Information Sciences 179 (2009) 809–819

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Data gravitation based classification

Lizhi Peng [a], Bo Yang [a], Yuehui Chen [a,*], Ajith Abraham [b]

[a] School of Information Science and Engineering, University of Jinan, 106 Jiwei Road, 250022 Jinan, PR China
[b] Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology Trondheim, Norway

ARTICLE INFO

Article history:
Received 9 June 2005
Received in revised form 13 September 2008
Accepted 3 November 2008

Keywords:
Data gravitation
Machine learning
Classification
Feature selection

ABSTRACT

Data gravitation based classification (DGC) is a novel data classification technique based on the concept of data gravitation. The basic principle of DGC algorithm is to classify data samples by comparing the data gravitation between the different data classes. In the DGC model, a kind of "force" called data gravitation between two data samples is computed. Data from the same class are combined as a result of gravitation. On the other hand, data gravitation between different data classes can be compared. A larger gravitation from a class means the data sample belongs to a particular class. One outstanding advantage of the DGC, in comparison with other classification algorithms is its simple classification principle with high performance. This makes the DGC algorithm much easier to be implemented. Feature selection plays an important role in classification problems and a novel feature selection algorithm is investigated based on the idea of DGC and weighted features. The proposed method is validated by using 12 well-known classification data sets from UCI machine learning repository. Experimental results illustrate that the proposed method is very efficient for data classification and feature selection.

1. Introduction

Classification is an important problem for machine learning and data mining researchers. The basic idea of a classification algorithm is to construct a classifier according to a given training set. Once the classifier is constructed, it can predict the class value(s) of unknown test data sample(s). For the training data set, in most classification problems, the class value of each sample should be known and hence most classification problems belong to the category of supervised machine learning. Many techniques have been applied for classification, including decision trees [20,7], neural network (NN) [13], support vector machine (SVM) [2,24], fuzzy inference systems [27], rough set [17] and so on. Among these techniques, decision tree is simple and easy to be comprehended. Neural networks have proven to be an efficient approach for many classification tasks, however, its training efficiency is usually a problem. SVM is a relatively new machine learning method based on the statistical learning theory and structural risk minimization (SRM) principle. SVM is gaining popularity due to many attractive features, and promising empirical performance. SVM is based on the hypothesis that the training samples obey a certain distribution, which restricts its application scope. Rough set [17] theory has also been applied to classification in recent years especially for feature selection [10] or as a hybridization tool with other classification methods [14,15,19,9].

Shi et al. [22] presented a novel data preprocessing technique called *shrinking*. Shrinking technique optimizes the inner structure of data inspired by the Newton's universal law of gravitation. During the preprocessing stage, a shrinking-based approach for multi-dimensional data analysis is used. The main idea of this approach is that data points move along the direction of the density gradient simulating the Newton's universal law of gravitation, leading to clusters, which are condensed and widely-separated. Using the shrinking approach, authors [21] also proposed a dimension deduction approach

* Corresponding author. Tel./fax: +86 531 82767581.
E-mail addresses: plz@ujn.edu.cn (L. Peng), yangbo@ujn.edu.cn (B. Yang), yhchen@ujn.edu.cn (Y. Chen), ajith.abraham@ieee.org (A. Abraham).

0020-0255/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.11.007

for multi-dimensional data analysis. Indulska and Orlowska [12] proposed a spatial clustering algorithm called GRAVIclust. The proposed algorithm uses a heuristic to pick the initial cluster centers and utilizes center of cluster gravity calculations in order to arrive at the optimal clustering solution. Although both of them are more focused on clustering problems, both methods had been inspired by the concepts of physical gravitation. Webster et al. [25] applied the natural principles of gravitation and developed two local search optimization algorithms called GLSA1 and GLSA2.

Data gravitation based classification method is a novel classification method developed by simulating the gravitation and the law of gravitation in the physical world [18,26]. The main ideas of the DGC method are (1) there exists a kind of "force" named data gravitation between any two data samples; (2) computational data gravitation obey the law of gravitation in the physical world; (3) the class value of a test sample is determined by comparing the gravitation values of different data classes. This paper also propose a feature selection algorithm for DGC by weighted features. Weights are used to describe the importance of features in the DGC model, and the weights are optimized by a random iterative algorithm named tentative random feature selection (TRFS). These additional strategies make the DGC approach more attractive and effective. The proposed method is validated using 12 well-known classification data sets. Experimental results illustrate that the proposed method is efficient as a data mining tool for classification problems.

This paper is organized as follows: Section 2 introduces the theory of data gravitation, including concepts, lemmas and the law of data gravitation. The principle of classification and other important principles of DGC are introduced in Section 3. In Section 4, a novel feature selection algorithm for DGC is proposed. Experiment results and analysis are reported in Section 5. Finally, we provide some conclusions in Section 6.

