Proceedings of the International Conference on Neural Networks (ICNN'96), vol. 2, pp. 1263-1267, June 1996.

Heterogeneous Radial Basis Function Networks

D. Randall Wilson, Tony R. Martinez
e-mail: *randy@axon.cs.byu.edu*, *martinez@cs.byu.edu*
Computer Science Department, Brigham Young University, Provo, UT 84602, U.S.A.

ABSTRACT

Radial Basis Function (RBF) networks typically use a distance function designed for numeric attributes, such as Euclidean or city-block distance. This paper presents a heterogeneous distance function which is appropriate for applications with symbolic attributes, numeric attributes, or both. Empirical results on 30 data sets indicate that the heterogeneous distance metric yields significantly improved generalization accuracy over Euclidean distance in most cases involving symbolic attributes.

1. Introduction

Much research has been directed at finding better ways of helping machines learn from examples. When domain knowledge in a particular area is weak, solutions can be expensive, time consuming and even impossible to derive using traditional programming techniques.

In such cases, neural networks can be used as tools to make reasonable solutions possible or good solutions more economical. Such an automated solution is often more accurate than a hard-coded program, because it learns from actual data instead of making assumptions about the problem. It often can adapt as the domain changes and often takes less time to find a good solution than a programmer would. In addition, inductive learning solutions may generalize well to unforeseen circumstances.

Radial Basis Function (RBF) networks [1][13][15] have received much attention recently because they provide accurate generalization on a wide range of applications, yet can often be trained orders of magnitude faster [7] than other models such as backpropagation neural networks [8] or genetic algorithms [9].

Radial basis function networks make use of a distance function to find out how different two input vectors are (one being presented to the network and the other stored in a hidden node). This distance function is typically designed for numeric attributes only and is inappropriate for nominal (unordered symbolic) attributes.

This paper introduces a heterogeneous distance function which allows radial basis function networks to appropriately handle applications that contain nominal attributes, numeric attributes, or both. Section 2 introduces the basic radial basis function network that will be used to demonstrate the usefulness of the heterogeneous distance function. Section 3 introduces the distance function itself. Section 4 presents empirical results which indicate that in most cases the heterogeneous distance function significantly improves generalization over Euclidean distance when symbolic attributes are present and never reduces accuracy in completely numeric domains. Section 5 presents conclusions and future research areas.

2. Radial Basis Function Network

This section presents a radial basis function (RBF) network that is used as a probabilistic neural network (PNN) [10] for classification. The distance function presented in this paper could be appropriately used on many different kinds of basis-function networks, so this particular network is just one example of its use. This network was chosen because of its simplicity, which helps to focus on the new distance function instead of on other factors.

The network learns from a *training set* T, which is a collection of examples called *instances*. Each instance i has an input vector $\boldsymbol{y}_i$, and an output class, denoted as $class_i$. During execution, the network receives additional input vectors, denoted as $\boldsymbol{x}$, and outputs the class that $\boldsymbol{x}$ seems most likely to belong to.

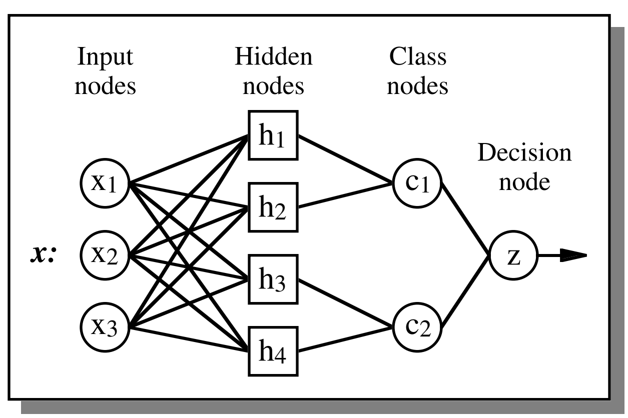

Figure 1. Radial Basis Function Network.

The probabilistic neural network is shown in Figure 1. The first (leftmost) layer contains input nodes, each of which receives an input value from the corresponding element in the input vector $\boldsymbol{x}$. Thus, for an application with m input attributes, there are m input nodes. All connections in the network have a weight of 1. In essence, this means that the input vector is passed directly to each hidden node.

There is one hidden node for each training instance i in the training set. Each hidden node h_i has a center point $\boldsymbol{y}_i$ associated with it, which is the input vector of instance i. A hidden node also has a value σ_i which determines the size of its *receptive field*. This value is set to the distance of the nearest neighbor of i in the training set, using the same distance function as that used during execution.

