Weighted *k*NNModel-based Data Reduction and Classification

Xuming Huang[1, 2], Gongde Guo[1, 2] and Daniel Neagu[2]

[1] Department of Computer Science, Fujian Normal University, Fuzhou 350007, China
[2] Department of Computing, University of Bradford, Bradford, BD7 1DP, UK

Abstract: A weighted *k*NNModel-based data reduction and classification algorithm, called *wk*NNModel, is proposed in this paper which aims to find some more meaningful representatives to replace the original dataset for further classification. Each representative is formed by an instance and its weighted neighborhood which satisfies a predefined threshold. Compared to *k*NNModel the proposed method as an alterative to other *k*NN algorithms, further alleviates the effect of abnormal data, i.e. noisy or boundary disturbance data, to data reduction and classification, thus contributing to the improvement of data reduction rate.

Key words：*k*NN；*k*NNModel；model construction；reduction；classification

1 Introduction

Classification is a method to establish categorization rules so as to classify new data on the basis that each class has already had been defined by a certain sample data. Among the well-known classification algorithms, *k*NN (the K-Nearest-Neighbors), which is simple but effective in many cases [1] [2], is widely studied and applied. The basic idea of *k*NN (the *k*-Nearest-Neighbors) is: among *k* data nearest to a new datum *t*, the class that most data belong to will decide the class of *t*. Therefore two major factors affecting the application of *k*NN include: (1) the selection of parameter *k*: the classification accuracy of *k*NN, to some extent, depends on the selection of the value *k*; (2) the classification efficiency: being a lazy learning algorithm without pre-modelling, *k*NN spends all computing time on new data classification.

When dealing with a large repository, *k*NN often results in pretty high computing cost, which has prohibited it from being applied to many areas. Many researchers are trying hard to alleviate this problem by data reduction [3]. The *k*NN algorithm considers that the class of a new instance to be classified is determined by the category of its highest weighted nearest neighbors, which also means instances from its nearest neighbors may belong to other classes. This paper proposed to discuss how to apply the weighted *k*NNModel to deal with datasets with large size by the proposed data reduction and classification method.

1.1 Related work

Two data reduction methods generally used in literature are: selection and replacing methods.

- Selection method: aims to reduce a dataset by selecting representatives from the training dataset.

Selection-based data reduction methods include: CNN (Condensed Nearest Neighbor Rule) proposed by Hart [4] at the end of the 1960's, RNN (Reduced Nearest Neighbor Rule) proposed by Gate [5] in the 1970's, SNN (Selective Nearest Neighbor Rule) proposed by Ritter [6], a series of sample-based learning methods (IB1~IB5) proposed by Aha [7, 8] at the early stage of the 1990's and the most recent (DROP1~DROP5, DEL) proposed by Wilson [3].

Their major shortcoming lies in the fact that there is no inductive learning in the selection process, that is, there is no additional information useful for classification. Therefore, although it can effectively reduce the number of training datasets, it can not "improve" but only "preserve" its classification accuracy.

- Replacing method [10, 11, 12]: replaces the original training dataset with a new training dataset, and the data from the new training dataset may differ from any data in the original training dataset.

Two commonly used methods are Hierarchical Clustering method by Chang and Bezdek [10] and the further improvements put forward by Mollineda [11, 12].

The major shortcoming of replacing method lies in its computing complexity. Plenty of computing time is exhausted on endless amalgamation and post-amalgamation check over the consistency with original data [3].

*k*NNModel proposed by Guo et al [9] combines the strength of both selection and replacing methods. By using kNNModel method, a representative datum chosen from training dataset (selection method), plus its own category and neighborhood information, forms a representative tuple (or global neighbourhood). It is a condensed result of a certain local region in its original data space. All data from this local region share the same class with the representative data. A new training dataset (model)

formed by these representative tuples will replace the original training datasets (replacing method) for further classification.

In order to improve the classification accuracy of *k*NNModel algorithm, it is advised by Guo et al [9] that a proper error tolerant degree should be set. During the modelling process, it is allowed that each local neighbourhood covers ε data different from the category of this neighbourhood. Here ε is the so-called error tolerant degree. In each modelling process, the value of ε is fixed. The purpose of setting the error tolerant degree ε is to effectively classify the datasets containing noisy data to some extent. And its effectiveness has been testified by the experiment results [9].

