International Journal of Database Theory and Application
Vol.9, No.3 (2016), pp. 211-228
http://dx.doi.org/10.14257/ijdta.2016.9.3.21

A Novel Feature Selection Based Gravitation for Text Categorization

Jieming Yang*, Zhiying Liu and Zhaoyang Qu

College of Information Engineering, Northeast Dianli University, Jilin, Jilin, China
yjmlzy@gmail.com

Abstract

The high dimensionality of feature space is a big hurdle in applying many sophisticated methods to text categorization. The feature selection method is one of methods which reduce the high dimensionality of feature space. In this paper, we proposed a new feature selection algorithm based on gravitation, named GFS, which regards a feature occurring in one category as an object, and all objects corresponding to a feature occurring in various categories can constitute a gravitational field, then the gravitation of a feature with unknown category label on which all objects in the gravitational field act is used for feature selection. We have evaluated GFS on three benchmark datasets (20-Newgroups, Reuters-21578 and WebKB), using two classification algorithms, Naïve Bayes (NB) and Support Vector Machines (SVM), and compared it with four well-known feature selection algorithms (information gain, document frequency, orthogonal centroid feature selection and Poisson distribution). The experiments show that GFS performs significantly better than other feature selection algorithms in terms of micro F1, macro F1 and accuracy.

Keywords: *text categorization; feature selection; gravitation; high dimensionality*

1. Introduction

Automatic text categorization, which assigns the predefined categories to new text documents based on the content of the document, is the viable method to deal with the scaling problem of the digital documents. In real world situations, the text categorization has many important characteristics. A major characteristic of text categorization is the high dimensionality of the feature vector space which can be tens and hundreds of thousands of terms for even a moderated size dataset [1, 2]. Another major characteristic of text categorization is the high level of feature redundancy and feature irrelevance [2]. The irrelevant feature does not affect the performance of the classifiers and the redundant feature does not add any new performance to the problem of text categorization[3]. The high dimensionality is a big hurdle in applying many sophisticated learning algorithms to the text categorization [4]. Furthermore, the irrelevant and redundant features not only slow down the classification process and hurt the performance of the classifier but also bring about overfitting. Hence the feature selection is one of the methods that solve the problem mentioned above.

In recent years, feature selection has been a hotspot to which many researchers pay attention. There are four definitions from various views. The first one is idealized that finds the minimally sized feature subset that is necessary and sufficient to problem [5]; the second is classical that selects a subset of the original features, such that the value of a criterion function is optimized [6]; the third is to choose a subset of features to improve the prediction accuracy and decrease the size of the feature space; the last one is that selects a small subset such that the resulting class distribution, given the selected features,

* Corresponding Author

ISSN: 2005-4270 IJDTA

is as close as possible to the original class distribution [7]. Dash and Liu [3] considered the factors mentioned above, and believed that the feature selection attempts to select the minimally sized subset of features according to the two criteria: (1) the classification accuracy does not significantly decrease; (2) the resulting class distribution given the selected features is as close as possible to the original class distribution.

There are four basic steps in a typical feature selection method, such as generation procedure, evaluation function, stopping criterion and validation procedure [3]. During the four steps of feature selection algorithm, the evaluation function is a vital one. It tries to measure the discriminating ability of a feature or a subset to distinguish the different class labels [3]. Blum & Langley [8] grouped the feature selection methods into three classes: embed, wrapper, and filtering. The characteristics of the embed approach is that the feature selection process is clearly embedded in the basic induction algorithm. The wrapper approach is to select feature subset using the evaluation function as a wrapper around the learning algorithm, and these features will be used on the same learning algorithm [9, 10]. The filtering approach selects the feature subset using the evaluation function that is independent to the learning method [9]. The most popular and computationally fast feature selection is the filtering approach [4], and the proposed method GFS in this study is also a filtering approach. There are numerous well-known feature selection algorithms, such as document frequency (DF), information gain (IG), χ^2-statistic [11],odds ratios (OR) [12], mutual information [11], bi-normal separation (BNS) [13], Best Terms [4], the Orthogonal Centroid Feature Selection (OCFS) [14], the most relevant with category [15, 16], improved Gini index [17], class discriminating measure (CDM) [18], measure using Poisson distribution [19], Bi-Test [20], and so on. Most of these feature selection algorithms calculate the score of a feature for categorization based on information theory, probability and mathematical statistics, then all of the features in the training set are ranked and the top k features are selected to form the reduced feature space.

In this paper, we proposed a new feature selection based on the theory of universal gravitation, named GFS, which assumes that a feature in every category of training set is an object and the amount of this feature occurring in every category of training set is the mass of the object. So a feature occurring in all the categories form a gravitation field, and then the gravitation of a feature in this gravitation field can be calculated. If the gravitation of category c_i acting on a feature is bigger, this feature contains more information for category c_i. To evaluate GFS method, we used two classification algorithms, Naïve Bayes (NB) and Support Vector Machines (SVM) on three benchmark text corpora (20-Newgroups, Reuters-21578 and WebKB) and compared it with four feature selection algorithms (information gain, document frequency, the orthogonal centroid feature selection and Poisson distribution).The experiments show that GFS performs significantly better than other feature selection algorithms in terms of micro F1, macro F1 and accuracy.

The rest of this paper is organized as follows: Section 2 presents the state of the art for feature selection methods. Section 3 describes and analyzes the basic principle and implementation of the proposed method. The experimental details are given in Section 4 and the experimental results are listed in the Section 5. The statistical analysis and discussion are presented in Section 6. Our conclusion and future work direction are provided in the last Section.

2. Related Work

2.1. Information Gain

Information Gain (IG) [21] is frequently used as a criterion in the field of machine learning [11]. The information gain of a given feature t_k with respect to the class $c_i$is the

reduction in uncertainty about the value of c_i when we know the value of t_k. The larger the value of a feature information gain is, the more significant for categorization the feature is. The information gain of a feature t_k toward a category c_i can be defined as follows:

$$IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t,c) \log \frac{P(t,c)}{P(t)P(c)} \quad (1)$$

Where t_k is one of the all features and c_i is one of the all classes; $P(c)$ is the fraction of the documents in category c over the total number of documents;$P(t,c)$ is the fraction of documents in the category c that contain the word t over the total number of documents;$P(t)$ is the fraction of the documents containing the term t over the total number of documents [22]

2.2 Document Frequency

Document Frequency (DF) is a simple and effective feature selection method, and it computes the number of documents in which a term occurs. The basic idea is that the rare terms are not useful for category predication and maybe degrade the global performance [11]. So if the number of the documents in which a term occurs is the largest, the term is retained [23]. The document frequency of a term is calculated as follows:

$$DF(t_k, c_i) = P(t_k \mid c_i) \quad (2)$$

where $P(t_k| c_i)$ is the conditional probability ofthe feature $t_k$given the categoryc_i.