A hidden node receives an input vector $\boldsymbol{x}$ and outputs an activation given by the function:

$$g(\boldsymbol{x}, \boldsymbol{y}_i, \sigma_i) = \exp[-D^2(\boldsymbol{x}, \boldsymbol{y}_i) / 2\sigma_i^2] \tag{1}$$

where D is a distance function such as Euclidean distance or the heterogeneous distance function that will be discussed in Section 3. This function g is a Gaussian function which returns a value of 1 if $\boldsymbol{x}$ and $\boldsymbol{y}_i$ are equal, and drops to an insignificant value as the distance grows.

Each hidden node h_i is connected to a single class node. If the output class of instance i is j, then h_i is connected to class node c_j. Each *class node* c_j computes the sum of the activations of the hidden nodes that are connected to it (i.e., all the hidden nodes for a particular class) and passes this sum to a decision node. The *decision node* outputs the class with the highest summed activation.

One of the greatest advantages of this network is that it does not require any iterative training. One disadvantage of this network is that it has one hidden node for each training instance and thus requires more computational resources during execution than many other models. In addition, it does not iteratively train weights on any of the connections, which can make its generalization less flexible.

3. Heterogeneous Distance Function

In Section 2, a probabilistic neural network was presented using radial basis functions and a simple weighting scheme that avoided iterative training. In this section, several alternatives for the distance function D are defined, including a new heterogeneous distance function H.

Radial basis functions typically use the *Euclidean distance* function:

$$E(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{2}$$

where m is the number of input variables (attributes) in the application. An alternative function, the *city-block* or *Manhattan* distance function, uses less computation and often does not significantly change the results [10]. It is defined as:

$$M(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{m} |x_i - y_i| \tag{3}$$

One problem with both of these distance functions is that they assume that the input variables are linear. However, there are many applications that have *nominal* attributes. A nominal attribute is one with a discrete set of attribute values that are unordered. For example, a variable representing *symptoms* might have possible values of *headache, sore throat, chest pains, stomach pains, ear ache,* and *blurry vision*. Using a linear distance measurement on such values makes little sense in this case, because numbers assigned to the values are in an arbitrary order. In such cases a distance function is needed that handles nominal inputs appropriately.

Stanfill & Waltz [11] introduced the *value difference metric* (VDM) which has been used as the basis of several distance functions in the area of machine learning [2][3][6]. Using VDM, the distance between two values x and y of a single attribute a is given as:

$$vdm_a(x, y) = \sum_{c=1}^{C} \left(\frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right)^2 \tag{4}$$

where $N_{a,x}$ is the number of times attribute a had value x; $N_{a,x,c}$ is the number of times attribute a had value x and the output class was c; and C is the number of output classes. Using this distance measure, two values are considered to be closer if they have more similar classifications, regardless of the order of the values.

Some models that have used the VDM or extensions of it (notably PEBLS [6]) have discretized continuous attributes into a somewhat arbitrary number of discrete ranges and then treated these values as nominal values. Discretization throws away much of the information available to the learning model and often reduces generalization accuracy [12].

The Heterogeneous Radial Basis Function (HRBF) model presented in this paper makes use of a new distance function that uses the above part of the VDM as a building block. In the HRBF model, the heterogeneous distance H between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is given as:

$$H(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{a=1}^{m} d_a{}^2(x_a, y_a)} \tag{5}$$

$$d_a(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown; otherwise...} \\ normalized\_vdm_a(x,y), & \text{if } a \text{ is nominal} \\ normalized\_diff_a(x,y), & \text{if } a \text{ is numeric} \end{cases} \tag{6}$$

where m is the number of attributes. The function $d_a(x,y)$ returns a distance between the two attribute values x and y using one of two functions (defined below), depending on whether the attribute is nominal or numeric. Many data sets contain unknown input values which must be handled appropriately in a practical system. The function $d_a(x,y)$ therefore returns a distance of 1 if either x or y is unknown. Other more complicated methods have been tried, as in [14], but with little effect on accuracy. The function H is similar to that used in [4], except that it uses VDM instead of an overlap metric for nominal values and normalizes differently.

One weakness of the basic Euclidean and Manhattan distance functions is that if one of the input variables has a relatively large range, then it can overpower the other input variables. For example, suppose an application has just two input attributes, f and g. If f can have values from 1 to 1000 and g has values only from 1 to 10, then g's influence on the distance function will usually be overpowered by f's influence.

Therefore, distances are often *normalized* by dividing the distance for each variable by the range of that attribute, so that the distance for each input variable is in the range 0..1. However, this allows outliers (extreme values) to have a profound effect on the contribution of an attribute. For example, if a variable has values which are in the range 0..10 in almost every case but with one (possibly erroneous) value of 50, then dividing by the range would almost always result in a value less than 0.2. A more robust alternative is to divide the values by the standard deviation in order to reduce the effect of extreme values on the typical cases.

In the heterogeneous distance metric, the situation is more complicated because the nominal and numeric distance values come from different types of measurements. It is therefore necessary to find a way to scale these two different measurements into approximately the same range in order to give each variable a similar influence on the overall distance measurement.

