Text Classification by Combining Different Distance Functions with Weights

Takahiro Yamada, Kyohei Yamashita, Naohiro Ishii
Aichi Institute of Technology
1247 Yachigusa, Yakusacho, Toyota, Japan 470-0392
v06723vv@aitech.ac.jp, ishii@aitech.ac.jp

Kazunori Iwata
Aichi University
370 Miyoshicho, Nishikamogun, Aichi Pref.,Japan 470-0296
kazunori@vega.aichi-u.ac.jp

Abstract

Since data is becoming greatly large in the networks, the machine classification of the text data, is not easy under these computing circumstances. Though the k-nearest neighbor (kNN) classification is a simple and effective classification approach, the improving performance of the classifier is still attractive to cope with the high accuracy processing. In this paper, the kNN is improved by applying the different distance functions with weights to measure data from the multi-view points. Then, the weights for the optimization, are computed by the genetic algorithms. After the learning of the trained data, the unknown data is classified by combining the multiple distance functions and ensemble computations of the kNN. In this paper we present a new approach to combine multiple kNN classifiers based on different distance functions, which improve the performance of the k-nearest neighbor method. The proposed combining algorithm shows the higher generalization accuracy when compared to other conventional learning algorithms.

1. Introduction

Recently, broad band networks have made a great progress in information technology for much data communication and transactions in the computer networks. Computer systems also realize the integrated processing of the bulk data by highly speedy technologies. Though the data is becoming greatly large in the volume, the machine classification of the data as text data, is not easy under these computing circumstances.

The kNN[2] is one of the most common instance-based learning algorithms. To classify an unknown object, it ranks the object's neighbors among the training data and then uses the class labels of the k nearest neighbors to predict the class of the new object. Despite its simplicity, the kNN has many advantages over other methods. For example, it provides good generalization accuracy for a variety of real-world classification tasks and applications. Researchers have begun paying attention to combining a set of individual classifiers, also known as a multiple model or ensemble approach, with the hope of improving the overall classification accuracy. Bay[1] has proposed MFS, a method of combining kNN classifiers using multiple features subsets. Each individual classifier in MFS can access all the patterns in the original training set, but only to a random subset of features. MFS uses a simple voting from the output of each classifier to decide the final result. In this paper, we present a new approach to combine multiple kNN classifiers based on different distance functions with weight and ensemble method.

2. Classification by Multiple Distance Functions

2.1. Distance Functions

The choice of distance function influences the bias of the k-nearest neighbor(kNN) classification. There are many distance functions that have been proposed for the kNN classification[11]. We propose a method to combine kNN classifiers based on different distance functions with weights as shown in Fig.1. The method is called here DkNN. Some functions work well for numerical attributes but do not appropriately handle nominal (i.e. discrete, and perhaps unordered) attributes. Some work well for nominal attributes. The most commonly used function is the

Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06)
0-7695-2611-X/06 $20.00 © 2006 IEEE

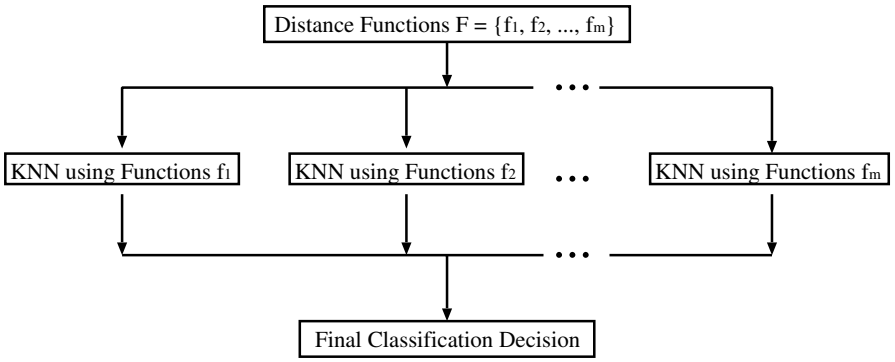

Figure 1. Multiple distance functions for the DkNN method

Euclidean Distance function (Euclid), which is defined as:

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

where x and y are two input vectors (one typically being from a stored instance, and the other an input vector to be classified) and m is the number of input variables (attributes) in the application.