2.3. Orthogonal Centroid Feature Selection

The Orthogonal Centroid Feature Selection (OCFS) selects features optimally according to the objective function implied by the Orthogonal Centroid algorithm [14, 16]. The centroid of each class and all training samples are firstly calculated, and then the score of the term is calculated according to the centroid of the each class and the entire training set. The higher the score of the term is, the more category information the term contains. The score of a term t_k is calculated as follows:

$$OCFS(t_k) = \sum_{j=1}^{|C|} \frac{n_j}{n} (m_j^k - m^k)^2 \quad (3)$$

Where n_j is the number of documents in the category c_j;n is the total number of documents in the training set; m_j^k is the value of the k-th element of the centroid vector m_j of category c_j;m^k is the value of the k-th element of the centroid vector m of entire training set; $|C|$ refers to the total number of categories in the corpus.

2.4. Measure Using Poisson Distribution

χ_p^2 is derived from Poisson distribution and applied to information retrieval. The main idea is that the degree of deviation from the Poisson distribution is used as a measure of effective features [19], that is, the farther a feature departs from Poisson distribution, the more effective the feature is. Conversely, if a feature can be predicted by Poisson distribution, then the feature is poor.

$$\chi_p^2(t_i, C_j) = \frac{(a_{ij} - \hat{a}_{ij})^2}{\hat{a}_{ij}} + \frac{(b_{ij} - \hat{b}_{ij})^2}{\hat{b}_{ij}} + \frac{(c_{ij} - \hat{c}_{ij})^2}{\hat{c}_{ij}} + \frac{(d_{ij} - \hat{d}_{ij})^2}{\hat{d}_{ij}}$$

$$\hat{a}_{ij} = n(C_j)(1 - e^{-\lambda_i})$$

$$\hat{b}_{ij} = n(C_j) e^{-\lambda_i}$$

$$\hat{c}_{ij} = n(\overline{C}_j)(1 - e^{-\lambda_i})$$
$$\hat{d}_{ij} = n(\overline{C}_j)e^{-\lambda_i}$$
$$\lambda_i = \frac{F_i}{N} \quad (4)$$

Where a_{ij} is the frequency of feature t_i and class C_j co-occurrence ;b_{ij} is the frequency of feature t_i occurs that does not belong to class C_j;c_{ij} is the frequency of class C_j occurrence that does not contain feature t_i;d_{ij} is the number of times neither C_j nor t_i occurs; F_i is the total frequency of term t_i in all messages; $n(C_j)$ and $n(\overline{C}_j)$ are the numbers of messages belonging to C_j and not belonging to C_j ,respectively. N is the total number of documents in the training set.

2. Algorithm Description

2.1. Activation

The universal law of gravitation was firstly discovered by Newton in 1687. It indicates that the strength of gravitation between two objects is directly proportional to the product of the masses of the two objects and inversely proportional to the square of the distance between them. The law can be described as follows:

$$F = G\frac{m_1 m_2}{r^2} \quad (5)$$

Where F is the gravitation between two objects; G is the constant of universal gravitation; m_1 is the mass of the object 1; m_2 is the mass of the object 2; r is the distance between two objects.

In recent years, the universal law of gravitation was introduced to the machine learning, such as gravitational clustering [24-26], data gravitation based classification [27], gravitational search algorithm [28-29], and so on. Guo *et al* [30] proposed a feature selection algorithm based on K-gravity clustering. They first grouped interdependent features into clusters used gravitational attraction and then used Embedded Classification Learning (ECL) to pick up some top feature groups and build a classifier on each selected feature group.

Inspired by the universal law of gravitation and its utility mentioned above, we think that a feature occurring in a category can be assumed as an object, and the amount of this feature occurring in a category can be regard as the mass of this object. Therefore, these objects corresponding to that a feature occurring in various categories can form a gravitational field. We assume that all the objects in the gravitational field will attract a new object (feature with unknown category). If the force that an object for category c_i acted on this new object is the biggest, the object (feature) contains most categorization information for category c_i. Take features listed in Table 1 as an example, there are 10 categories and each feature in Table 1 will form one gravitational field that contains 10 objects. We assume that there is a feature “home” from one document with unknown category and the force acted on this object (feature) by object in category C1 is the biggest in the “home” gravitational field, so the feature “home” can well represent for category C1. According to this theory, the feature “remodeling” contains more information of category C4 and the feature “sales” is stand for category C7.

Table 1. The Term Frequency of Features Occurring in Every Category

features	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
home	230	9	14	31	20	4	7	4	4	16
remodeling	478	7	36	699	15	4	6	4	12	4
sales	80	16	61	75	35	31	138	14	54	15

2.2. Algorithm Implement

In order to apply the law of universal gravitation for feature selection, the parameters in the formula of the gravity should be determined.

Definition 1 (data object o_i^k). Data object is similar to the object in physics, which also has "data mass". Data object is defined as a feature t_k occurring in one category. As far as a feature is concerned, the number of data objects is the same as the categories. For instance, there are n data objects ($o_1^k, o_2^k, ..., o_n^k$) for a feature t_k in the classification problem which have n categories ($c_1, c_2, \ldots, c_n$).

Definition 2 (data mass m_i^k). The mass of a data object o_i^k is the term frequency of a feature t_k occurring in the category c_i.

Definition 3 (atomic data object o_a^k). The atomic data object is a feature t_k that comes from a document with unknown category. Since the mass of the atomic data object is same for each data object in the gravitation field, the mass of the atomic data object is a constant. In this paper, the mass of the atomic data object (m_a^k) is assigned by 1.

Definition 4 (data gravitational field). All the data objects of a feature t_k form the data gravitational field. All the data objects in this gravitational field can yield an attraction force on atomic data object entering into this field. If the force of the data object o_i^k acting on the atomic data object o_a^k is the biggest, this feature t_k corresponding to the data object o_i^k and atomic data object contains most categorization information for category c_i.

Definition 5 (object distance r_i). The object distance is the distance between the object o_i^k and the atomic data object o_a^k. To simplify the calculation, we define the square of the object distance (r_i) as follows:

$$r_i^2 = \frac{tf_i}{\frac{|C|}{\min_{i=1}(tf_i)}} \tag{6}$$

Where tf_i is the amount of term frequency of all features occurring in category c_i; $|C|$ is the number of categories.