2. Theory of data gravitation

2.1. Gravitation and the universal law of gravitation

As we know, there exists a kind of force between any two objects in the universe and this force is called gravitation in Physics. Gravitation obey the universal law of gravitation. In 1687, Newton published an important paper in which he illustrated the universal law of gravitation for the first time. The law indicates that the strength of gravitation between two objects is in direct ratio to the product of the masses of the two objects, but in inverse ratio to the square of distance between them. The law can be described as follows:

$$F = G\frac{m_1 m_2}{r^2}. \tag{1}$$

Since force has direction, the precise description of the law takes the following vector form:

$$\mathbf{F} = G\frac{m_1 m_2 \mathbf{r}}{|\mathbf{r}|^3}, \tag{2}$$

where F is the gravitation between two objects; G the constant of universal gravitation; m_1 the mass of object 1; m_2 the mass of object 2; r the distance between the two objects; $\mathbf{F}$ the vector form of F and $\mathbf{r}$ is the vector form of r.

2.2. Data gravitation

Many classification and clustering methods are proposed on the basis of the similarity between data samples. The similarity of two single data samples is only affected by the Euclidian distance between them. When we study the relationship of a single data sample and a group of data samples with certain properties, e.g. a data cluster or a data class, two primary factors are often accounted, one is the distance, the other is the number of data samples in the group and its density. The shorter the distance is, the more the single data sample is similar to the group; the more number of samples the group contains, also the more the single data sample is similar to the group. This relationship is illustrated in Fig. 1. By drawing an analogy with gravitation in physics, we introduce the concept of gravitation and the Law of Gravitation in physics for a data classification problem. We extract two essential features of data similarity: distance and data "mass". The similarity between data is treated as a new concept called data gravitation.

Definition 1 (*Data particle*). Data particle is a kind of data unit that also has "data mass". Data particle is made up of a group of data elements in data space. These data elements have a certain relationship. Usually this relationship refers to the geometrical neighborhood of these data elements. That is to say the distances between any two data elements in a data particle must be shorter than a predefined value. Data particle has two basic properties: data mass and data centroid.

Definition 2 (*Data mass*). The mass of a data particle is the number of data elements in the data particle.

Definition 3 (*Data centroid*). Suppose $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m (\mathbf{x}_i = \langle x_{i1}, x_{i2}, \ldots, x_{in}\rangle, i = 1, 2, \ldots, m)$ are a group of data elements in a n-dimensional data space S, P is a data article built up by $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$. Therefore, the data centroid of P, $\mathbf{x}_0 = \langle x_{01}, x_{02}, \ldots, x_{0n}\rangle$ is the geometrical center of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ described as follows:

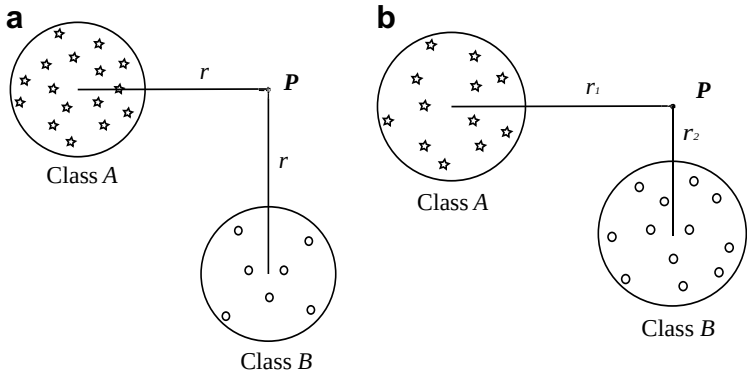

Fig. 1. The relationship of distance, data mass and data similarity. In (a) P belongs to class A because A contains more samples than B. In (b) P belongs to class B because the distance between P and B is shorter than that of A.

$$x_{0j} = \frac{\sum_{i=1}^{m} x_{ij}}{m}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n. \tag{3}$$

Since data particle has data mass and data centroid, data particle is described by a pair expression $\langle m, \mathbf{x} \rangle$. Where m is the data mass of the data particle and $\mathbf{x}$ is the data centroid. After the class information (feature y) is taken into count, data particle is described as a triple expression $\langle m, \mathbf{x}, y \rangle$.

Definition 4 (*Atomic data particle*). An atomic data particle is a data particle containing only one data element. The data mass of an atomic data particle is 1.

Definition 5 (*Data gravitation*). Data gravitation is defined as the similarity between data particles and is a kind of scalar. This is an important factor different from the physical force. For the same data particle, gravitation from different data classes can be compared.

On the other hand, data gravitation from the same class obey the superposition principle.

Lemma 1 (Superposition principle). *Suppose $p_1, p_2, \ldots, p_m$ are m data particles in a data space, and they belong to the same data class. The gravitation they act on another data particle are $F_1, F_2, \ldots, F_m$, and then the composition of gravitation is given by:*

$$F = \sum_{i=1}^{m} F_i. \tag{4}$$

Definition 6 (*Data gravitation field*). Data particles act on each other by data gravitation, and form a field that congests the whole data space. Data particles in different data classes produce different data gravitation fields, and these fields can be compared.