1.2 Contributions of this paper

Compared to *k*NNModel in terms of representative selection, the proposed *wk*NNModel (Weighted *k*-Nearest Neighbors Model) data deduction and classification method takes into account the similarity-based weighted cohesion of the data from local neighbourhood to the central point. The setting of threshold value alleviates noisy data's effect over the area covered by each representative tuple, that is, to make the area covered by a representative tuple and its cohesion to the data from the same class as large as possible, while the effect of noisy data from different category becomes as little as possible.

Compared with representative tuple in [9], the class-representative generated from the model construction process enlarges the number of same-class data covered by a neighbourhood. Due to the use of weighted method, the number of different–class data contained in each class representative is decided by weight values. Different class representatives may include different number of data belonging to other classes. However, the error tolerant degree ε in [9] limits the number of data from different classes in each representative tuple. Therefore, *wk*NNModel aims at improving both data reduction rate and classification accuracy.

The rest of this paper is organized as follows: Section 2 gives a brief introduction about the basic idea of *wk*NNModel and its algorithm; Section 3 describes and evaluates experimental environment and results. This paper ends with some conclusions and further research directions.

2 The proposed wkNNModel method

2.1 Basic idea

"Birds of a feather flock together" is a basic principle of data classification. Data from the same class is characterized with strong cohesion among them. However, applications indicate that it doesn't necessarily mean there is no data from other class existing in the local space occupied by a certain class. There may exist some noisy and boundary disturbance data generated in the process of using different similarity measures or measuring errors generated during data collecting process. The proposed weighted method is to deal with such data cohesive to each other in a local space whilst allowing few data belonging to other classes. These data and their neighborhoods will serve as class representatives (called representative tuples in [9]). It aims at reducing the effect of data from other classes in the process of classification, and to form a classification model (the reduced dataset of the original training dataset). On this basis, a new instance is still classified in the spirit of *k*NN. This is the basic idea of the proposed *wk*NNModel method.

When Euclidean distance is used as similarity measure, data of a class often cluster in a neighbourhood whose corresponding central data have rather strong cohesion to data of the same class, while data from different classes have weaker weighted influence to the central part. The basic idea of data reduction is to find those representative data d and its neighbourhood $\overline{O}(d,r)$ which covers a number of nearest neighbours through defining the influence between any two data. In $\overline{O}(d,r)$, d is the generalised center, r is the radius of the neighbourhood and c is the effect among the nearest neighbours whose class is the same as that of representative data d and should be far stronger than that of the data with different class. The effect from different-class data is set within a scope controlled by threshold value.

Therefore, for the purpose of data reduction, it is expected that $\overline{O}(d,r)$ contains as much data from the same class with d as possible. So d becomes the representative data of the class covered by $\overline{O}(d,r)$. The meaning of other symbols used in this paper are: $Num(d)$ — the number of data with the same class as d covered by $\overline{O}(d,r)$; $Sim(d)$ or r — the distance between the most marginal data and its central point d within $\overline{O}(d,r)$. As a model, the group formed by those representatives will replace the original training dataset, thus complete the task of reducing original training dataset. For a new

data *t* to be classified, if it is only covered by $\bar{O}(d,r)$ of one representative, it will be determined to share the class with *d;* if it is covered by several representatives, the number of those representatives will be counted according to the number of their categories. The class with the most members will serve as the category of *t*. If *t* is not covered by $\bar{O}(d,r)$ of any representative, its category will be determined by $\bar{O}(d,r)$ whose border is the nearest to it.

The following graphs briefly illustrate the difference of class representative selection between *k*NNModel and *wk*NNModel.

Given a training dataset with two classes denoted as squares and triangles in Figure 1, its distribution in the two-dimensional space is presented as follows:

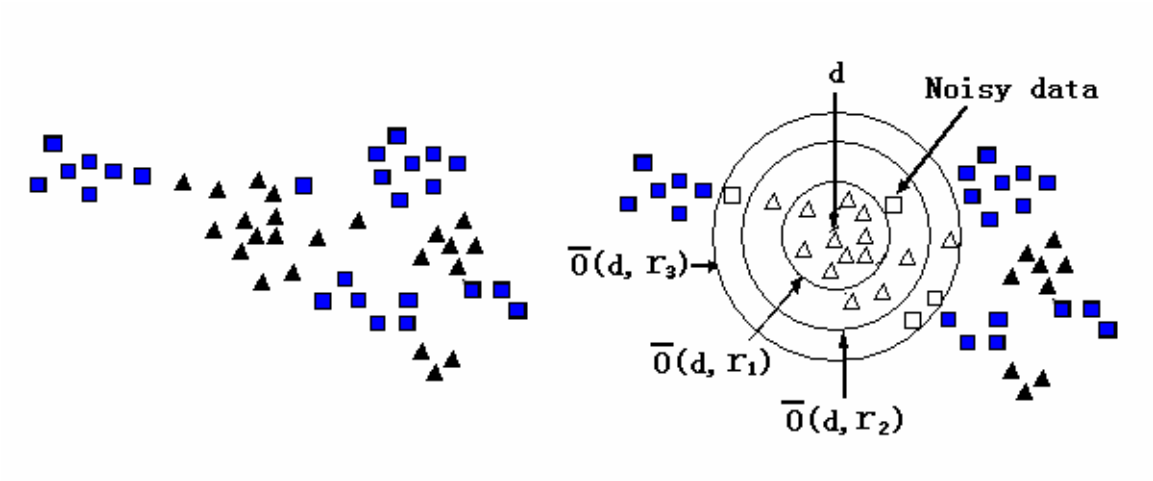

Fig 1. Data distribution in two-dmensional space

Fig 2. The difference between $\bar{O}(d,r_2)$ of *wk*NNModel and $\bar{O}(d,r_1)$ of *k*NNModel

In Figure 2, $\bar{O}(d,r_1)$ represents a neighborhood covered by the first representative tuple d that is found by *k*NNModel (ε=0). $\bar{O}(d,r_2)$ represents a neighborhood covered by the first class representative found by *wk*NNModel. It is clear that there is a noise because of an instance from other class in $\bar{O}(d,r_2)$ within the scope controlled by weighted threshold value; but d and its neighborhood $\bar{O}(d,r_3)$ cannot serve as a class representative, because the cohesion (weighted value) of the four instances from other classes has exceeded the scope allowed by the threshold value (maximum 3 for example).

2.2 Description of algorithm

To maintain the generality, the "data" mentioned here is not particularly referring to the data in a certain application case. Similarly, the so-called "similarity" can be any one of commonly used similarity measures if it is suitable for a given application. Taking Euclidean distance as an example, the distance between data d_1, d_2 is presented as $\rho(d_1, d_2)$.

Given a dataset D with n classes denoted as $c_1, c_2, ..., c_n$, where C_i is a collection of data from c_i class, then *D* can be represented as

$$\boldsymbol{D}=\bigcup_{i=1}^{n} C_i \tag{1}$$

For $\forall d \in \boldsymbol{D}$, $\bar{O}(d,r)$ represents a data group from D, whose distance to *d* is less than or equal to *r*. If $d \in C_i$ the proposed *wk*NNModel algorithm finds the $\bar{O}(d,r)$ which can serve as a class representative neighborhood. When *r* extends outwards, the cohesion of data from $\bar{O}(d,r)$ to the data of the same class should be far stronger than the cohesion of data from other classes (which is controlled by threshold value.). For each C_i such class representative is probably more than one.

The cohesion between every two data and the distance between them is inversely proportional, that is, the closer they are to each other, the stronger the cohesion between them, and vice versus. For any instance d_j from $\bar{O}(d,r)$, its effect to d, is defined as follows:

$$\text{effect} = 1 - \frac{\rho(d,d_j)}{r} \text{ where, } 0 \le 1 - \frac{\rho(d,d_j)}{r} \le 1 \tag{2}$$

The total effect from all instances in $\bar{O}(d,r)$ to the center d is defined as $w(d,r)$:

$$w(d,r) = \sum_{d_j \in \bar{O}(d,r) \wedge d_j \in D} (1 - \frac{\rho(d,d_j)}{r}) \tag{3}$$

Here, $w_C(d,r)$, the cohesion weight of the data from $\bar{O}(d,r)$ sharing a category C_i with d, is defined as:

$$w_C(d,r) = \frac{\sum_{d_j \in C_i \wedge d_j \in \bar{O}(d,r)} (1 - \frac{\rho(d,d_j)}{r})}{w(d,r)} \tag{4}$$

$w_F(d,r)$, the cohesion weight of data different from the class of d is

$$w_F(d,r) = \frac{\sum_{d_j \notin C_i \wedge d_j \in \bar{O}(d,r) \wedge d_j \in D} (1 - \frac{\rho(d,d_j)}{r})}{w(d,r)} \tag{5}$$

Obviously, the sum of $w_C(d,r)$ and $w_F(d,r)$ satisfies the following condition:

$$w_C(d,r) + w_F(d,r) = 1 \tag{6}$$

As a class representative, it is expected that r and $w_C(d,r)$ is as big as possible, while $w_F(d,r)$ is as small as possible.