According to the notions defined above, the strength of gravitation between data object o_i^k and the atomic data object o_a^k is directly proportional to the product of data mass m_i^k of data object o_i^k and the mass m_a^k of the atomic data object o_a^k and inverse proportional to the square of the object distance. The formula is described as follow:

$$F(t_k, c_i) = G\frac{m_i^k m_a^k}{r_i^2} \tag{7}$$

Where $F(t_k,c_i)$ is the gravitation between data object o_i^k and the atomic data object o_a^k; m_i^k is the data mass of data object o_i^k; m_a^k is the data mass of the atomic data object o_a^k; r_i is the distance between data object o_i^k and atomic data object o_a^k; G is a constant and assigned 1 inthis paper. We use the maximum of the $F(t_k,c_i)$ as the global significance of a feature t_k in classification problem. The formula is defined as follow:

$$F(t_k) = \max(F(t_k, c_i)) \tag{8}$$

The details of the GFS algorithm are given as following:

Algorithm 1

Input: V– the original vocabulary of the features extracted from the training set
C– the predefined category set

k– the number of the selected features

Output: V_{sub}– the feature subset of V which contains the best features for categorization

Step 1: for each feature in $v_k \in V$ ($0<k\leqslant|V|$)

Step 2: calculate the term frequency of feature v_k in each catetory c_i (tf_i^k).

Step 3: calculate the sum of term frequency of all features in each category c_i ($tf_i = tf_i + tf_i^k$).

Step 4: end

Step 5: for each category $c_i \in C$ ($0<i\leqslant|C|$)

Step 6: calculate the minimal value of tf_i (tf_{min})

Step 7: end

Step 8: for each feature in $v_k \in V$ ($0<k\leqslant|V|$)

Step 9: for each category $c_i \in C$ ($0<i\leqslant|C|$)

Step 10: calculate the gravitational force ($F(t_k, c_i) = \frac{m_i^k m_a^k}{r_i^2}$,

$m_i^k = tf_i^k$, $m_a^k = 1, r_i^2 = \frac{tf_i}{tf_{\min}}$)

Step 11: end

Step 12: calculate the maximal value of $F(t_k, c_i)$ ($F(t_k) = \max(F(t_k, c_i))$)

Step 13: end

Step 14: ranks all features in V based on $F(t_k)$

Step 15: selects top *k* features into V_{sub}

3. Experimental Setup

3.1. Validation

There are two commonly used validation procedures for feature selection methods: (a) using artificial datasets and (b) using real-world datasets [3]. In this paper, we choose three real-world datasets that are benchmark datasets. If the dataset is large, it can be split into *K* parts, and then each part is randomly divided into two subsets; the one subset is for training and the other one is for testing. Unfortunately, the dataset in real world is usually limited. So the 10-fold cross validation was adopted in our experiment. The 10-fold validation produces fairly good estimates in small dataset [31-32]. The dataset is randomly split into 10 mutually exclusive subsets of approximately equal size. The classifier is trained and tested 10 times; each time $t \in \{1,2,\ldots,10\}$, it is trained on $\{D\text{-}D_t\}$ and tested on D_t [32]. When the dataset is split, we ensure that each subset contains approximately the same proportions of labels as the original dataset.

3.2. Datasets

In order to evaluate the performance of the proposed method, three benchmark datasets - 20-Newsgroups, Reuters-21578 and WebKB - were used in this paper. The documents in every corpus must be transformed into succinct representations suitable for the classifiers during the process of preprocessing [33]. In our experiment, all the words were converted to lower case, punctuation marks were removed, and stop lists were used but do not stem. Document frequency of a term was applied in text representation.

The 20-Newgroups were collected by Ken Lang (1995) and has become one of the standard corpora for text categorization. It contains 19997 newsgroup postings, and all documents were assigned evenly to 20 different UseNet groups. We ignore the UseNet header and only consider the content of the document when tokenizing the document.

The Reuters-21578 corpus contains 21578 stories taken from the Reuters newswire. All stories are non-uniformly divided into 135 categories. In this paper, we only consider the top 10 categories such as “Earn”, “Acquisition”, “Money-fx”, “Grain”, “Crude”, “Trade”, “Interest”, “Wheat”, “Ship” and “Corn”. The 9982 documents in the top 10 categories are split into 10 parts, and then 9 parts is used to train, the rest part is used to test.

The World Wide Knowledge Base dataset (WebKB), which was collected by Craven *et al*.[34], is a collection of web pages from four different college web sites. The 8282 web pages are non-uniformly assigned to 7 categories. Following Nigam *et al*. [35] we selected 4 categories, “course”, “faculty”, “project” and “student”, as our corpus.There are 4199 documents in 4 categories. Since the documents in WebKB corpus are HTML format files and contain much non-textual information, we remove all the HTML tagsin the documents during the preprocessing.

3.3. Text Representation

The representation of document is a important aspect in text categorization [23]. One of most popular representations is commonly referred to as the bag of words (BOW). Each document in the corpus is represented as a N-dimensional feature vector $X = [x_1, x_2, \ldots, x_N]$, where the value of x_i is determined by the representation of features adopted [36]. There are many feature representation methods used in text categorization, such binary representation, term frequency representation and term frequency-inverse document frequency (*tf-idf*). We adopted the term frequency representation in our experiment. It assigns x_i as the number of occurrences of a term t_i in a document.

3.4. Classifiers

Many classifiers are used in text categorization in recent years, such as Naïve Bayes (NB), K-nearest neighbor (KNN), Support Vector Machines (SVM), decision tree, and so on. The Naïve Bayes classifier, which is one of the most extensively used machine learning methods, is popular in text categorization. It successful applications to text document datasets have been shown in many literatures [22]. the Naïve Bayes classifier is simple to be performed and no parameters need to be adjusted [22]. Compared to the state-of-art methods, Support Vector Machines is a higher efficient classifier in text categorization. There is much empirical support for using Support Vector Machines for text categorization [19, 37-38]. In this paper, we use NB and SVM classifier to compare the performance of various feature selection methods.

The Naïve Bayes [39] is a classifiable algorithm based on the assumption that a term occurring in a document is independent from the occurrence of other terms. There are two commonly used models about Bayesian classifier, the one is a multinomial model and the other is the multivariate Bernoulli model. Schneider [40] indicated that multinomial model can generate higher accuracy than multivariate Bernoulli model. In this study, we use multinomial model.

Support Vector Machines are based on the structural risk minimization principle for computational learning theory, and it was originally developed by Drucker, *et al*. [41] and applied to text categorization by Joachims [1]. Since Joachims [1] thought that most text categorization problems are linearly separable, the linear kernel for SVM is selected. In this study, we use LIBSVM toolkit [42], and the C-SVM [43-44] is selected and the penalty parameter C is 1.