2.3. The law of data gravitation

The strength of gravitation between two data particles in data space is the direct ratio to the product of data mass of the two data particles, and inverse ratio to the square of distance between them. The law is described as follows:

$$F = \frac{m_1 m_2}{r^2}, \tag{5}$$

where F is the gravitation between two data particles; m_1 the data mass of data particle 1; m_2 the data mass of data particle 2; and r is the Euclidian distance between the two data particles in data space.

3. Classification based on data gravitation

3.1. Principle of classification

Lemma 2. *Suppose c_1 and c_2 are two data classes in a training data set. For a given test data element P (an atomic data particle), the gravitation that data particles in c_1 acts on P is F_1, and F_2 is the gravitation that data particles in c_2 acts on P. If $F_1 > F_2$, then the degree P belongs to c_1 is stronger than c_2.*

Fig. 2 describes the principle of classification.

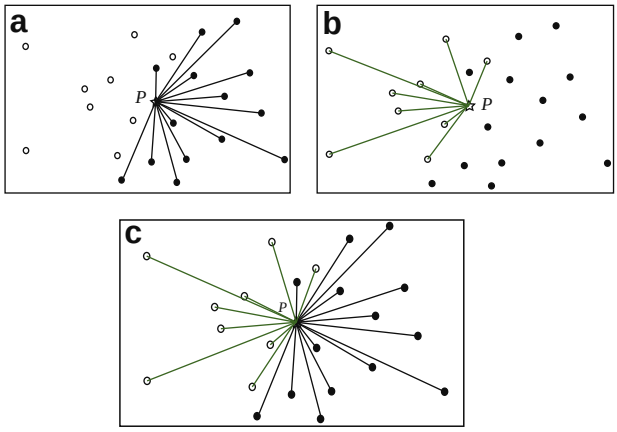

Fig. 2. Classification based on data gravitation. The strength of gravitation determines which class a test data element belongs to. The black dots denote data particles in class c_1. The circles denote data particles in class c_2.

Suppose $T = \{\langle \mathbf{x}_1, y_1\rangle, \langle \mathbf{x}_2, y_2\rangle, \ldots, \langle \mathbf{x}_l, y_l\rangle\}$ is a training set in n-dimensional data space, $y \in \{c_1, c_2, \ldots, c_k\}$, c_i represents data class i, k is the number of data classes, l is the number of training samples. A new set of training data particles is created from the original training set. The new training data particle set is $T' = \{\langle m_1, \mathbf{x}'_1, y_1\rangle, \langle m_2, \mathbf{x}'_2, y_2\rangle, \ldots, \langle m_l, \mathbf{x}'_l, y_l\rangle\}$, where l' is the number of data particles, $l' \leqslant l$, $\mathbf{x}'_i$ is the centroid of data particle i, m_i is the data mass of data particle i.

Once the training data particle set is constructed, the strength of data gravitation field on any position in the data space can be calculated. When a test data element is given, which data class it belongs to can be determined by the field strength of different data classes.

Suppose $c_1, c_2, \ldots, c_k$ are the data classes in the training set, they have $l_1, l_2, \ldots, l_k$ samples (data elements), the training data particle set is created from training set comprising of $l'_1 + l'_2 + \ldots + l'_k$ data particles, where l'_i is the number of data particles which belong to data class i. A given test data can be treated as an atomic data particle P, the centroid is its position $\mathbf{x}$. The gravitation that data class i acts on it is given by:

$$F_i = \sum_{j=1}^{l'_i} \frac{m_{ij}}{|\mathbf{x}_{ij} - \mathbf{x}|^2}, \tag{6}$$

where m_{ij} is the data mass of data particle j in data class i and $\mathbf{x}_{ij}$ is its centroid.

If $F_{i'} = max\{F_1, F_2, \ldots, F_k\}$, then according to lemma 2, the test data element belongs to data class i'.

3.2. Principles to create data particle

The simplest method to create data particle is to treat a single data element as one data particle. In other words, a training sample in the training data set can be used to create a data particle. According to how many data elements the original training data set possess, the required number of data particles are to be created. This method is simple and easy to realize. Another advantage of this method is that the field strength of data gravitation calculated by this method can achieve the highest accuracy. The weakness of the method is also obvious: The calculation might grow up tremendously with the expansion of the training data set and the efficiency of classification might be compromised.

Another method to create data particle is by using the maximum distance principle (MDP).

Algorithm 1

(1) For a given distance threshold ε and a training sample $\mathbf{x}_0$, if there are some other training samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ belonging to the same data class with $\mathbf{x}_0$, they make that $|\mathbf{x}_i - \mathbf{x}_0| < \varepsilon, i = 1, 2, \ldots, p$. Then $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ can be combined into a single data particle $\langle m_0, \mathbf{x}'_0\rangle$, where $\mathbf{x}'_0$ is centroid of the data particle, $m_0 = p + 1$ is its data mass.
(2) Repeat the step (1) above with the newly created data particle $(\mathbf{x}'_0, m_0)$ to find other training samples that match the condition. If found, combine them with the data particle, otherwise, this data particle is created finally.

