RBF networks.

Twenty datasets obtained from the Machine Learning Repository at University of California Irvine (UCI) were used to compare the different methods of locating center points. The following sections present the results obtained from the experiments mentioned.

Section 2 presents reduction techniques that are used in this research. Section 3 gives a brief overview of the Radial Basis Function Networks. Sections 4 and 5 describe the results obtained from our experiments. Section 6 is the conclusion section.

II. Instance Reduction Techniqes

Many reduction techniques were designed for instance-based learning algorithms to decide what instances to store for use during generalization in order to avoid excessive storage and time complexity, and possibly to improve classification accuracy by eliminating noise. The techniques that are used in this research were chosen carefully to cover the wide spectrum of reduction techniques from noise filtering (ENN and ALKNN) to pure reduction techniques (EXPLORE), in addition to those that combine between the two (DROP2 and DROP5). See [5] for an excellent survey of these and other techniques.

Table I shows the average training sets' sizes after applying the chosen reduction techniques. The sizes are shown as proportions to the size of the original training sets in case of noise free data, and when adding 5%, 10% and 15% noise.

As can be seen from table I the reduction algorithms ENN and ALLKNN have the highest storage requirements. ENN and ALLKNN are considered to be noise filtering techniques rather than pure reduction techniques [5], so they only remove noisy instances from the training set and keeps the rest of the instances, that's why the amount of reduction achieved increases when noise increases.

[1]KASIT, University of Jordan, Amman, Jordan. Email: rana@ju.edu.jo
[2]KASIT, University of Jordan, Amman, Jordan. Email: hindi@ju.edu.jo

TABLE I
AVERAGE TRAINING SETS' SIZES AS A RATIO OF THE ORIGINAL TRAINING SETS IN CASE OF NOISE FREE DATA, AND WHEN ADDING 5%, 10% AND 15% NOISE

Algorithm	Noise free	5% noise	10% noise	15% noise
ENN	76.21	72.75	68.74	65.29
ALLKNN	70.02	64.44	58.34	53.59
DROP2	17.84	19.47	21.26	23.30
DROP5	13.60	14.00	14.40	15.49
EXPLORE	1.06	1.07	1.06	1.07

III. RADIAL BASIS FUNCTION NETWORKS

Learning with Radial Basis Functions is one approach to function approximation that is closely related to distance-weighted regression and also to artificial neural networks [1, 3]. The learned hypothesis is a function of the form:

$$f^*(x) = w_0 + \sum_{u=1}^{k} w_u K_u(d(x_u, x)) \quad (1)$$

where w_0 is the bias unit, each x_u is an instance from X, the set of training instances, and where the kernel function $K_u(d(x_u, x))$ is defined so that it decreases as the distance $d(x_u, x)$ increases. k is a user-provided constant that specifies the number of kernel functions to be included. Even though *f*(x)* is a global approximation of *f(x)*, the contribution from each of the $K_u(d(x_u, x))$ terms is localized to a region surrounding the point x_u. It is common to choose each function $K_u(d(x_u, x))$ to be a Gaussian function centered at point x_u with some variance σ_u^2:

$$K_u(d(x_u, x)) = e^{\frac{1}{2\sigma_u^2} d^2(x_u, x)} \quad (2)$$

The function given by Equation 1 can be viewed as describing a two-layer network where the first layer of units computes the values of the various $K_u(d(x_u, x))$ and each of the hidden units produces an activation determined by a Gaussian function centered at some instance x_u. Therefore, its activation will be close to zero unless the input *x* is near x_u. Then the second layer computes a linear combination of these first-layer unit values [2, 3].

Probabilistic Neural Network Architecture:

In PNN the first layer computes distances from the input vector to the training input vectors, and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities. Finally, a "compete" transfer function on the output of the second layer picks the maximum of these probabilities, and produces a 1 for that class and a 0 for the other classes [4].

IV. TRAINING RBF NETWORKS USING CENTERS LOCATED BY INSTANCE REDUCTION TECHNIQES

Table II shows the average classification accuracy for the 20 datasets and the average training time as a proportion to the training time required by RBF-ALL, these results are shown in case of noise free data, and when adding 5%, 10% and 15% noise. The first column in the table shows the name of the technique used to locate the center points prefixed with RBF'.

As can be seen from table II, among all reduction techniques, ENN and ALLKNN have the highest classification accuracies. This is expected since ENN and ALLKNN are noise filters, and retain most of the instances in the training set, they only edit out noisy instances and retain the center points. A good RBF network is the one which uses a set of centers that is noise-free and is a good representative of the training set. The two algorithms can be used to locate centers having these two important characteristics.

The most important achievement is the reduction in training time. In all cases, training time was reduced compared to the time needed to train RBF-ALL. On average, RBF-ALLKNN needs only half the training time needed by RBF-ALL, and RBF-ENN needs about 60% of the training time required by RBF-ALL, i.e. there is a substantial reduction in time.

Comparing results with PNN

The experiments showed that using noise free datasets, PNN has an average of 76.01% classification accuracy and needs 1.5% of RBF-ALL training time. The classification accuracy for RBF-ENN and RBF-ALLKNN is much better than that of PNN, but the training time is greater. PNN uses all instances in the training set as a weight vector for the first layer and tries to learn the center points by adjusting these weights. From the results, it is apparent that learning the center points is fast but this is at the expense of the classification accuracy.