An algorithm based on aforementioned idea and the definition of weight, called *wk*NNModel_o, represents the original algorithm of *wk*NNModel. Unfortunately, *wk*NNModel_o doesn't reach anticipated accuracy and reduction rate in the experiment. Through carefully analysing each representative generated by *wk*NNModel_o, we found that some representatives have pretty low density. A typical example of such representatives is shown in Fig. 3. A candidate of class representative is expanded outwards along the radius r1, r2, r3, until R. Because d1, d2 are far away from *d*, their effects to *d* are quite small. Adding them into $\bar{O}(d,r_4)$ may not exceed predefined threshold of $\bar{O}(d,r_4)$, thus $\bar{O}(d,r_4)$ could be chosen as a representative. However, $\bar{O}(d,r_4)$ covers a too large size of sparse space between $\bar{O}(d,r_1)$ and $\bar{O}(d,r_4)$, resulting in relatively lower density of this representative, thus affecting its representativity to the local area covered by it. In order to solve this problem, a new parameter *p* is introduced to control density of each representative in the final algorithm, called *wk*NNModel_d. The density control value is used to avoid choosing a candidate as a class representative which covers a too large size of sparse space, in order to improve the quality of each class representative. The experimental results indicate that the density control value indeed very much influence the reduction rate and classification accuracy of *wk*NNModel.

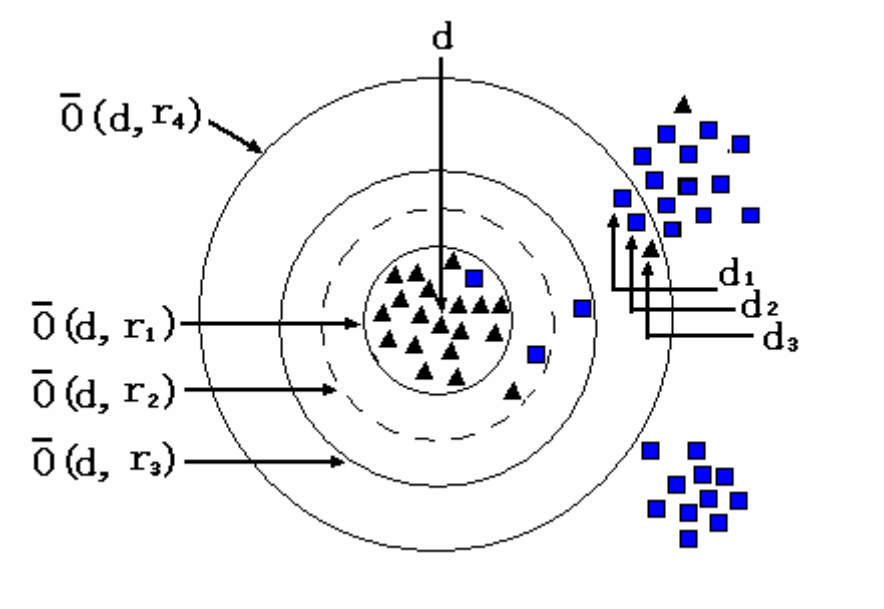

Fig 3. $\bar{O}(d,r_4)$: An example of a representative without counting density

The detailed model construction algorithm of *wk*NNModel_d, is described as follows:

(1) Input a threshold α , a density control value *p* and a dataset *D* with *n* classes and *q* features.

(2) Create a normalized version of *D*: D_N ={ $d_1,d_2,...d_{|D|}$ }, where $d_i \in D_N$ denoted as $(d_{i1},d_{i2},\ldots,d_{iq})$, Obviously, for Euclidean distance, $D_N \subseteq \bar{O}(o,2\sqrt{q})$. o is the centre of area covered by $\bar{O}(o,2\sqrt{q})$; Select a similarity measure and create a similarity matrix from D_N; Calculate $p_i = \frac{m_i}{2\sqrt{q}}$ - the density of class c_i on the radius of $\bar{O}(o,2\sqrt{q})$, where m_i is the number of instances with class c_i in D_N; Finally, set to '0' the tag of all data in D_N.