Fig. 3 illustrates the main flowchart of MDP. The advantage of MDP method is that it can combine data elements that have similar influence on data gravitation field and hence the efficiency of classification can be enhanced remarkably. However, this method reduces the accuracy of the calculation of the strength of data gravitation field, especially on the area nearby the data particle centroid.

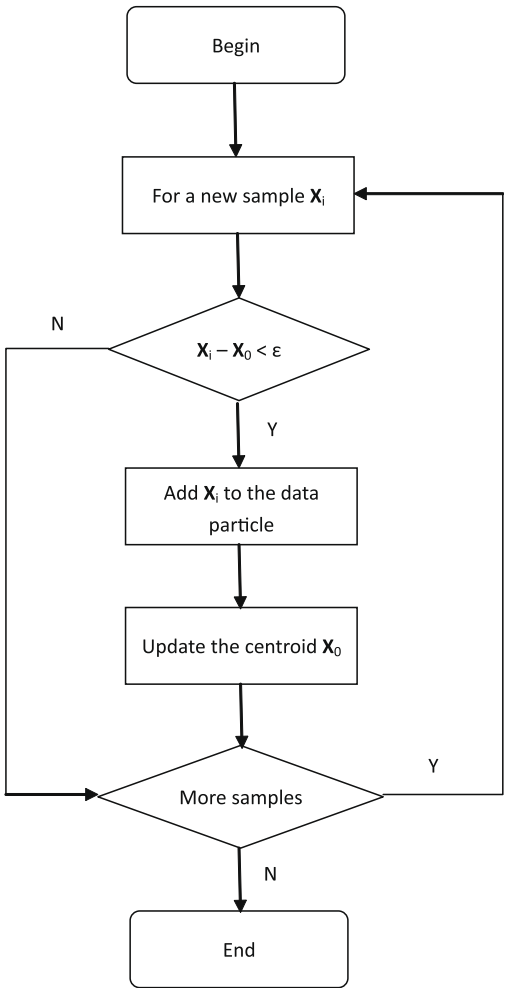

Fig. 3. Flowchart of maximum distance principle.

4. Feature selection

In real classification problems, a feature set must be defined to describe the target problem and the importance of each feature is often unknown. For many classification problems, irrelevant or redundant features decrease not only the classification efficiency, but also the accuracy [1]. So, feature selection is necessary for most classification problems. In fact, feature selection can be performed without reducing the accuracy of classification remarkably [3,6]. Choosing pivotal features can not only reduce the complexity of the target problem, but also improves the performance of algorithms in many cases [4].

4.1. Weighting to features

Most feature selection methods define feature selection using a binary description of selected/not-selected. That means a feature is useful or useless to the target problem. In many real world problems, the degree of importance of features is not identical and hence the binary description of feature selection cannot describe the importance of features accurately.

In this paper, we propose the concept of weighted features, and by weighting every feature of the target classification problem, the degree of importance of every feature can be obtained by its weight.

Suppose there are n features in the target problem feature set, every feature has a weight value, so all feature weights form a n-expression: $\langle w_1, w_2, \ldots, w_n \rangle$. We describe it as feature weight vector $\mathbf{w}$.

4.2. Tentative random feature selection algorithm

Since weight is a good descriptor for feature selection, deciding the weight of a certain feature is the major challenge. There are no general principles to calculate the weights of features. This paper propose a tentative random feature selection

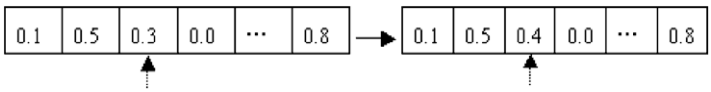

Fig. 4. Tuning of feature weights.

(TRFS) algorithm to calculate the weights of features by simulating the mutation operation in a genetic algorithm (GA). The main algorithm is as follows:

(1) Initialize the weight of each feature by a zero value.
(2) The weights of features are then optimized by an iterative procedure.
(3) The iteration of the weights of features is similar to mutation operation of GA.

First, randomly select a feature and add a small disturbance to its weight, then a new feature weight vector $\mathbf{w}'$ is obtained. We use training data set cross validation method to evaluate $\mathbf{w}$, if the result is better than the cross validation result of $\mathbf{w}$, then $\mathbf{w}'$ is observed and $\mathbf{w}$ is replaced by $\mathbf{w}'$, otherwise $\mathbf{w}'$ is discarded.