TABLE II
AVERAGE CLASSIFICATION ACCURACY (ACC) AND TRAINING TIME (T) AS A RATIO OF THE TRAINING TIME OF RBF-ALL FOR THE 20 DATASETS IN CASE OF NOISE FREE DATA, AND WHEN ADDING 5%, 10% AND 15% NOISE

Algorithm	Noise free		5% noise		10% noise		15% noise	
	Acc	T(%)	Acc	T(%)	Acc	T(%)	Acc	T(%)
RBF-ALL	81.47	100	78.66	100	75.09	100	72.11	100
RBF-ENN	81.17	60.53	78.45	53.82	75.08	47.84	71.88	44.66
RBF-ALLKNN	81.21	49.60	78.08	42.16	75.68	34.39	71.99	29.75
RBF-DROP2	54.59	11.35	75.15	10.14	71.35	10.97	53.38	14.60
RBF-DROP5	74.45	9.47	73.39	7.81	70.38	8.30	69.69	11.12
RBF-EXPLORE	74.64	0.73	54.33	0.64	54.36	0.64	67.69	0.78

V. TRAINING RBF NETWORKS USING REDUCED TRAINING SETS

In the previous session of experiments, the reduction techniques were used to locate the set of center points of an RBF network. As we saw from the results, using these reduced sets to locate the center points caused a substantial reduction in the training time, but there was a reduction in classification accuracy with a minimal reduction when ENN and ALLKNN algorithms were used. The reason for this is that these two algorithms are noise-filtering techniques that remove noisy instances. In the following session of experiments we used the reduced sets not only as center points but also to train the RBF networks.

Table III gives the results obtained from such experiments. The experiments were performed on the original training sets, then they were performed on noisy training sets, noise was added artificially with percentages equal to 5%, 10% and 15%. The table shows the names of the reduction techniques used prefixed with RBF-T' .

As can be seen from table III, there's a substantial improvement in classification accuracy when RBF networks are trained using a reduced set generated by ENN and ALLKNN, especially in the presence of noise. Using all instances in the training set to train the RBF network is vulnerable to overfitting. Hence, the misclassification occurs, especially in the presence of noise. Since ENN and ALLKNN are noise filtering techniques, RBF networks trained using reduced sets generated by these reduction techniques have the highest classification accuracies compared to RBF-ALL and all other reduction techniques.

In all cases, the training time was considerably reduced compared to the time needed to train RBF-ALL. This is because there is a reduction in storage requirement. It is obvious from table III that RBF-T-ENN and RBF-T-ALLKNN achieve the best combination of classification accuracy and training time.

VI. CONCLUSION

In this work, several techniques for locating the centers of Radial Basis Function Networks were examined. These techniques are ENN, ALLKNN, EXPLORE, DROP2 and DROP5 which represent a sample of instance reduction techniques. This sample of techniques was chosen carefully to represent the wide spectrum of techniques.

The performance, in terms of classification accuracy and training time, for RBF networks trained using these reduction techniques was compared with two extremes: RBF-ALL and PNN. The former achieves high classification accuracies and the latter requires smaller training time. Results showed that RBF networks trained using sets of centers located by noise-filtering techniques (ALLKNN and ENN) rather than pure reduction techniques produce the best results in terms of classification accuracy.

Our results also show that using noise filtering techniques to determine the training set as well as the center points substantially improves the classification accuracy.

TABLE III
AVERAGE CLASSIFICATION ACCURACY (ACC) AND TRAINING TIME (T) AS A RATIO OF THE TRAINING TIME OF RBF-ALL FOR THE 20 DATASETS IN CASE OF NOISE FREE DATA, AND WHEN ADDING 5%, 10% AND 15% NOISE

Algorithm	Noise free		5% noise		10% noise		15% noise	
	Acc	T(%)	Acc	T(%)	Acc	T(%)	Acc	T(%)
RBF-ALL	81.47	100	78.66	100	75.09	100	72.11	100
RBF-T-ENN	82.16	55.48	81.19	49.6	80.67	41.63	79.56	38.98
RBF- T-ALLKNN	81.74	45.69	81.56	37.9	81.41	31.45	80.08	27.77
RBF- T-DROP2	68.57	8.9	68.66	6.69	67.87	7.82	68.68	8.95
RBF- T-DROP5	69.76	8.54	67.71	8.04	66.04	8.33	64.66	8.27
RBF- T-EXPLORE	53.27	3.62	55.64	3.82	55.67	3.24	54.65	3.87

REFERENCES

[1] Bors, A.G. (2001). Introduction of the Radial Basis Function (RBF) Networks. Online Symposium for Electronics Engineers, 1(1), 1 - 7.
[2] Ghosh, J. and Nag, A., (2002). Knowledge enhancement and reuse with radial basis function networks, IJCNN '02. Proceedings of the 2002 International Joint Conference on Neural Networks, Vol. 2, 1322 -1327
[3] Mitchell, T. (1997). Machine Learning. New York: McGraw-Hill
[4] Wasserman, P.D. (1993). Advanced Methods in Neural Computing, New York: Van Nostrand Reinhold, 35-55.
[5] Wilson, D. and Martinez, T., (2000). Reduction Techniques for Instance-Based Learning Algorithms, Machine Learning, 38(3), 257-286.