(3) For each instance *d* with tag *'0'* (assume that $d \in C_i$) , the algorithm starts from the nearest instance with tag "0" or "1" and expands it outwards along the radius until it does not satisfy the condition of $\frac{Num(d)}{r} \geq \frac{p_i}{p}$, the latest $\bar{O}(d,r)$ satisfies the condition of $w_F(d,r) \leq \alpha$ is chosen as a candidate of class representative. If it doesn't satisfy the condition of $\frac{Num(d)}{r} < \frac{p_i}{p}$ once the nearest data is chosen, the tag of *d* is set to "-1" .

(4) Among the candidates generated from step 3, we choose one with largest *Num(d)* as a class representative and add it into M_α . Meanwhile, we set "1" to the tag of all the data covered by it.

(5) Repeat step 4 to 5 until all the data's tag in D_N is set to "1" or "-1".

(6) Output the created model M_α

In the above algorithm, a data with tag "0' means it has not been treated; a data with tag "-1" means it is not qualified as a candidate of class representative; a data with tag "1" means the data has already been covered by one of generated representatives.

In Step 4, if there is more than one candidate having the same largest value of *Num(d)*, the one with smallest value of *r* is chosen as a class representative. Choosing a candidate with smallest radius but covering largest number of data as a class representative is to improve reduction rate whilst keeping higher density of a class representative as possible.

For model M_α , a class representative $\bar{O}(d,r)$ is denoted as the form of $<Cls(d)$, $Sim(d)$, $Num(d)$, $\mathrm{Rep}(d)>$, where $Cls(d)$ represents the class label of *d*, $Sim(d)$ represents the radius r of $\bar{O}(d,r)$, $Num(d)$ represents the number of data in $\bar{O}(d,r)$ with category *d*, and $\mathrm{Rep}(d)$ represents a representation of *d* itself.

The classification process of *wk*NNModel_d is described as follows:

(1) For a new data t to be classified, calculate its similarity to all representatives in the model $M\alpha$.

(2) If t is covered only by one representative $\bar{O}(d,r)$, t is classified as the category of d.

(3) If t is covered by at least two representatives with different categories, count the number of representatives in each class, classify t as the category with the largest number.

(4) If no representative in the model $M\alpha$ covers t, find a $\bar{O}(d,r)$ with minimal $\rho(d,t)-r$ and classify t as the category of d.

3 Experiment and Evaluation

3.1 Datasets

Experiments using the 10-fold cross validation method have been carried out to evaluate the prediction accuracy of w*k*NNModel_d, and to compare the experimental results with *wk*NNModel_o, *k*NN, *k*NNModel and other data reduction algorithms in literature.

Fifteen public datasets were chosen from the UCI machine learning repository. Some information about these datasets is listed in Table 1.

Table 1 General information about the chosen datasets

Dataset	NA	NN	NO	NB	NE	CD
Aust	14	4	6	4	690	383:307
Colic	22	16	7	0	368	232:136
Diabetes	8	0	8	0	768	500: 268
Glass	9	0	9	0	214	70:76:17:13:9:29
HCleveland	13	3	7	3	303	164:139
Heart	13	3	7	3	270	150:120
Hepatitis	19	7	1	12	155	32:123
Ionosphere	34	0	34	0	351	126:225
Iris	4	0	4	0	150	50:50:50
LiverBupa	6	0	6	0	345	145:200
Sonar	60	0	60	0	208	97:111
Vehicle	18	0	16	0	846	212:217:218:199
Vote	16	0	0	16	232	124:108
Wine	13	0	13	0	178	59:71:48
Zoo	16	16	0	0	90	37:18:3:12:4:7:9

In Table 1, the meaning of the title in each column is follows: NA-Number of attributes, NN-Number of Nominal attributes, NO-Number of Ordinal attributes, NB-Number of Binary attributes, NE-Number of Examples, CD-Class Distribution.

As we use Euclidean distance as a similarity measure in the experiment for *k*NN, *k*NNModel and *wk*NNModel_o and *wk*NNModel_d, fifteen datasets were pre-processed, i.e. normalized, before conducting the classification.

3.2 Evaluation

In the experiment, the parameters setting for each algorithm is described as follows:

- For *wk*NNModel_o (*wk*NNM_O in the Table 2 and 3 for short), the threshold α is set from 0.01 to 0.1 with step 0.01. the α value with best classificaiton accuracy is chosen and reported in Table 2 and 3.
- For *wk*NNModel_d (*wk*NNM_D for short), the threshold α is set from 0.01 to 0.1 with step 0.01, and p is set from 1 to 20 with step 1 for each α, we choose the α and p values with the best classification accuracy.
- For *k*NNModel (*k*NNM for short), the error tolerant degree r is set to 0.