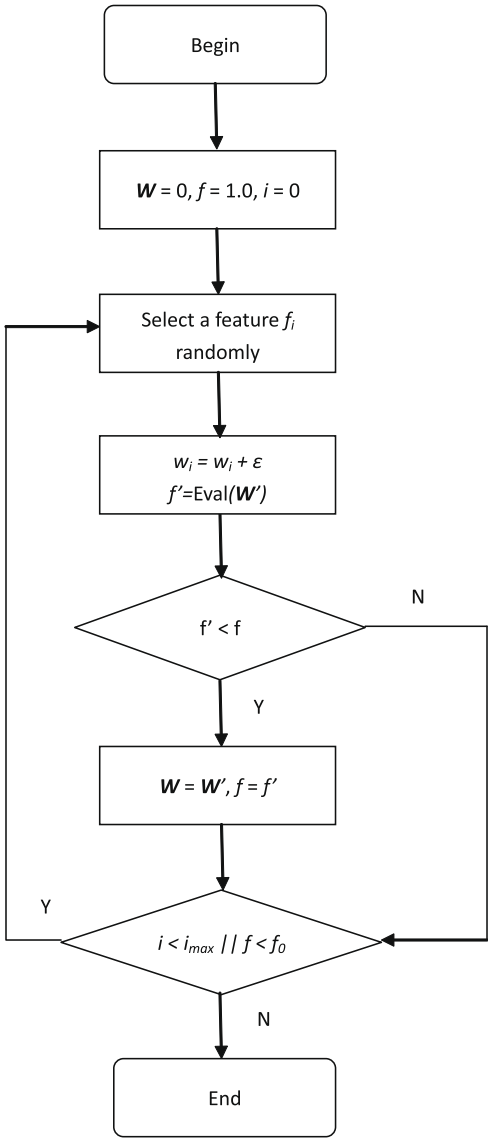

Fig. 5. Flowchart of tentative random feature selection.

Suppose the small disturbance is ε, $\mathbf{w} = \langle w_1, w_2, \ldots, w_n \rangle$, and the selected feature is i, then $\mathbf{w}' = \langle w_1, w_2, \ldots, w_i + \varepsilon, \ldots, w_n \rangle$, $-1 < \varepsilon < 1$. The tuning of weights is illustrated in Fig. 4 and the flowchart of tentative random feature selection algorithm is depicted in Fig. 5.

Each time the feature weight vector is updated, it should be evaluated. This paper uses cross validation of training data set to evaluate the newly obtained feature weight vector. In this method, the training set T is partitioned to two subsets T_a and T_b uniformly, the data elements in T_a and T_b are all selected from T randomly, but the ratio of every class in T_a and T_b should be kept identical to original training set T. According to DGC model, two data particle sets T'_a and T'_b are built according to T_a and T_b. We use T'_a as the training data particle set, T_b as testing set and classification accuracy is calculated using the DGC model, then T'_b as training data particle set, T_a as testing set and the classification accuracy is again calculated. By this method, the performance of the feature weight vector $\mathbf{w}$ can be evaluated. Suppose the feature set is $F = \langle f_1, f_2, \ldots, f_n \rangle$, the feature weight vector is $\mathbf{w} = \langle w_1, w_2, \ldots, w_n \rangle$, the training accuracy is defined as f and f_0 is target accuracy. Then the algorithm is described as follows:

Algorithm 2 (***TRFS***)

1: Split the training data set into two subsets T_a and T_b.
2: $\mathbf{w} = 0, f = 1.0$
3: for $i = 0$ to $i = i_{max}$ or $f < f_0$
4: select a feature f_i randomly
5: $w_i = w_i + \varepsilon$
6: evaluate $\mathbf{w}'$ using cross validation method on T_a and T_b, the result is f'
7: if $f' < f$
8: $\mathbf{w} = \mathbf{w}', f = f'$
9: end if
10: end for

4.3. Tuning of selection probability

If the features to be tuned were selected in a purely random way, then the probability that every feature were selected should be identical. That means the irrelevant or redundant features can gain the same chance of being selected as one of the pivotal features. Obviously this is not a good idea, since it is not biased in selecting the most contributing features. It is necessary to find an effective strategy to control the selection procedure.

In order to control the selection procedure, a probability tuning method is introduced. The initial probability of every feature to be selected is p_0, and all the probabilities form a selection probability vector $\mathbf{p} = \langle p_1, p_2, \ldots, p_n \rangle$. We define a "small" probability constant δ, and is used to tune the components' values of $\mathbf{p}$ in the selection procedure. δ is called as probability tuning constant, and $0 < \delta < 1$. The meaning of "small" is that the tuning of p is like micro tuning. When a feature i has been selected and its weight has been tuned, if the new weight vector is better than the old weight vector, then the corresponding probability component p_i are updated by adding δ to it, and the probability that the feature be selected is increased. Else if the new weight vector is worse than the old one, then the corresponding probability component p_i is decreased by a value of δ. So, the strategy stimulates the selection of good features and restrains the selection of bad features by tuning the probability vector. This strategy can be described in the following algorithm.

Suppose the feature set is $F = \langle f_1, f_2, \ldots, f_n \rangle$, the feature weight vector is $\mathbf{w} = \langle w_1, w_2, \ldots, w_n \rangle$, and the selection probability vector is $p = \langle p_1, p_2, \ldots, p_n \rangle$.