3.5. Evaluations

The classification effectiveness in text categorization is usually measured in terms of the precision (P) and recall (R) [23] which are originally defined for binary classification [37]. The precision is the ratio of the number of messages which are correctly identified as the positive category to the amount of messages which are identified as the positive

category, and the recall is the ratio of the number of the messages which are correctly identified as the positive category to the amount of the messages which actually belong to the positive category. If we consider the category c_i as the target category, the precision (P_i) and the recall (R_i) are defined as follow.

$$P_i = \frac{TP_i}{TP_i + FP_i} \qquad R_i = \frac{TP_i}{TP_i + FN_i} \tag{9}$$

Where TP_i is the number of the documents that is correctly classified to category c_i, FP_i is the number of the documents that is misclassified to the category c_i, FN_i is the number of the documents belonging to category c_i were misclassified to other categories.

To compute the averaged estimates in multiclass classification context, the macro-averaging and micro-averaging methods are used [19]. The micro-averaged F1 and the macro-averaged F1 measure are computed as follow.

$$P_{micro} = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \qquad R_{micro} = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)}$$

$$F1_{micro} = \frac{2 P_{micro} R_{micro}}{P_{micro} + R_{micro}}$$

$$P_{macro} = \frac{\sum_{i=1}^{|C|} P_i}{|C|} \qquad R_{macro} = \frac{\sum_{i=1}^{|C|} R_i}{|C|}$$

$$F_{macro} = \frac{2 R_{macro} P_{macro}}{R_{macro} + P_{macro}} \tag{10}$$

Where $|C|$ is the number of the categories.

The accuracy, which is defined to be the percentage of correctly labeled documents in test set, is widely used in text categorization [22, 38, 45-47]. The formula of the accuracy is defined as follow.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

4. Results

In the experiment, we selected top k features from the original feature space to compare the performance of feature selection algorithms. The value of k is equal to 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 and 2000.

Table 2. The Micro F1 of Naïve Bayes Used Five Feature Selections on 20-Newsgroups. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	**200**	**400**	**600**	**800**	**1000**	**1200**	**1400**	**1600**	**1800**	**2000**
GFS	**66.15**	**71.33**	**73.84**	**75.00**	**75.89**	**76.65**	**76.92**	**77.27**	**77.63**	**78.01**
IG	43.05	49.55	54.83	57.76	60.70	62.88	64.58	65.99	67.40	68.34
DF	48.19	59.23	63.47	66.37	68.58	70.32	71.90	72.78	73.49	74.07
OCFS	36.10	46.50	51.87	55.37	58.24	60.64	62.81	64.72	65.84	66.87
XP2	62.33	67.36	70.08	71.41	72.40	73.11	73.56	74.14	74.47	74.77

Table 3. The Macro F1 of Naïve Bayes Used Five Feature Selections on 20-Newsgroups. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	200	400	600	800	1000	1200	1400	1600	1800	2000
GFS	**63.47**	**69.41**	**71.93**	**73.26**	**74.12**	**75.00**	**75.38**	**75.77**	**76.19**	**76.60**
IG	37.99	44.78	50.16	53.44	56.63	59.21	61.03	62.75	64.48	65.55
DF	43.03	54.70	59.47	63.05	65.63	67.84	69.64	70.60	71.35	72.01
OCFS	30.44	41.63	46.98	50.61	53.97	56.77	59.28	61.41	62.73	63.89
XP2	59.12	64.58	67.61	69.03	70.21	71.05	71.63	72.30	72.73	73.03

Table 4. The Micro F1 of Support Vector Machines Used Five Feature Selections on 20-Newsgroups. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	200	400	600	800	1000	1200	1400	1600	1800	2000
GFS	**71.08**	**73.04**	**73.10**	**72.90**	**72.76**	**73.02**	**72.89**	**73.43**	**73.35**	**73.66**
IG	53.51	58.15	60.30	62.53	64.04	65.17	66.24	67.28	67.66	68.45
DF	57.00	63.57	65.28	66.69	68.37	69.57	70.51	71.23	71.55	72.06
OCFS	47.90	54.87	57.78	60.30	62.44	63.59	65.08	66.36	66.72	67.61
XP2	68.16	69.97	71.01	71.21	71.83	72.00	72.13	72.50	72.82	72.92

4.1. Results on 20-Newsgroups Corpus

Table 2 and Table 3 shows the comparison of micro-averaged F1 and macro-averaged F1 among the different feature selection algorithms when NB classifier was used on 20-Newgroups, respectively. It can be seen from Table 2 that the micro F1 measure of NB with GFS on 20-Newgroups is superior to that with other four feature selections. Table 3 indicates that the macro F1 of NB combined with GFS outperforms that combined with other four feature selections. Table 4 and Table 5 show the comparison of micro-averaged F1 and macro-averaged F1 among the different selection algorithms when SVM classifier was used on 20-Newgroups, respectively. When SVM is used on 20-Newgroups, the proposed method GFS is superior to other feature selection algorithms in terms of micro F1 and macro F1. It can be seen from Table 2 – 5 that the XP2 acquires the second highest performance in terms of micro F1 and macro F1 only inferior to that of GFS. Figure 1 lists the accuracy curve of NB and SVM combined with five feature selections used on 20-Newgroups. Figure 1(a) shows that the curve of NB combined with GFS is significantly higher than that with other methods. Figure 1(b) indicates that the curve of SVM combined with GFS used on 20-Newgroups is higher than that with other four methods.

Table 5. The Macro F1 of Support Vector Machines Used Five Feature Selections on 20-Newsgroups. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	200	400	600	800	1000	1200	1400	1600	1800	2000
GFS	**70.26**	**72.57**	**72.71**	**72.55**	**72.40**	**72.66**	**72.50**	**73.05**	**73.00**	**73.29**
IG	52.72	57.46	59.64	61.98	63.51	64.70	65.75	66.87	67.22	68.02
DF	56.32	63.01	64.82	66.25	67.96	69.20	70.12	70.85	71.17	71.69
OCFS	47.18	54.25	57.22	59.80	62.01	63.15	64.65	65.92	66.27	67.17
XP2	67.45	69.45	70.54	70.76	71.35	71.56	71.72	72.11	72.40	72.51