- For *k*NN, the value of *k* is set to 5. In our experiment, the average classificaiton accuracy of 5-NN over 15 datasets is better than 1-NN and 3-NN.

The classification accuracies of *k*NNM_O, *wk*NNM_D，*k*NN, *k*NNM and other data reduction algorithms [3] using 10-fold cross validation method are presented in Table 2. The corresponding reduction rates are presented in Table 3.

Table 2 A comparison of data reduction methods by classification accuracy

Dataset	α	*wk*NNM_o	α	*p*	*wk*NNM_d	*k*NN	*k*NNM	*C*NN	SNN	IB3	DROP3	DEL
Aust	0.05	71.88	0.09	3	83.47	**85.22**	84.64	77.68	81.31	**85.22**	83.91	84.78
Colic	0.05	78.28	0.04	13	82.08	**83.06**	82.50	59.90	64.47	66.75	70.13	67.73
Diabetes	0.01	65.37	0.01	19	67.84	74.21	74.08	65.76	67.97	69.78	**75.01**	71.61
Glass	0.03	57.05	0.03	17	61.68	67.62	65.24	68.14	64.39	62.14	65.02	**69.60**
HCleveland	0.10	73.53	0.07	17	78.19	81.00	80.33	73.95	76.25	**81.16**	80.84	79.49
Heart	0.05	73.70	0.09	5	80.37	80.37	80.74	70.00	77.04	80.00	**83.33**	78.89
Hepatitis	0.08	81.25	0.03	2	84.58	83.33	**85.33**	75.50	81.92	73.08	81.87	80.00
Ionosphere	0.06	80.65	0.07	5	87.17	84.00	**93.71**	82.93	81.74	85.75	87.75	86.32
Iris	0.02	94.66	0.08	2	**96.67**	**96.67**	96.00	90.00	83.34	94.67	95.33	93.33
LiverBupa	0.03	64.87	0.04	13	65.52	66.47	64.41	56.80	57.70	58.24	**78.00**	61.38
Sonar	0.04	75.50	0.01	10	81.33	**85.00**	82.50	74.12	79.81	69.38	78.00	83.59
Vehicle	0.01	59.42	0.01	15	62.87	**69.29**	65.36	67.50	67.27	67.62	65.85	68.10
Vote	0.09	88.76	0.08	8	92.21	92.17	88.70	93.59	95.40	95.64	**95.87**	94.27
Wine	0.07	89.80	0.04	7	93.82	94.71	94.71	92.65	**96.05**	91.50	94.93	94.38
Zoo	0.07	96.66	0.09	13	**98.89**	**95.56**	92.22	91.11	76.67	92.22	90.00	90.00
Average	/	76.76	/	/	81.11	**82.58**	82.03	75.98	76.76	78.21	81.72	80.23

Table 3 A comparison of data reduction methods by reduction rate

Dataset	α	wkNNM_o	α	*p*	*wk*NNM_d	*k*NN	*k*NNM	*C*NN	SNN	IB3	DROP3	DEL
Aust	0.05	95.65	0.09	3	**99.13**	0	90.43	75.78	75.85	95.22	94.04	**97.40**
Colic	0.05	**92.93**	0.04	13	92.39	0	84.24	64.34	51.35	**91.51**	89.70	78.18
Diabetes	0.01	83.98	0.01	19	83.59	0	86.98	63.11	57.05	**89.03**	83.10	87.36
Glass	0.03	76.17	0.03	17	76.17	0	**88.32**	61.47	57.37	66.20	76.12	61.58
HCleveland	0.10	**98.68**	0.07	17	98.02	0	87.79	69.16	66.12	**88.89**	87.24	86.36
Heart	0.05	95.56	0.09	5	**97.41**	0	88.52	73.83	66.22	86.42	86.38	**95.30**
Hepatitis	0.08	98.06	0.03	2	**99.35**	0	88.39	74.70	69.04	**94.91**	92.20	92.41
Ionosphere	0.06	80.65	0.07	5	**98.86**	0	85.15	78.38	80.79	85.41	**92.94**	87.12
Iris	0.02	97.33	0.08	2	**98.00**	0	**96.00**	87.26	85.93	80.22	85.19	90.44
LiverBupa	0.03	79.71	0.04	13	83.77	0	83.48	59.13	47.41	**89.34**	75.01	87.36
Sonar	0.04	**94.23**	0.01	10	91.83	0	86.06	67.15	71.74	**87.98**	73.13	70.14
Vehicle	0.01	78.72	0.01	15	79.43	0	**87.83**	62.96	56.79	71.64	77.00	67.49
Vote	0.09	**99.14**	0.08	8	**99.14**	0	93.53	90.88	89.79	94.56	94.89	**98.00**
Wine	0.07	97.75	0.04	7	**98.31**	0	90.45	85.70	85.77	83.40	83.89	**91.00**
Zoo	0.07	91.11	0.09	13	91.11	0	**92.22**	87.53	89.38	70.62	80.00	81.73
Average	/	90.65	/	/	**92.43**	0	**88.63**	73.43	70.04	85.02	84.72	84.79