Algorithm 3 (*Tuning of selection probability*)

1: Feature selection procedure, f_i is selected, and its weight w_i is to be tuned.
2: w_i is tuned and $\mathbf{w}$ is updated. The new vector $\mathbf{w}'$ is evaluated.
3: if $\mathbf{w}'$ is better than $\mathbf{w}$
4: $\mathbf{w}$ is replaced by $\mathbf{w}'$.
5: $p_i = p_i + \delta$
6: else
7: $\mathbf{w}'$ is discarded and $\mathbf{w}$ is reserved.
8: if $p_i > \delta$
9: $p_i = p_i - \delta$
10: else
11: $p_i = 0$
12: end if
13: end if
14: Normalization of $\mathbf{p}$.

4.4. Calculation of data gravitation

In the DGC model, the data gravitation between two data particles is calculated by Eqs. (3) and (4). After the features in the feature set are weighted, the calculation method for data gravitation must be changed. As per the principles of data gravitation, the value of data gravitation is influenced by two factors: the data masses of the two data particles and the Euclidian distance between them. The data mass of a data particle is the number of data elements that belong to the data particle and the feature weight vector **w** cannot influence it. But **w** can influence the distance between the two data particles in the data space. As the features have been weighted, the distance between two data particles in the data space is not their Euclidian distance in the data space, but the weighted distance given below:

$$r' = \sqrt{\sum_{i=1}^{n} w_i (x_{1i} - x_{2i})^2}. \tag{7}$$

The calculation of the data gravitation is described as follows:

$$F' = \frac{m_1 m_2}{r'^2}. \tag{8}$$

5. Experiments and results

5.1. Results and analysis

Twelve well-known classification data sets were selected for the experiments, including iris data, glass data, segment data, vehicle data, wine data, vowel data, Pima Indian diabetes data, Wisconsin breast cancer data (WBCD), ionosphere data, hepatitis data, sonar data and zoo data. All these data sets are from the UCI machine learning repository, which are available from the web site: http://www.ics.uci.edu/~mlearn/databases/. Main characteristics of these data sets are depicted in Table 1, including number of data samples, number of features (inputs) and number of data classes. For comparison purposes, the first six data sets were selected according to [16] and [27], and the last two were selected according to [11] and [27].

First, all features were scaled for each data set. For a feature F in a data set, if the maximum value is f_{max} and the minimum value is f_{min}, original value of this feature of a sample is f. Then the feature value of this sample is scaled as follows:

$$f_{norm} = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{9}$$

In this paper, two widely used validation method were applied, 10-fold cross-validation and full train-full test (FT-FT). Ten-fold cross-validation was employed in [16] and [11], while FT-FT was used in [27].

Ten fold cross-validation. Each data set was split into 10 subsets randomly, and all samples of each class are uniformly assigned to these subsets. Then a subset was used as the testing set, the other nine subsets as a whole training set, one time validation was performed for this pair of training and testing set. The procedure was repeated ten times and each subset was used as test set for one time.

Full train-full test. The full data set was used for training, and the full data set was also used for testing. In [27], 10 times of experiments were performed with different model parameters. We also executed 10 times of experiments on the same data sets, but different executions generated feature weight vectors in a random phrase.

Table 2 illustrates the 10-fold cross-validation results using DGC algorithm. All empirical results for the eight data sets are listed, including the worst, the best and mean testing accuracy of each data set.

Table 1
Characters of used data sets.

Data set name	No. of samples	No. of features	No. of classes
Iris	150	4	3
Glass	214	9	7
Segment	2310	19	7
Vehicle	846	18	4
Wine	178	13	3
Vowel	528	10	11
Pima	768	8	2
WBCD	683	10	2
Ionosphere	351	34	2
Hepatitis	155	19	2
Sonar	208	60	2
Zoo	101	17	7

Table 3 compares the DGC results with the results reported in [16]. Since there is no worst and mean testing accuracies in [16], only best results were compared. As illustrated in Table 3, DGC achieved better accuracies than other classifiers for four out of six data sets, especially for glass data. DGC achieved testing accuracy of 90%, this is far higher than other classifiers. Also it can be seen in Table 2, the mean accuracy of DGC for glass data is higher than other classifiers. For segment and vehicle data, DGC and SGF network achieved basically the same level of accuracy, but lower than SVM. The benchmark results of these two data sets suggest that DGC and other classifiers have some blind spots, and therefore may not be able to perform as well in some cases.

Fuzzy integral-based perceptron is a two-class pattern classification model proposed in [11] and the authors used Pima Indian diabetes data and Wisconsin breast cancer data for experiments. Srinivasa et al. [23] proposed a self-adaptive migration model GA (SAMGA) and they used Pima and WBCD data set for experiments. We compared the DGC performance for Pima data and WBCD data with the results reported in [11,23] and Table 4 depicts the comparison results. DGC achieved the best accuracy for both data sets.