One way to handle applications with both continuous and nominal attributes is to use a heterogeneous distance function that uses different attribute distance functions on different kinds of attributes. The Heterogeneous Euclidean-Overlap Metric (HEOM) uses the overlap metric for nominal attributes and normalized Euclidean distance for linear attributes. This function defines the distance between two values x and y of a given attribute a as:

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a(x_a - y_a)^2}$$

,where

$$d_a(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown,} \\ overlap(x,y) & \text{else if } a \text{ is nominal,} \\ \frac{|x-y}{man_x - main_a}, & \text{otherwise} \end{cases}$$

and function overlap is defined as:

$$overlap(x,y) = \begin{cases} 0, & x = y \\ 1, & \text{otherwise} \end{cases}$$

The Value Difference Metric (VDM), introduced by Stanfill and Waltz (1986)[10], is an appropriate distance function for nominal attributes. A simplified version of the VDM (without the weighting schemes) defines the distance between two values x and y of an attribute a as:

$$\begin{aligned} udm_a(x,y) &= \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q \\ &= \sum_{c=1}^{C} |P_{a,x,c} - P_{a,y,c}|^q \end{aligned}$$

where $N_{a,x}$ is the number of instances in the training set T that have value x for attribute a; $N_{a,x,c}$ is the number of instances in T that have value x for attribute a and output class c; C is the number of output classes in the problem domain; q is a constant, usually 1 or 2; and $P_{a,x,c}$ is the conditional probability that the output class is c given that attribute a has the value x, i.e., $P(c|x_a)$. The $P_{a,x,c}$ is defined as:

$$P_{a,x,c} = \frac{N_{a,x,c}}{N_{a,x}}$$

where $N_{a,x}$ is the sum of $N_{a,x,c}$ over all classes, i.e.,

$$N_{a,x} = \sum_{c=1}^{C} N_{a,x,c}$$

and the sum of $P_{a,x,c}$ over all C classes is 1 for a fixed value of a and x.

In [4], Wilson and Martinez proposed three new alternatives to overcome the weakness of VDM. The one is a Heterogeneous Value Difference Metric(HVDM) that uses Euclidean distance for linear attributes and VDM for nominal attributes. This method requires careful attention to the problem of normalization so that neither nominal nor linear attributes are regularly given too much weight. The other two distance functions are the Interpolated Value Difference Metric (IVDM) and the Windowed Value Difference Metric (WVDM). In [4], Wilson and Martinez also proposed a generic version of the VDM distance function, called the discretized value difference metric (DVDM).

Proceedings of the Seventh ACIS International Conference on Software Engineering,
Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06)
0-7695-2611-X/06 $20.00 © 2006 IEEE

Table 1. Data set for text classification in UCI repository

Data Set	Size	FN	Features type			No. of Classes	No. of Misiing
			Continuous	Linear	Nominal		
Bridges	106	11	1	3	7	7	65
Flag	194	28	3	7	18	8	
Heart	270	13	5	2	6	2	
Hepatitis	155	19	6	0	13	2	167
Promoters	106	57	0	0	57	2	
Zoo	90	16	0	0	16	7	

Table 2. Generalization accuracy using one distance function with weights

Data Set	HEOM		HVDM		DVDM		IVDM	
	Aver.	Combining	Aver.	Combining	Aver.	Combining	Aver.	Combining
Bridges	62.21	66.31	63.48	63.22	62.67	65.83	64.18	**66.53**
Flag	52.64	53.45	56.90	57.88	56.67	**58.21**	55.16	55.41
Glass	74.07	77.05	73.05	75.08	65.03	66.51	80.98	**85.61**
Heart	77.28	77.86	78.07	79.23	80.50	**81.00**	77.74	79.02
Hepatiti	79.48	80.33	75.08	75.67	80.05	82.83	88.78	**89.00**
Promot	81.59	88.50	90.28	90.38	90.61	91.38	**91.68**	91.38
Zoo	97.29	**98.89**	97.57	**98.89**	95.75	96.11	96.09	96.11
Average	74.94	77.48	76.34	77.19	75.90	77.41	79.23	**80.44**