In Table 2 and 3, the number with bold-font in each row represents the best classification accuracy or the highest reduction rate over the leftmost dataset. Some selection technique-based data reduction methods are chosen for comparison in the experiment. More information about data reduction methods can be found in literature [3].

*wk*NNM_d mainly relies on selection technique. So in the experiment, we compared *wk*NNM_d with few well-known selection technique-based data reduction methods, i.e., CNN, SNN, IB3, Drop3 and Del. From the experimental results, it is clear that wkNNM_o obtained the highest average reduction rate among 9 algorithms for comparison without decreasing classification accuracy. *wk*NNM_d also obtains the highest reduction rate over 7 out of 15 individual datasets.

From the experimental results, it is also clear that the problems of wkNNM_o appeared in Figure 3 have been solved by introducing the density control value in wkNNM_d. This justifies our hypothesis. Moreover, a study of the influence of density control value *p* to classification accuracy and reduction rate of *wk*NNM_d has been conducted in the experiment. The experimental results are shown in Figure 4 and Figure 5.

In the experiment, the relationship between classification accuracy and density control value, reduction rate and density control value are similar. They have similar movement over fifteen datasets. Figure 4 demonstrates the relationship between classification accuracy and density control value p over five datasets. Figure 5 demonstrates the relationship between reduction rate and density control value p over other five datasets.