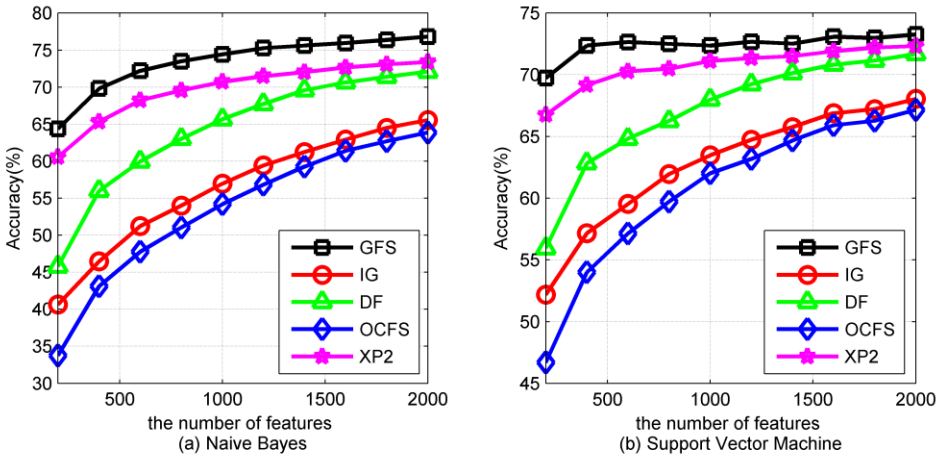

Figure 1. The Accuracy Curve of Naïve Bayes and Support Vector Machines Used Five Feature Selections on 20-Newsgroups, Respectively

4.2. Results on Reuters-21578 Corpus

Table 6 and Table 7 shows the comparison of micro-averaged F1 and macro-averaged F1 among the different selection algorithms when NB classifier was used on Reuters-21578, respectively. It can be seen from Table 6 that the micro F1 measure of NB with GFS on Reuters-21578 is superior to that with other four feature selections. Table 7 indicates that the macro F1 of NB combined with GFS outperforms that combined with other four feature selections. Table 8 and Table 9 show the comparison of micro-averaged F1 and macro-averaged F1 among the different selection algorithms when SVM classifier was used on Reuters-21578, respectively. When SVM is used on Reuters-21578, the proposed method GFS is superior to other feature selection algorithms in terms of micro F1 and macro F1 except for the number of selected features is 1600, 1800 and 2000. Figure 2 lists the accuracy curve of NB and SVM combined with five feature selections used on Reuters-21578. Figure 2(a) shows that the curve of NB combined with GFS is significantly higher than that with other methods and reaches the highest point (84.71%) when the number of selected features is 1000. Figure 2(b) indicates that the curve of SVM combined with GFS used on Reuters-21578 is higher than that with IG, OCFS and XP2, and lower than that with DF when the number of selected features is 1000, 1200 and 1600.

Table 6. The Micro F1 of Naïve Bayes Used Five Feature Selections on Reuters-21578. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	**200**	**400**	**600**	**800**	**1000**	**1200**	**1400**	**1600**	**1800**	**2000**
GFS	**64.97**	**65.63**	**66.43**	**66.50**	**66.75**	**66.88**	**66.59**	**66.42**	**66.30**	**66.10**
IG	59.37	62.09	63.86	64.60	64.59	64.76	64.58	65.11	65.17	65.22
DF	58.05	62.41	63.27	64.09	64.97	64.99	65.31	65.36	65.41	65.51
OCFS	55.20	60.90	62.41	63.43	63.96	64.41	64.59	64.39	64.57	64.96
XP2	57.09	57.20	57.05	57.04	57.00	57.03	57.05	57.05	57.09	57.03

Table 7. The Macro F1 of Naïve Bayes Used Five Feature Selections on Reuters-21578. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	200	400	600	800	1000	1200	1400	1600	1800	2000
GFS	**60.13**	**60.70**	**61.70**	**61.95**	**62.19**	**62.40**	**62.40**	**62.43**	**62.36**	**62.30**
IG	52.67	55.23	57.40	58.57	58.70	59.32	59.46	60.06	60.32	60.56
DF	51.68	55.29	56.33	58.16	59.23	59.69	60.30	60.52	60.68	60.99
OCFS	49.09	53.80	55.72	56.95	57.67	58.36	58.84	59.26	59.74	60.25
XP2	51.44	51.56	51.45	51.43	51.45	51.64	51.66	51.65	51.66	51.59

Table 8. The Micro F1 of Support Vector Machines Used Five Feature Selections on Reuters-21578. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	200	400	600	800	1000	1200	1400	1600	1800	2000
GFS	**63.96**	**63.16**	**63.06**	**62.97**	**62.95**	**62.72**	**62.76**	62.67	62.75	62.74
IG	62.19	61.31	61.73	62.19	62.20	62.66	62.59	62.59	62.58	**62.88**
DF	60.65	61.07	61.76	62.75	62.81	62.69	62.56	**62.70**	**62.86**	62.82
OCFS	60.23	60.28	61.02	61.49	62.26	62.69	62.64	62.52	62.73	62.69
XP2	61.55	61.56	61.47	61.39	61.43	61.27	61.27	61.44	61.12	61.17

Table 9. The Macro F1 of Support Vector Machines Used Five Feature Selections on Reuters-21578. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	200	400	600	800	1000	1200	1400	1600	1800	2000
GFS	**63.02**	**62.20**	**62.00**	**61.83**	**61.84**	**61.63**	**61.62**	61.45	61.61	61.57
IG	60.27	60.21	60.56	61.08	61.08	61.57	61.49	61.42	61.44	**61.73**
DF	59.56	60.05	60.61	61.62	61.71	61.56	61.44	**61.53**	**61.67**	61.67
OCFS	57.34	59.16	59.76	60.32	61.14	61.59	61.52	61.42	61.60	61.55
XP2	59.22	59.21	59.06	58.97	58.73	58.56	58.54	58.64	58.29	58.37

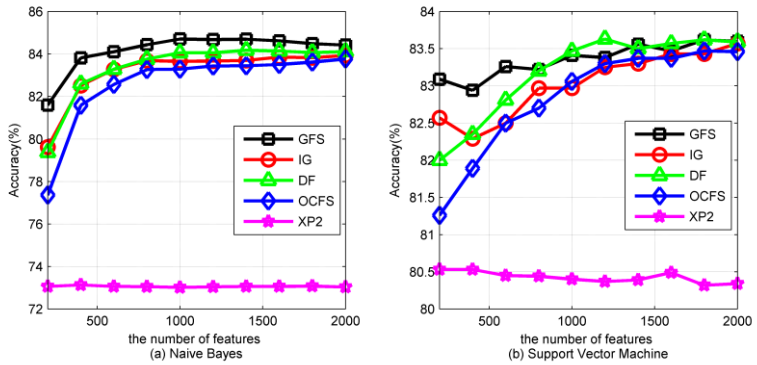

Figure 2. The Accuracy Curve of Naïve Bayes and Support Vector Machines Used Five Feature Selections on Reuters-21578, Respectively

4.3. Results on WebKB Corpus

Table 10 and Table 11 shows the comparison of micro-averaged F1 and macro-averaged F1 among the different selection algorithms when NB classifier was used on WebKB, respectively. It can be seen from Table 10 that the micro F1 measure of NB with GFS on WebKB is superior to that with other four feature selections except for the number of selected features is 400. Table 11 indicates that the macro F1 of NB combined with GFS outperforms that combined with other four feature selections when the number of the selected features is not equal to 400. Table 12 and Table 13 show the comparison of micro-averaged F1 and macro-averaged F1 among the different selection algorithms when SVM classifier was used on WebKB, respectively. When SVM is used on WebKB, the proposed method GFS is superior to other feature selection algorithms in terms of micro F1 and macro F1 except for the number of selected features is 200, 1200 and 1800. Figure 3. lists the accuracy curve of NB and SVM combined with five feature selections used on WebKB. Figure 3(a) shows that the curve of NB combined with GFS is significantly higher than that with IG, DF and XP2 and is very close to that with OCFS. Figure 3(b) indicates that the curve of SVM combined with GFS used on WebKB is higher than that with other feature selections except for the number of selected features is 200 and 1200 and reaches the peak (88.02%) when the number of selected features is 1400.