Since 95% of the values in a normal distribution fall within two standard deviations of the mean, the difference between numeric values is divided by 4 standard deviations in order to scale each value into a range that is usually of width 1.0.

Using VDM, the average value for $N_{a,x,c}/N_{a,x}$ (as well as for $N_{a,y,c}/N_{a,y}$) is $1/C$. Since the difference is squared and then added C times, the sum is usually in the neighborhood of $C(1/C^2)=1/C$. This sum is therefore multiplied by C to get it in the range 0..1, making it roughly equal in influence to normalized numeric values.

The functions *normalized_vdm* and *normalized_diff* are thus defined as:

$$normalized\_vdm_a(x,y) = \sqrt{C * \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^2} \tag{7}$$

$$normalized\_diff_a(x,y) = \frac{|x-y|}{4\sigma_a} \tag{8}$$

where C is the number of classes, N is defined as in (4), and σ_a is the standard deviation of the numeric values of attribute a. Note that in practice the square root in (7) is not performed since the squared attribute distances are needed in (5) to compute H. Similarly, the square root in (5) is not typically performed in computing H, since the squared distance (H^2 instead of D^2 in this case) is used in (1) to compute g, the activation of a hidden node.

4. Empirical Results

The Heterogeneous Radial Basis Function (HRBF) algorithm was implemented and tested on several databases from the Machine Learning Database Repository at the University of California Irvine [5].

Each test consisted of ten trials, each using one of ten partitions of the data randomly selected from the data sets, i.e., 10-fold cross-validation. Each trial consisted of building a network using 90% of the training instances in hidden nodes and then using this network to classify the remaining 10% of the instances to see how many were classified correctly.

In order to see what effect the new heterogeneous distance function has on accuracy, a homogeneous version of the algorithm was implemented as well, which is exactly the same as HRBF, except that it uses a normalized Euclidean distance function. This is accomplished by using $normalized\_diff_a(x,y)$ instead of $normalized\_vdm_a(x,y)$ in (6) for nominal as well as for numeric attributes. The homogeneous algorithm will be referred to as the *default* algorithm, or simply *RBF*. Both algorithms used the same training sets and test sets for each trial.

The average accuracy for each database over all trials is shown in Figure 2. A bold value indicates which value was highest for each database. One asterisk (*) indicates that the higher value is statistically significantly higher at a 90% confidence level, using a one-tailed paired *t*-test. Two asterisks (**) are used to mark differences that are significant at a 95% or higher confidence interval.

Figure 2 also lists the number of continuous and nominal input attributes for each database. Note that the accuracy for every application that has only numeric attributes is exactly the same for both RBF and HRBF. This is no surprise, since the distance functions are equivalent on numeric attributes.

However, on the databases that have some or all nominal attributes, HRBF obtained higher generalization accuracy than RBF in 12 out of 23 cases, 10 of which were significant at the 95% level or above. RBF had a higher accuracy in only four cases, and only one of those (the *Zoo* data set) had a difference that was statistically significant.

It is interesting to note that in the *Zoo* data set, 15 out of 16 of the attributes are boolean, and the remaining attribute, while not linear, is actually an ordered attribute. These attributes are tagged as nominal, but the Euclidean distance function is appropriate for them as well.

In all, HRBF performed as well or better than the default algorithm in 26 out of 30 cases.

The above results indicate that the heterogeneous distance function is typically more appropriate than the Euclidean distance function for applications with one or more nominal attributes, and is equivalent to it for domains without nominal attributes.

Database	RBF (Euclidean)	HRBF	Numeric Attributes	Nominal Attributes
Annealing	**76.19**	76.06	9	29
Audiology	36.00	**54.00****	0	69
Australian-Credit	80.14	**83.77****	6	8
Bridges	52.36	**55.27***	4	7
Credit-Screening	75.36	**83.48****	6	9
DNA Promoters	54.27	**76.91****	0	57
Echocardiogram	78.04	**79.46**	7	2
Flags	45.74	**57.11****	10	18
Hayes-Roth	52.17	**65.96****	0	4
Heart	**80.74**	80.00	7	6
Heart-Disease (Hungarian)	64.00	**74.92****	7	6
Heart-Disease (More)	45.95	45.95	7	6
Heart-Disease (Swiss)	38.85	38.85	7	6
Hepatitis	79.33	79.33	6	13
Horse-Colic	67.09	67.09	7	16
House-Votes-84	69.22	**79.77****	0	16
Image Segmentation	80.48	80.48	18	1
Solar-Flare 1	**81.71**	81.41	1	9
Solar-Flare 2	99.53	99.53	1	11
Soybean-Large	13.01	**35.10****	6	29
Thyroid-Disease (Euthyroid)	90.74	90.74	7	18
Tic-Tac-Toe	65.78	**79.74****	0	9
Zoo	**78.89****	73.33	0	16
Average:	65.46	**71.23****		
Numeric Databases				
Breast-Cancer-Wisconsin	97.00	97.00	9	0
Liver-Disorders	62.50	62.50	6	0
Iris	94.00	94.00	4	0
Pima-Indians-Diabetes	76.30	76.30	8	0
Sat.Test	85.65	85.65	36	0
Vowel	92.01	92.01	10	0
Wine	94.38	94.38	13	0

Figure 2. Comparative experimental results of RBF and HRBF.