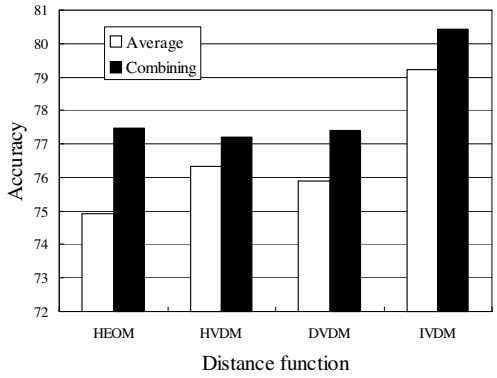

Figure 2. Generalization accuracy by one distance function

Proceedings of the Seventh ACIS International Conference on Software Engineering,
Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06)
0-7695-2611-X/06 $20.00 © 2006 IEEE

Table 3. Generalization accuracy by combining two distance functions

Data Set	HOHV	HODV	HOIV	HVDV	HVIV	DVIV
Bridges	64.75	63.92	**67.97**	64.44	65.44	65.42
Flag	57.86	60.57	58.23	**62.74**	59.82	58.33
Glass	77.48	75.30	84.95	75.22	**85.15**	81.48
Heart	79.51	82.86	81.24	80.93	**81.45**	81.43
Hepatit	77.67	82.67	83.33	78.17	81.83	**85.83**
Promote	92.38	**93.38**	94.38	90.38	92.38	91.38
Zoo	**98.89**	**98.89**	**98.89**	**98.89**	**98.89**	96.11
Average	78.36	79.65	**81.29**	78.68	80.71	80.00

2.2. Weighted Distance Functions for Optimization

The distance function in HEOM is defined in the previous section 2.1. To characterize the respective distance function for the training data, the weighted distance function is proposed in this paper as follows,

$$Weighted\ HEOM(x,y) \quad = \quad \sqrt{\sum_{i=1}^{n} \omega_i \times (x_i - y_i)^2}$$

where ω_i shows the weight of the i-th component of the data x_i and y_i. Here, the weights are normalized as follows,

$$\sum_{i=1}^{n} \omega_i = 1, \quad \omega_i \geq 0$$

The problem, here is how to derive the optimized weights $\{omega_i\}$. The optimized weights of the distance function, are computed by applying Genetic Algorithm(GA) to the training data.

2.3. The DkNN Algorithm (Combining Distance Functions)

Fig.1 shows the basic framework of the proposed new system. We call this system k-nearest neighbor classification by combining multiple different distance functions(DkNN). First, the DkNN inputs several distance functions. Then, it uses each distance function to generate k nearest samples in the training data. It combines the all k nearest samples and determines the class of unknown object based on the simple voting.

2.4. Combining Ensemble Computation

To improve the accuracy in the classification, the operation of classifiers as a decision committee, is needed. A committee as the final classifier, here, is composed of ensemble classifiers as committee members, each of which makes its own classifications that are combined to create a single classification result of the whole committee. The algorithm is described in [13].

3 Experimental Results for Combining Computations

For evaluating the classification generalization accuracy of our algorithm, the DkNN algorithm was implemented and tested on 7 benchmark dataset from the UCI Machine Learning Repository [8]. The basic characteristics of the data sets contain the size of training data and test data, the number of the features (FN) and feature type (Continuous Linear, Nominal), the number of class and the number of missing values (Missing) in Table 1. For the ensemble computations, we used the 10-fold cross validation[4].