Table 10. The Micro F1 of Naïve Bayes Used Five Feature Selections on WebKB. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	**200**	**400**	**600**	**800**	**1000**	**1200**	**1400**	**1600**	**1800**	**2000**
GFS	**70.52**	72.73	**74.30**	**75.28**	**76.50**	**77.25**	**77.43**	**77.74**	**77.97**	**78.17**
IG	69.17	71.36	73.26	74.17	75.23	75.84	76.32	76.61	77.17	77.46
DF	67.37	71.04	72.91	73.74	74.82	75.82	76.29	76.87	76.98	77.23
OCFS	69.85	**73.07**	74.12	75.23	76.23	76.46	77.10	77.09	77.50	78.02
XP2	65.05	65.17	64.71	64.74	64.76	64.65	64.73	64.66	65.32	65.33

Table 11. The Macro F1 of Naïve Bayes Used Five Feature Selections on WebKB. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	**200**	**400**	**600**	**800**	**1000**	**1200**	**1400**	**1600**	**1800**	**2000**
GFS	**61.80**	64.26	**68.33**	**70.48**	**72.72**	**74.07**	**74.40**	**75.09**	**75.59**	**75.95**
IG	60.01	63.56	66.35	67.50	69.81	71.29	72.16	72.86	73.98	74.44
DF	59.03	62.68	64.86	66.56	69.00	70.92	72.34	73.15	73.42	74.00
OCFS	61.28	**64.87**	67.48	69.44	71.59	72.43	73.70	73.80	74.53	75.38
XP2	55.04	55.11	55.20	55.29	55.31	55.45	55.64	55.80	56.06	56.21

Table 12. The Micro F1 of Support Vector Machines Used Five Feature Selections on WebKB. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	**200**	**400**	**600**	**800**	**1000**	**1200**	**1400**	**1600**	**1800**	**2000**
GFS	83.86	**84.81**	**85.62**	**86.16**	**86.81**	87.10	**87.36**	**87.31**	87.16	**87.14**
IG	82.97	83.52	85.22	85.66	86.02	86.10	86.65	86.80	86.52	86.77
DF	83.26	83.84	84.64	85.87	86.21	**87.27**	87.16	87.04	**87.19**	87.05
OCFS	**84.17**	84.39	85.31	85.64	86.31	86.49	86.79	86.84	86.75	87.04
XP2	81.26	80.69	80.87	81.46	81.21	81.27	81.37	81.26	80.77	80.28

Table 13. The Macro F1 of Support Vector Machines Used Five Feature Selections on WebKB. The Value of GFS which Outperforms that of Other Four Methods is Indicated in Boldface

The number of features	**200**	**400**	**600**	**800**	**1000**	**1200**	**1400**	**1600**	**1800**	**2000**
GFS	83.62	**84.58**	**85.39**	**85.91**	**86.56**	86.89	**87.16**	**87.13**	86.97	**86.95**
IG	82.68	83.26	84.98	85.44	85.74	85.83	86.41	86.56	86.27	86.56
DF	83.05	83.58	84.39	85.62	85.99	**87.07**	86.95	86.86	**87.02**	86.88
OCFS	**83.95**	84.14	85.08	85.33	86.03	86.23	86.56	86.63	86.53	86.84
XP2	80.84	80.26	80.42	80.28	79.96	79.98	80.11	79.97	80.10	79.76

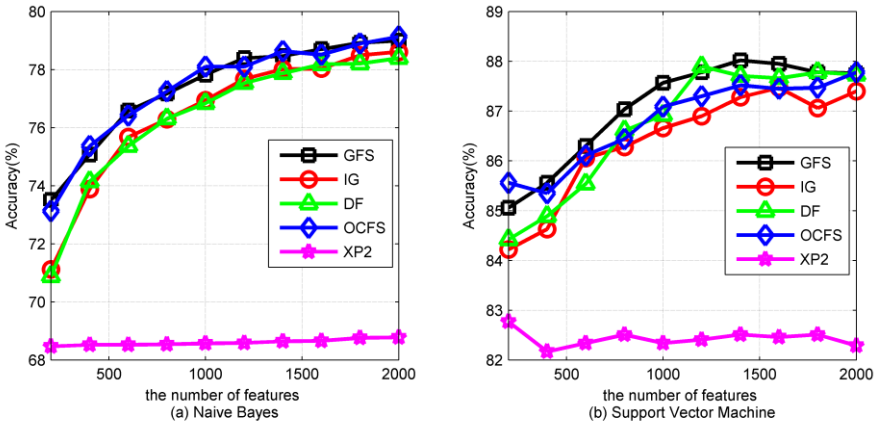

Figure 3. The Accuracy Curve of Naïve Bayes and Support Vector Machines Used Five Feature Selections on WebKB, Respectively

5. Analysis and Discussion

5.1. Statistical Analysis

In order to compare the performance of the proposed method with the previous approaches, Friedman and Iman & Davenport [48] tests, which are non-parametric tests, are used in the statistical analysis. If the null hypothesis of Friedman and Iman & Davenport tests is rejected, the post test (Holm test) [49, 50] can be used to detect significant differences between the control algorithm and other algorithms. It compares the *p*-value with the value of α/i under the null hypothesis that the control algorithm is equivalent to other algorithms. If the *p*-value is below α/i, the corresponding hypothesis is rejected, namely the control algorithm is significantly outperforms the corresponding algorithm. In this paper, we compare the accuracy of five feature selections using 30 data sets which consist of the 10-fold cross validation on three data sets. Table 14 and Table 15 show the Holm test table for α=0.05 when the Naïve Bayes and Support Vector Machines are used, respectively. It can be seen from Table 14 and Table 15 that all *p*-value are less than the corresponding α/i. So the GFS significantly outperforms the other four algorithms.

