INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(5), 689-703, October 2017.

Gravitation Theory Based Model for Multi-Label Classification

L. Peng, Y. Liu

Liwen Peng, Yongguo Liu*
Knowledge and Data Engineering Laboratory of Chinese Medicine,
School of Information and Software Engineering,
University of Electronic Science and Technology of China,
Chengdu, 610054, PR, China
*Corresponding author: liuyg@uestc.edu.cn

Abstract: The past decade has witnessed the growing popularity in multi-label classification algorithms in the fields like text categorization, music information retrieval, and the classification of videos and medical proteins. In the meantime, the methods based on the principle of universal gravitation have been extensively used in the classification of machine learning owing to simplicity and high performance. In light of the above, this paper proposes a novel multi-label classification algorithm called the interaction and data gravitation-based model for multi-label classification (ITDGM). The algorithm replaces the interaction between two objects with the attraction between two particles. The author carries out a series of experiments on five multi-label datasets. The experimental results show that the ITDGM performs better than some well-known multi-label classification algorithms. The effect of the proposed model is assessed by the example-based F1-Measure and Label-based micro F1-measure.

Keywords: data gravitation theory, interaction, multi-label classification.

1 Introduction

As a major area of research into machine learning systems, the traditional supervised classification learning technology attributes each instance to one class (label). With x as the set of all instances and l as the label space, the traditional supervised classification learning system aims to get the function $f : x \rightarrow l$ from the training dataset $\{(x_i, l_j) : 1 \leq i \leq m, 1 \leq j \leq k\}$, where m and k denote the number of instances and the number of labels, respectively. In other words, the purpose of the system lies in predicting the one correct label for each test instance. There are many algorithms for single-label classification, such as the decision tree C4.5 [4], the support vector machine (SVM) [1], the rough set [11], the neural network (NN) [8] and so on.

The traditional single-label classification may apply to some fields, but many instances do not belong to a specific class in reality, namely text categorization, music information retrieval and video classification. The limitation of the traditional single-label classification has contributed to the growing popularity in multi-label classification algorithms.

In general, multi-label classification methods are roughly divided into two categories [2]: the problem transformation method and the algorithm adaptation method. The former transforms multi-label dataset into other well-established learning scenarios, and processes the transformed dataset with the single-label classification algorithm. Typical examples are the binary relevance algorithm (BR), the label combination method (LC), high-order classifier chains [13], and random k label sets (RAkEL) [17]. By contrast, the algorithm adaptation method directly applies adaptive learning methods to solve the problem of multi-label learning. The representative algorithm adaptation methods include the k-nearest neighbor algorithm for multi-label learning (ML-kNN) [24], the back-propagation multi-label learning (BPMLL) [23], the bichromatic reverse k-nearest neighbor algorithm (BRkNN) [15] and the dependent multi-label k-nearest neighbor algorithm (DML-kNN) [21]. As an extended version of the KNN method, the ML-kNN approach employs the maximum posteriori probability to determine the label set of test instances. Based on the label sets of the neighbor instances, the BRkNN method calculates the confidence of each label, and decides the k nearest neighbors of each test instance. The BPMLL is another classic back-propagation multi-label classification algorithm which proposes a new error function for multiple labels.

As far as we know, the methods based on the principle of universal gravitation has been extensively used in the classification or clustering of machine learning owing to simplicity and high performance. The existing physical gravitation-based classification or clustering methods mostly focus on two important factors: the data distance and density. For example, Reyes et al. [14] proposed a multi-label lazy algorithm based on the data gravitation model, named the MLDGC. It is the first algorithm developed for data gravitation model on multi-label classification. However, the data gravitation-based methods often ignore the interaction of data mass, and directly define data gravitation with density, resulting in inconsistency with human perception. In view of the necessity to include the attraction between two particles in the data gravitation-based classification, this paper proposes a novel algorithm based on gravitation theory called the interaction and data gravitation-based model for multi-label classification (ITDGM).

The rest of this paper is organized as follows: Section 2 briefly introduces gravitation and reviews the related research on data gravitation, Section 3 presents the proposed algorithm, Section 4 describes the experiment, Section 5 shows and discusses the experimental results, and Section 6 wraps up the research with some conclusions.

2 Literature review

Whenever there is an interaction between two objects, there is a force upon each of the objects. One of the forces can be characterized by Newton's law of universal gravitation. The universal gravitation-based methods are widely used in classification or clustering.