Here, we computed function, HEOM, HVDM, DVDM and IVDM, respectively. Then, the generalization accuracy is shown in Table 2. The highest accuracy achieved for each dataset is shown in bold type. Table 2 is shown in Fig.2 by bar chart. The combining ensemble is superior than the averaging processing. The combining ensemble in the distance function, IVDM, shows high accuracy.

In Fig.3, the generalization accuracy by one distance function is shown in bar chart. The dotted bar shows the average value, while the black bar shows the combining ensemble in the respective distance function.

In Table 3, the generalization accuracy by combining two different distance functions, is shown, which include the combining ensemble processing. The highest accuracy achieved for combining two functions, is shown in bold type. These values in Table 3 show the improved accuracy than the one distance function in Table 2. Further, combining three and four distance functions, are computed. In Table 4, the generalization accuracy by combining three or four distance functions, is shown.

These values improve further the accuracy than the results in Table 3. The combining four different distance functions, show the value with the highest accuracy among all combining functions. Table 4 is shown in Fig.3 by bar chart.

Table 4. Generalization accuracy by combining three and four distance functions

Data Set	HOHVDV	HOHVIV	HODVIV	HVDVIV	HOHVDVIV
Bridges	66.97	67.97	**69.19**	65.56	68.08
Flag	60.31	61.36	61.06	60.48	**63.17**
Glass	77.77	83.33	82.59	82.56	**84.31**
Heart	81.94	81.88	83.60	**84.12**	**84.12**
Hepatit	79.67	83.33	84.67	**86.67**	84.00
Promote	92.38	92.38	92.38	92.38	**93.38**
Zoo	**98.89**	**98.89**	**98.89**	97.22	**98.89**
Average	79.70	81.31	81.77	81.28	**82.28**

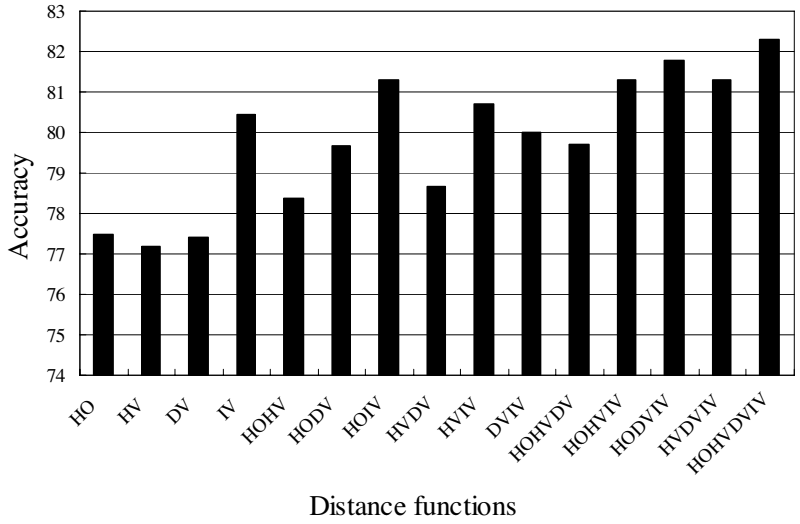

Figure 3. Generalization accuracy by combining distance functions

4 Comparison of Combining Functions with Other Learning Algorithms

To compare the proposed combining distance functions with other conventional learning algorithms, we show the computed results in Table 5. The bold type in Table 5, shows the value with almost the highest accuracy in the data. In the Table, the C4.5 system is a famous inductive decision tree algorithm. The accuracy by applying C4.5, is shown in [7](release 8). For C4.5, we give results of using tree, pruned tree and rules. IB1 and IB2 are instance-based algorithms. IB1 is a simple nearest neighbor classifier with k=1, while IB2 prunes the training set. The results are published as the classification accuracy by applying the instance-based algorithms, IB1 and IB2[5]. A naive Bayesian classifier(Bayes)[9, 3] and the backpropagation(BP) in neural network[4, 6], are also applied to these data. IDIBL[4] is an integrated decremental instance-based learning algorithm. It uses the Interpolated Value Difference Metric(IVDM) to have an appropriate distance measure between input vectors that can have both linear and nominal attributes. For IDIBL, we just use the results reported by Wilson[4]. CPP/p means the accuracy of using pruned center points and good border points to reduce the storage in our previous studies[12, 13]. Combin. shows the proposed combining method in this paper, which the consists of combining four functions of HOHVDVIV. The combining method shows almost the highest generalization accuracy superior to other conventional learning algorithms.