Table 14. Holm Test Table for α= 0.05 when Naïve Bayes is Used

i	algorithms	z=(R_0-R_i)/SE	*p*-value	α/*i*
4	XP2	6.7769	1.2276E-11	0.0125
3	IG	5.5114	3.5609E-8	0.0167
2	DF	4.5316	5.8551E-6	0.025
1	OCFS	3.5926	3.2741E-4	0.05

Table 15. Holm Test Table for α= 0.05 when Support Vector Machines is Used

i	algorithms	z=(R_0-R_i)/SE	*p*-value	α/*i*
4	IG	5.6338	1.7625E-8	0.0125
3	OCFS	3.2660	0.00109	0.0167
2	XP2	3.1843	0.00145	0.025
1	DF	3.0210	0.00252	0.05

5.2. Discussions

In our experiment, three text benchmark corpora are used to compare the performance of the feature selection algorithms. During these corpora, the 20-Newgroups is a balanced dataset, namely the documents in 20-Newgroups are evenly assigned to 20 categories. While the distribution of the documents in the Reuters-21578 and WebKB are imbalanced. If we do not consider the difference of the length of documents in corpus, the amount of a feature occurring in a category also has bias in the imbalanced dataset. Take "flex" in Reuters-21578 corpus as an example, the amount of term frequency of this feature occurring in category "earn", which contains 3864 documents, is 722, and the amount of term frequency of this feature occurring in category "crude", which contains 578 documents, is 216. Although the amount of feature "flex" occurring in category "earn" is greater than that in category "crude", we cannot determine that the feature "flex" can stand for category "earn". So we must get rid of the effect of the imbalance dataset on the feature selection. In the proposed method, we eliminate the effect of the imbalance dataset using the object distance between the data object and the atomic data object. The bigger the amount of term frequency of all features in one category is, the farther the distance between the data object and the atomic data object is and the less the strength of gravitation between data object and the atomic data object is.

The gravity had been used for feature selection by Guo *et al*. [30], but their method is entirely different from the GFS in three respects. Firstly, the formations of the gravitational field are different. Every feature in the corpus is regarded as an entity and all features are used to constitute the gravitational field by Guo *et al*.[30], while the GFS only regards a feature occurring in one category as an object, and the number of the objects in the gravitational field is equal to the number of the categories. Secondly, Guo's method calculated the gravitation between any two features in the gravitational field, and the GFS only considers the gravitation between an atomic data object with all the objects in the gravitational field. Finally, Guo' method used gravitation between features for clustering, and then selected some feature groups for classification; however, the proposed method calculates the significance of a feature for categorization using the gravitation.

6. Conclusion

We proposed a novel feature selection algorithm based on gravitation, named GFS. We think that a feature occurring in one category can be regard as an object, and the gravitational field can be formed by all the objects corresponding to a feature occurring in various categories. We assume that there is a feature that comes from a document with unknown category label and it is regarded as an atomic object. The atomic object will be attracted by all objects in the gravitational field. The strength of gravitation of an object represents the significance of the feature for corresponding category. If the strength of

gravitation of an object which stands for the feature occurring in one category is the biggest, the feature can well represent this category.

To evaluate the effect of GFS, we use two classifiers: Naïve Bayes (NB) and Support Vector Machines (SVM) on three benchmark text corpora: 20-Newgroups, Reuters-21578 and WebKB, and compare it with the four well-known feature selection algorithms: information gain (IG), document frequency (DF), orthogonal centroid feature selection (OCFS) and Poisson distribution. The experiments results indicate that GFS significantly outperforms this four feature selection when Naïve Bayes and Support Vector Machines are used.

Acknowledgment

This research is supported by the project development plan of science and technology of Jilin Province under Grant no. 20140204071GX.

References

[1] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features", Machine Learning: ECML-98, **(1998)**, pp. 137-142.
[2] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization", Information Processing & Management, vol. 42, no. 1, **(2006)**, pp. 155-165.
[3] M. Dash and H. Liu, "Feature Selection for Classification", Intelligent Data Analysis, vol. 1, **(1997)**, pp. 131-156.
[4] D. Fragoudis, D. Meretakis and S. Likothanassis, "Best terms: an efficient feature-selection algorithm for text categorization," Knowledge and Information Systems, vol. 8, no. 1, **(2005)**, pp. 16-33.
[5] K. Kira and L. A. Rendell, "The feature selection problem: traditional methods and a new algorithm", Proceedings of the tenth national conference on Artificial intelligence, San Jose, California, **(1992)**.
[6] P. M. Narendra and K. Fukunaga, "A Branch and Bound Algorithm for Feature Subset Selection," IEEE Trans. Comput., vol. 26, no. 9, **(1977)**, pp. 917-922.
[7] D. Koller and M. Sahami, "Toward Optimal Feature Selection", 13th International Conference on Machine Learning, **(1996)**, pp. 284 - 292.
[8] A. L. Blum, and P. Langley, "Selection of relevant features and examples in machine learning", Artificial Intelligence, vol. 97, no. 1-2, **(1997)**, pp. 245-271.
[9] D. Mladenic and M. Grobelnik, "Feature selection on hierarchy of web documents", Decision Support Systems, vol. 35, no. 1, **(2003)**, pp. 45-87.
[10] G. H. John, R. Kohavi and K. Pfleger, "Irrelevant Features and the Subset Selection Problem,"Proceedings of the Eleventh International Conference on Machine Learning, San Francisco, CA, **(1994)**, pp. 121-129.
[11] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization", Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, TN, **(1997)**, pp. 412-420.
[12] D. Mladenic and M. Grobelnik, "Feature selection for unbalanced class distribution and Naive Bayes", Proceedings of the Sixteenth International Conference on Machine Learning, **(1999)**, pp. 258 - 267.
[13] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification", Journal of Machine Learning Research, vol. 3, **(2003)**, pp. 1289-1305.
[14] J. Yan, N. Liu, B. Zhang, S. Yan, Z. Chen, Q. Cheng, W. Fan and W.-Y. Ma, "OCFS: optimal orthogonal centroid feature selection for text categorization", in Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil, **(2005)**, pp. 122-129.
[15] Z. Chen and K. Lu, "A preprocess algorithm of filtering irrelevant information based on the minimum class difference", Knowledge-Based System, vol. 19, **(2006)**, pp. 422 - 429.
[16] S. S. R. Mengle and N. Goharian, "Ambiguity Measure Feature-Selection Algorithm", Journal of the American Society for Information Science and Technology, vol. 60, no. 5, **(2009)**, pp. 1037-1050.
[17] W. Shang, H. Huang and H. Zhu, "A novel feature selection algorithm for text categorization", Expert Systems with Applications, vol. 33, **(2007)**, pp. 1-5.
[18] J. Chen, H. Huang, S. Tian and Y. Qu, "Feature selection for text classification with Naive Bayes," Expert Systems with Applications, vol. 36, **(2009)**, pp. 5432-5435.
[19] H. Ogura, H. Amano and M. Kondo, "Feature selection with a measure of deviations from Poisson in text categorization", Expert Systems with Applications, vol. 36, **(2009)**, pp. 6826-6832.
[20] J. Yang, Y. Liu, Z. Liu, X. Zhu and X. Zhang, "A new feature selection algorithm based on binomial hypothesis testing for spam filtering", Knowledge-Based Systems, vol. 24, no. 6, **(2011)**, pp. 904-914.