5. Conclusions & Future Research

The Heterogeneous Radial Basis Function (HRBF) network uses a normalized heterogeneous distance function which is typically more appropriate than the Euclidean distance function for applications that have at least some nominal or symbolic attributes. By using a more appropriate distance function, higher generalization accuracy can be obtained on most typical heterogeneous or completely symbolic domains. Furthermore, the heterogeneous distance function is equivalent to a normalized Euclidean distance function in completely numeric domains so generalization accuracy will be identical in those domains as well.

In this paper the heterogeneous distance function was used with a probabilistic neural network for classification, which allowed very fast training at the cost of a large, static network. However, this function is appropriate for a wide range of basis function networks that use distance functions.

Current research is seeking to test the heterogeneous distance function on a variety of other models, including various Radial Basis Function networks and instance-based machine learning systems. The normalization factors are also being examined to see if they provide the best possible normalization.

In addition, it appears that some data which is tagged as "nominal" is often somewhat ordered. It is hypothesized that if the values of nominal attributes are randomly rearranged then the HRBF would perform about the same (since it does not depend on the ordering of nominal values), but that the homogeneous RBF would suffer a loss in accuracy. The accuracy of this hypothesis and the severity of the loss in accuracy are currently being explored.

The results of this research are encouraging, and show that heterogeneous distance functions can be used to apply basis function networks to a wider variety of applications and achieve higher generalization accuracy than the homogeneous distance functions used in the past.

References

[1] Chen, S., C. R. N. Cowen, and P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302-309, 1991.

[2] Cost, Scott, and Steven Salzberg, "A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features," *Machine Learning*, vol. 10, pp. 57-78, 1993.

[3] Domingos, Pedro, "Rule Induction and Instance-Based Learning: A Unified Approach," to appear in *The 1995 International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995.

[4] Giraud-Carrier, Christophe, and Tony Martinez, "An Efficient Metric for Heterogeneous Inductive Learning Applications in the Attribute-Value Language," *Intelligent Systems*, pp. 341-350, 1995.

[5] Murphy, P. M., and D. W. Aha, *UCI Repository of Machine Learning Databases*. Irvine, CA: University of California Irvine, Department of Information and Computer Science. Internet: ftp://ftp.ics.uci.edu/pub/machine-learning-databases, 1993.

[6] Rachlin, John, Simon Kasif, Steven Salzberg, David W. Aha, "Towards a Better Understanding of Memory-Based and Bayesian Classifiers," in *Proceedings of the Eleventh International Machine Learning Conference,* New Brunswick, NJ: Morgan Kaufmann, pp. 242-250, 1994.

[7] Renals, Steve, and Richard Rohwer, "Phoneme Classification Experiments Using Radial Basis Functions," *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'89)*, vol. 1, pp. 461-467, 1989.

[8] Rumelhart, D. E., and J. L. McClelland, *Parallel Distributed Processing*, MIT Press, 1986.

[9] Spears, William M., Kenneth A. De Jong, Thomas Bäck, David B. Fogel, and Hugo de Garis, "An Overview of Evolutionary Computation," *Proceedings of the European Conference on Machine Learning*, vol. 667, pp. 442-459, 1993.

[10] Specht, Donald F., (1992). "Enhancements to Probabilistic Neural Networks," in *Proceedings International Joint Conference on Neural Networks (IJCNN '92)*, vol. 1, pp. 761-768.

[11] Stanfill, C., and D. Waltz, "Toward memory-based reasoning," *Communications of the ACM*, vol. 29, December 1986, pp. 1213-1228, 1986.

[12] Ventura, Dan, and Tony Martinez, "An Empirical Comparison of Discretization Models," *Proceedings of the 10th International Symposium on Computer and Information Sciences*, pp. 443-450, 1995.

[13] Wasserman, Philip D., *Advanced Methods in Neural Computing*, New York, NY: Van Nostrand Reinhold, pp. 147-176, 1993.

[14] Wilson, D. Randall, *Prototype Styles of Generalization*, Master's Thesis, Brigham Young University, 1994.

[15] Wong, Yiu-fai, "How Gaussian Radial Basis Functions Work," *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'91)*, vol. 2, pp. 133-138, 1991.