5 Conclusion

The proposed functions combining and ensemble combining method in this paper is a new algorithm, which improves the performance of the k-nearest neighbor. It is based on genetic algorithm and the k-nearest neighbor algo-

Table 5. Generalization accuracy of combining function and well-known algorithms

DataSet	C4.5			IB		Bayes	BP	IDIBL	CBP/p	Combin.
	Tree	P-Tree	Rule	IB1	IB2					
Bridges	**68.00**	65.30	59.50	53.80	45.60	66.10	67.60	63.20	63.92	**68.08**
Flag	59.20	61.30	60.70	**63.80**	59.80	52.50	58.20	57.70	60.00	**63.17**
Glass	68.30	68.80	68.60	70.00	66.80	71.80	68.70	70.60	83.22	**84.31**
Heart	73.30	72.10	80.00	76.20	68.90	75.60	82.60	83.30	**84.33**	**84.12**
Hepat	77.70	77.50	78.80	80.00	67.80	57.50	68.50	81.90	83.33	**84.00**
Promo	73.30	71.90	79.10	81.50	72.90	78.20	87.90	88.60	92.00	**93.38**
Zoo	91.00	91.00	91.40	96.40	97.50	97.80	95.60	92.20	94.86	**98.89**
Average	72.97	72.56	74.01	74.53	68.47	71.36	75.59	76.79	80.24	**82.28**

rithm. Combining multiple classifiers is an effective technique for improving accuracy. In this paper we present a new approach to combine multiple kNN classifiers based on different distance functions, which improve the performance of the k-nearest neighbor classifier. The proposed combining algorithm shows the higher generalization accuracy when compared to other conventional learning algorithms.

References

[1] S. Bay. Combining nearest neighbor classifiers through multiple feature subsets. *Intelligent Data Analysis*, 3:191–209, 1999.

[2] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[3] D. D.Michie and C.Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, Hertfordshire, England, 1994.

[4] D.R.Wilson and T. Martinez. An integrated instance-based learning algorithm. *Computational Intelligence, 16(1)*, pages 1–28, 2000.

[5] D. D.W.Aha and M.K.Albert. Instance-based learning algorithms. *Machine Learning, 6*, pages 37–66, 1991.

[6] J.L.McClelland and D.E.Rumelhart. *Explorations in Parallel Distributed Processing*. MIT Press, 1988.

[7] J.R.Quinlan. *C4.5:Programs for Machine Learning*. Morgan Kaufmann, 1993.

[8] C. Merz and P. M. Murphy. Uci repository of machine learning databases irvine. 1998. CA: University of California Irvine, Department of Information and Computer Science, Internet: http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[9] W. P.Langley and K.Thompson. An analysis of bayesian classifiers. *Proc. of 10th National Conf. on A.I.(AAAI-92), AAAI/MIT press*, pages 223–228, 1992.

[10] C. Stanfill and W. D. Toward memory-based reasoning. *Communications of the ACM*, 29:1213–1228, 1986.

[11] D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6(1):1–34, 1997.

[12] Y.Bao and N.Ishii. Combining multiple k-nearest neighbor classifiers for text classification by reducts. *Proc.5th International Conference on Discovery Science(LNAI2534,Springer-Verlag)*, pages 361–368, 2002.

[13] N. Y.Bao, E.Tsuchiya and X.Du. Classification by instance-based learning algorithm. *Proc. IDEAL 2005(LNCS3578, Springer-Verlag)*, pages 133–140, 2005.