[21] J. R. Quinlan, "Induction of Decision Trees", Machine Learning, vol. 1, no. 1, **(1986)**, pp. 81-106.
[22] E. Youn and M. K. Jeong, "Class dependent feature scaling method using naive Bayes classifier for text datamining", Pattern Recognition Letters, vol. 30, no. 5, **(2009)**, pp. 477-485.
[23] F. Sebastiani, "Machine Learning in Automated Text Categorization", ACM Computing Surveys, vol. 34, no. 1, **(2002)**, pp. 1-47.
[24] W. E. Wright, "Gravitational clustering", Pattern Recognition, vol. 9, no. 3, **(1977)**, pp. 151-166.
[25] Y. Endo and H. Iwata, "Dynamic Clustering Based on Universal Gravitation Model", Modeling Decisions for Artificial Intelligence, **(2005)**, pp. 367-382.
[26] J. Gomez, D. Dasgupta and O. Nasraoui, "A New Gravitational Clustering Algorithm", Proceedings of the third SIAM International Conference on Data Mining, **(2003)**.
[27] L. Peng, B. Yang, Y. Chen and A. Abraham, "Data gravitation based classification", Information Sciences, vol. 179, no. 6, **(2009)**, pp. 809-819.
[28] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "GSA: a gravitational search algorithm", Information Sciences, vol. 179, no. 13, **(2009)**, pp. 2232-2248.
[29] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "BGSA: binary gravitational search algorithm", Natural Computing, vol. 9, no. 3, **(2010)**, pp. 727-745.
[30] W. Guo, J. Chen, Z. Yang, J. Yin, X. Yang and L. Huang, "Embedded Classification Learning for Feature Selection Based on K-Gravity Clustering", Computational Intelligence and Intelligent Systems, **(2009)**, pp. 452-460.
[31] T. G. Dietterich, "Approximate Sratistical Tests for Comparing Supervised Classification Learning Algorithms", Neural Computation, vol. 10, **(1998)**, pp. 1895-1923.
[32] R. Kohavi, "Wrappers for performance enhancement and oblivious decision graphs", **(1995)**.
[33] F. Song, S. Liu and J. Yang, "A comparative study on text representation schemes in text categorization", Pattern Analysis & Applications, vol. 8, no. 1, **(2005)**, pp. 199-209.
[34] M. Craven, D. DiPasquo, D. Freitag, A. K. McCallum, T. M. Mitchell, K. Nigam and S. Slattery, "Learning to Extract Symbolic Knowledge from the World Wide Web", Proceedings of AAAI'98, 15th Conference of the American Association for Artificial Intelligence, madison, US, **(1998)**, pp. 509-516.
[35] K. Nigam, A. Mccallum and T. Mitchell, "Learning to Classify Text from Labeled and Unlabeled Documents", Proceedings of AAAI'98, 15th Conference of the American Association for Artificial Intelligence, Madison, US, **(1998)**.
[36] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to Spam filtering", Expert Systems with Applications, vol. 36, **(2009)**, pp. 10206–10222.
[37] T. Joachims, "Learning to Classify Text Using Support Vector Machines: Methods", Theory and Algorithms: Kluwer Academic Publishers, **(2002)**.
[38] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, "Distributional word clusters vs. words for text categorization", J. Mach. Learn. Res., vol. 3, **(2003)**, pp. 1183-1208.
[39] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification,"AAAI-98 Workshop on Learning for Text Categorization, **(1998)**.
[40] K.-M. Schneider, "A Comparison of Event Models for Naive Bayes Anti-Spam E-Mail Filtering", ACM Transactions on Asian Language Information Processing (TALIP) vol. 3, no. 4, **(2004)**, pp. 243 - 269.
[41] H. Drucker, D. Wu and V. N. Vapnik, "Support Vector Machines for Spam Categorization", IEEE Transactions on Neural Networks, vol. 10, **(1999)**, pp. 1048-1054.
[42] C.-C. Chang and C.-J. Lin, "LIBSVM : a library for support vector machines", http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[43] C.-C. Chang and C.-J. Lin, "Training v-Support Vector Classifiers: Theory and Algorithms", Neural Computation, vol. 13, no. 9, **(2001)**, pp. 2119-2147.
[44] M. A. Davenport, R. G. Baraniuk and C. D. Scott, "Controlling False Alarms with Support Vector Machines", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), **(2006)**, pp. 589-592.
[45] Y. H. Li and A. K. Jain, "Classification of Text Documents", The Computer Journal, vol. 41, no. 8, **(1998)**, pp. 537-546.
[46] M. Benkhalifa, A. Mouradi and H. Bouyakhf, "Integrating External Knowledge to Supplement Training Data in Semi-Supervised Learning for Text Categorization", Information Retrieval, vol. 4, no. 2, **(2001)**, pp. 91-113.
[47] A. Markov, M. Last and A. Kandel, "The hybrid representation model for web document classification", International Journal of Intelligent Systems, vol. 23, no. 6, **(2008)**, pp. 654-679.
[48] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the Friedman statistic", Communications in Statistics, vol. 18, **(1980)**, pp. 571-579.
[49] S. García, A. Fernández, J. Luengo and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability", Soft Computing - A Fusion of Foundations, Methodologies and Applications, vol. 13, no. 10, **(2009)**, pp. 959-977.
[50] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," J. Mach. Learn. Res., vol. 7, **(2006)**, pp. 1-30.

Authors

Jie-Ming Yang, received his MSc and PhD degree in computer applied technology from Northeast DianLi University, China in 2008 and Computer Science and Technology from Jilin University, China in 2013, respectively. His research interests are in machine learning and data mining.

Zhi-Ying Liu, received her MSc degree in computer applied technology from Northeast DianLi University, China in 2005. His research interests are in information retrieval and personalized recommendation.

Zhao-Yang Qu, received his MSc and PhD degree in computer science from Dalian University of Technology, China in 1988 and North China Electric Power University, China in 2012, respectively. His research interests are in artificial intelligence, machine learning and data mining.