For the data sets of ionosphere, hepatitis, sonar and zoo, we compared our experimental results with several new classification technics. QPL is a customized classification learning method based on query projections [8]. Cohen et al. [5] presented a novel decision-tree instance-space decomposition method with grouped gain-ratio termed CPOM, and this method can be applied for classifiers such as neural network (CPOM-NN) and Naive-Bayes (CPOM-NB). Han et al. [8] used ionosphere, hepatitis, sonar and zoo data sets for the evaluation of their method, and [5] used sonar and zoo data sets for their experiments. DGC was used and the empirical results and comparison with SAMGA, QPL and CPOM-NB for the four data sets are illustrated in Tables 5 and 6.

Table 2
Ten fold cross-validation testing results using DGC.

Data set name	Worst (%)	Best (%)	Mean (%)
Iris	86.67	100.00	95.33
Glass	68.18	90.00	79.08
Segment	93.07	96.97	95.41
Vehicle	67.06	74.12	70.69
Wine	94.12	100.00	98.30
Vowel	94.44	100.00	98.49
Pima	68.83	81.82	76.56
WBCD	92.65	98.53	96.19
Ionosphere	86.11	94.29	90.63
Hepatitis	87.10	95.48	91.51
Sonar	85.71	95.45	90.85
Zoo	93.07	99.01	96.39

Table 3
Comparison of best testing results of six data sets.

Data set name	DGC (%)	SGF network [16] (%)	SVM (%)	1NN (%)
Iris	**100.00**	97.33	97.33	94.00
Glass	**90.00**	75.74	71.50	69.65
Segment	96.97	97.27	**97.40**	96.84
Vehicle	74.12	73.53	**86.64**	70.45
Wine	**100.00**	99.44	99.44	96.08
Vowel	**100.00**	99.62	99.05	99.43

Table 4
Comparison of best testing results of Pima and WBCD data.

Data sets	DGC (%)	Fuzzy integral-based perceptron [11] (%)	SAMGA [23] (%)	Decision tree-based fuzzy classifier (%)
Pima	**81.82**	74.81	73.00	73.05
WBCD	**98.53**	96.38	94.10	96.82

Table 5
Comparison of mean testing results of ionosphere and hepatitis data.

Data sets	DGC (%)	SAMGA [23] (%)	QPL [8] (%)
Ionosphere	**90.63**	86.20	90.30
Hepatitis	**91.51**	84.70	90.00

Table 6
Comparison of mean testing results of sonar and zoo data.

Data sets	DGC (%)	CPOM-NB [5] (%)	QPL [8] (%)
Sonar	**90.85**	76.44	87.00
Zoo	**96.39**	95.04	96.20

Table 7
FT-FT evaluation for glass data set.

Classification model	Worse (%)	Best (%)	Mean (%)
CF^1 rule-base classifier	54.20	66.35	59.58
CF^4 rule-base classifier	57.00	62.14	60.32
ROC rule-base classifier	60.28	70.09	65.18
DGC	**68.22**	**73.36**	**70.55**

Table 8
FT-FT evaluation for Pima data set.

Classification model	Worse (%)	Best (%)	Mean (%)
CF^1 rule-base classifier	67.83	73.30	69.64
CF^4 rule-base classifier	**69.92**	73.82	71.55
ROC rule-base classifier	73.95	75.00	**74.65**
DGC	65.10	**76.82**	72.45

Table 9
FT-FT evaluation for wine data set.

Classification model	Worse (%)	Best (%)	Mean (%)
CF^1 rule-base classifier	89.32	94.38	92.97
CF^4 rule-base classifier	89.88	96.06	94.26
ROC rule-base classifier	91.01	96.06	94.77
DGC	**98.31**	**99.44**	**98.99**

Zolghadri and Mansoori [27] used receiver operating characteristic (ROC) analysis for weighing fuzzy classification rules and glass, wine and Pima data were applied for the validation. Experiments were executed 10 times for each data set using full train-full test method. Authors used various number of fuzzy rules for each time, and the value of the number of fuzzy rules varied from 1 to 10. DGC was run 10 times for each data set using different parameter, such as the maximum radius of mass point. Tables 7–9 illustrate the comparison results.

5.2. Discussions

DGC model is able to obtain high classification performances in many cases, especially for the balanced data sets such as iris, wine and vowel data sets. For these data sets, the numbers of instances of different classes are approximately the same and the data gravitation is also balanced.

Our experiment results also reveal that if there exists a class, which includes extremely small or extremely large number of instances in the data set, the DGC approach may fail because of the imbalance of data gravitation. This imbalance means data gravitation from a certain class is extremely strong or extremely weak, so a test instance is always (not) classified to this class.

6. Conclusions

In this paper, a data gravitation based classification method with weighted feature is proposed. An important feature of DGC is its simple classification principle. By simulating the gravitation and the Law of Gravitation in the natural world, we proposed a novel classification technique, which is also very easy to implement. The paper also studied the feature selection problem of DGC. It was found that selecting good features is very important for DGC. By simulating the mutation operator in GA, the paper proposed a novel feature selection algorithm. Numerous experiments clearly illustrated that the feature selection algorithm is effective for DGC.