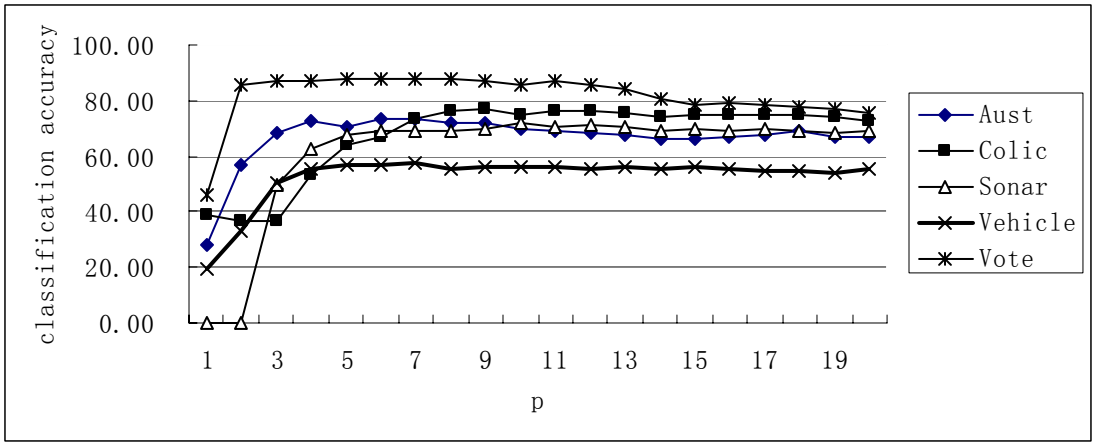

Fig 4. Relationship between classification accuracy and density control value

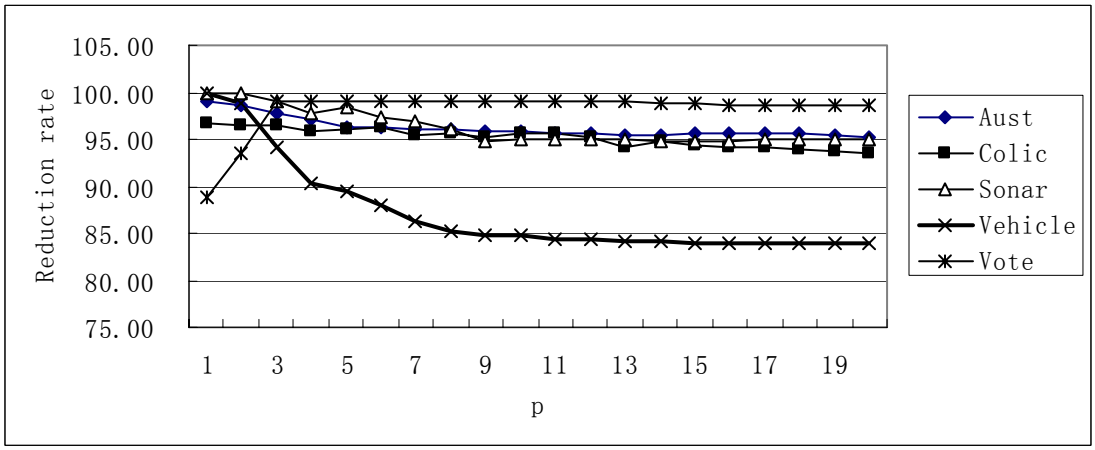

Fig 5. Relationship between reduction rate and density control value

In Figure 4, *p* is varied from 1 to 20 with step 1. The graph demonstrates similar movement over 5 different datasets – when *p*=1 (least control), the classification accuracy is pretty low. With the increase of p, the classification accuracy increases quickly at the first few steps. But after reaching a certain p value, the classification accuracy moves steady within a small range. However, with the increase of p, the reduction rate decreases slightly. So, the value of p can be tuned within a small range.

4 Conclusions

In this paper, we propose a similarity-based weighting computing technique aimed at overcoming the influence of different noisy data to the representative tuple. The introduction of a density control value in *wk*NNM_d significantly improves the quality of chosen class representatives thus improving the classification accuracy of *wk*NNM_o as well. Moreover, the proposed *wk*NNM_d significantly improves the reduction rate, thus improving the efficiency of classification of *wk*NNM_d. The experimental results on 15 public datasets show that *wk*NNM has obtained the highest average reduction rate among nine data reduction algorithms compared whilst preserving the classification accuracy.

Acknowledgment

This work was partly supported by the EPSRC project PYTHIA:GR/T02508/01 and Fujian Normal University

References:

1 D. Hand, H. Mannila, P. Smyth. Principles of Data Mining. The MIT Press, 2001.

2 F. Sebastiani. Machine Learning in Automated Text Categorization. In ACM Computing Surveys, Vol. 34, No. 1, pp. 1-47, March, 2002.

3 D.R. Wilson, T. R. Martinez. Reduction Techniques for Instance-Based Learning Algorithms. Machine learning, 38-3, pp. 257-286, 2000.

4 P. Hart. The Condensed Nearest Neighbour Rule, IEEE Transactions on Information Theory, 14, pp. 515-516, 1968.

5 G. Gates. The Reduced Nearest Neighbour Rule. IEEE Transactions on Information Theory, 18, pp. 431-433, 1972.

6 G.L. Ritter, H. B. Woodruff et al., An Algorithm for a Selective Nearest Neighbor Decision Rule. IEEE Transactions on Information Theory, 21-6, November 1975, pp.665-669, 1975.

7 D.W. Aha. Tolerating Noisy, Irrelevant and Novel Attributes in Instance-based Learning Algorithms. International Journal of Man-Machine Studies, 36, pp. 267-287. 1992.

8 D. W. Aha et al., Instance-Based Learning Algorithms. Machine Learning, 6, pp. 37-66， 1991.

9 G. Guo, H.Wang, D. Bell, Y.Bi, K.Greer: KNN Model-Based Approach in Classification. in Proc.of CoopIS/DOA/ODBASE 2003,pp. 986-996,2003

10 C. Chang, Finding Prototypes for Nearest Neighbor Classifiers, IEEE Trans. Comput. 23 (11), pp. 1179-1184, 1974.

11 R.A. Mollineda, F.J. Ferri, E.Visal. A Cluster-Based Merging Strategy for Nearest Prototype Classifiers, proceedings of the 15th ICPR. Vol. 2, Barcelona, Spain, pp.759-762, 2000.

12 R.A. Mollineda, F.J. Ferri, E.Visal. An Efficient Prototype Merging Strategy for the Condensed 1-NN Rule Through Class-Conditional Hierarchical Clustering, Pattern Recognition 35 (2002) 2771-2782, Published by Elsevier Science Ltd, 2002

13 C. Stanfill and D.Waltz., Toward Memory-Based Reasoning Communications of ACM, 29, pp. 1213-1228, 1986.