Twelve well-known UCI machine learning data sets were used for the experiments. For most of these data sets (iris, glass, wine, vowel, Pima and WBCD), DGC achieved better classification accuracies than several classical and new classification

methods. For the other data sets, DGC also achieved reasonably high accuracies. Empirical results show that DGC is really an effective classification method. Our experiment results also indicate that DGC suffers from imbalanced data, since it performed poorly for extremely unbalanced data sets. This is an important research topic in our future research.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Contract Numbers 60873089 and 60573065, the Natural Science Foundation of Shandong Province (Y2007G33 and Z2006G03), and the Key Subject(Laboratory) Research Foundation of Shandong Province XTD0709.

References

[1] A. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1997) 245–271.
[2] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, Pittsburgh, Pennsylvania, USA, 1992, pp. 144–152.
[3] S. Chebrolu, A. Abraham, J. Thomas, Feature deduction and ensemble design of intrusion detection systems, Computers and Security 24 (2005) 295–307.
[4] Y.H. Chen, B. Yang, J.W. Dong, A. Abraham, Time-series forecasting using flexible neural tree model, Information Science 174 (3/4) (2005) 219–235.
[5] S. Cohen, L. Rokach, O. Maimon, Decision-tree instance-space decomposition with grouped gain-ratio, Information Sciences 177 (2007) 3557–3573.
[6] M. Dash, H. Liu, Feature selection for classification, Intelligent Data Analysis 1 (3) (1997) 131–156.
[7] Y. Freund, Boosting a weak learning algorithm by majority, Information Computation 121 (1995) 256–285.
[8] Y. Han, W. Lam, C.X. Ling, Customized classification learning based on query projections, Information Sciences 177 (2007) 3557–3573.
[9] X. Hu, Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining applications, in: Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 2001, pp. 233–240.
[10] X. Hu, N. Cercone, Data mining via discretization, generalization and rough set feature selection, Knowledge and Information System 1 (1) (1999) 33–60.
[11] Y.C. Hu, Fuzzy integral-based perceptron for two-class pattern classification problems, Information Sciences 177 (2007) 1673–1686.
[12] M. Indulska, M.E. Orlowska, Gravity based spatial clustering, in: A. Voisard, S. Chen (Eds.), Tenth International Symposium on Advances in Geographic Information Systems, McLean, Virginia, USA, 2002, pp. 125–130.
[13] H. Lu, S. Rudy, H. Liu, Effect data mining using neural networks, IEEE Transaction on Knowledge and Data Engineering 8 (6) (1996) 957–961.
[14] S. Minz, R. Jain, Rough set based decision tree model for classification, Lecture Notes in Computer Science 2737 (2003) 172–181.
[15] S. Minz, R. Jain, Hybridized rough set framework for classification: an experimental view, Design and Application of Hybrid Intelligent Systems (2003) 631–640.
[16] Y.J. Oyang, S.C. Hwang, Y.Y. Ou, C.Y. Chen, Z.W. Chen, Data classification with the radial basis function network based on a novel kernel density estimation algorithm, IEEE Transactions on Neural Networks 16 (1) (2005) 225–236.
[17] A. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, 1991.
[18] L.Z. Peng, Y.H. Chen, B. Yang, Z.X. Chen, A novel classification method based on data gravitation, in: Proceedings of International Conference on Neural Networks and Brain, Beijing, China, 2005, pp. 667–672.
[19] J.F. Peters, A. Skowron, A rough set approach to knowledge discovery, International Journal of Intelligent Systems 17 (2002) 109–112.
[20] J.R. Qinlan, Introduction of decision trees, Machine Learning 1 (1986) 86–106.
[21] Y. Shi, A. Zhang, A shrinking-based dimension reduction approach for multi-dimensional data analysis, in: The 16th International Conference on Scientific and Statistical Database Management, Santorini Island, Greece, 2004, pp. 427–428.
[22] Y. Shi, Y. Song, A. Zhang, A shrinking-based approach for multi-dimensional data analysis, in: The 29th VLDB Conference, Berlin, Germany, 2003, pp. 440–451.
[23] K.G. Srinivasa, K.R. Venugopal, L.M. Patnaik, A self-adaptive migration model genetic algorithm for data mining applications, Information Sciences 177 (2007) 4295–4313.
[24] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer Verlag, 1995.
[25] B. Webster, P.J. Bernhard, A local search optimization algorithm based on natural principles of gravitation, in: Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE'03), Las Vegas, Nevada, USA, 2003, pp. 255–261.
[26] B. Yang, L.Z. Peng, Y.H. Chen, A DGC-based data classification method used for abnormal network intrusion detection, Lecture Notes in Computer Science 4232 (2006) 209–216.
[27] M.J. Zolghadri, E.G. Mansoori, Weighting fuzzy classification rules using receiver operating characteristics (ROC) analysis, Information Sciences 177 (2007) 2296–2307.