Wright [18] was the first to apply such a method in cluster analysis. Gomez et al. [6] proposed a clustering algorithm based on feature space that considers each of the instances. On account of gravitational collapse, Wang et al. [19] put forward an improved algorithm based on k-NN classifier. In light of the theoretical model of data gravitation and data gravitation field, Yang et al. [22] presented a new classification model called data gravitation-based classifier (DGC), and used it in an intrusion detection system (IDS). Li et al. [9] developed a nonlinear classification algorithm to handle data gravity as data vector. Shafigh et al. [16] proposed a method based on the gravitational potential energy of particles named gravitation-based classification (GBC) algorithm, sought for the equilibrium condition of the classifier, and proved that the algorithm had strong robustness but high computational complexity. Wena et al. [20] came up with the cognitive gravitation model (CGM) algorithm, which uses the self-information of each instance instead of the mass based on the law of gravitation and the cognitive laws. Starting from the DGC method, Peng et al. [12] created the improved DGC (IDGC) in an attempt to handle imbalance datasets with amplified gravitation coefficient (AGC), which could strengthen and weaken the gravitational field of the minority and majority instances. Reyes et al. proposed an algorithm based on lazy learning and data gravitation model for classification which simultaneously measured classification accuracy and label ranking.

Among the aforementioned data gravitation-based algorithms, most of them are applied in single-label learning systems. The only exception is the MLDGC, which is adopted for multi-label classification. Overall, few scholars have integrated the data gravitation model and multi-label classification.

3 The proposed algorithm based on gravitation theory

So far, the laws of physics have been extensively applied to the information domain. It is stressed that the interaction of data instances must conform to the physical laws. The laws are implemented to simulate actual data classification in many novel data gravitation-based classification algorithms. However, the simulation results may not fully correspond with the reality as the data gravitation is directly defined with the density instead of the interaction between data instances.

To solve the problem, this paper proposes a multi-label learning algorithm based on data gravitation model (ITDGM). In a multi-label dataset, an instance i can be transformed into a data particle expressed as (X_i, Y_i, m_i). The feature vector of the data particle i is defined as the label set of the particle and denoted by X_i and Y_i.

The symbol m_i means the data mass of particle i. For any two particles i and j, the distance between them can be calculated as

$d_F(i,j) = \sqrt{\sum_{\forall f \in F} \delta(x_{if}, x_{jf})^2}$.

Since the Heterogeneous Euclidean-Overlap Metric (HEOM) can handle both nominal and continuous features, the above heterogeneous distance function is adopted in the experiment.

Therefore, the term $\delta(x_{if}, x_{jf})^2$ should be solved under three different conditions:

- $\delta(x_{if}, x_{jf})^2 = 1$ when $x_{if} \neq x_{jf}$ and the data attribute is discrete;
- $\delta(x_{if}, x_{jf})^2 = 0$ when $x_{if} = x_{jf}$ and the data attribute is discrete;
- $\delta(x_{if}, x_{jf})^2 = \frac{|x_{if} - x_{jf}|}{\max(f) - \min(f)}$ when the data attribute is continuous;

Where x_{if} and x_{jf} are the f-th eigenvalues of i and j, respectively; F is the feature space; $\delta(x_{if}, x_{jf})$ is the type of f-th feature; $\max(f)$ and $\min(f)$ are the maximum and minimum values, respectively. Hence, it is possible to identify the k-nearest neighbors of particle i by the $d_F(i,j)$ function. The k-nearest neighbors of particle i can calculate by $N_i = (i_1, i_2, \ldots, i_k) \, | d_F(i, i_1), d_F(i, i_2), ..., d_F(i, i_k)$, which is sorted by ascending order.

According to the Newton's law of universal gravitation, the greater the mass of two objects, the greater the resulting interaction between them. In other words, the gravitation increases with the interaction between the two objects. In this research, a new concept is defined: the interaction-based gravitation coefficient (IGC). The coefficient is created to evaluate the attraction between two particles like the interaction between two objects. Data has not mass. For the reason that data has no mass, the data mass is usually set as one. However, the setting is inconsistent with human perception without considering the interactions between data particles. Thus, the author defines the IGC and a data gravitation-based model to measure the strong or weak gravitational force of particles. The dissimilarity between the label set of the particle i and that of particle j is expressed by the operator $\Theta(Y_i, Y_j)$ and the interaction between the two particles $M(x_{i,j})$ is defined as:

$$P(i,j) = \frac{\Theta(\mathrm{Y_i}, \mathrm{Y_j})}{q} \times \frac{1}{f_j} \tag{1}$$

$$M(x_{i,j}) = \log\left(\frac{1}{P(i,j)}\right) = \log\left(\frac{q f_j}{\Theta(Y_i, Y_j)}\right) \tag{2}$$

Where $P(i,j)$ is the dissimilarity degree between the label set of the particle i and that of particle j, and f_j is the number of the instance labels, and q is the number of the total labels.

For the sake of data stability, $M\left(x_{i,j}\right)$ is introduced to measure the dissimilarity in the classification between particles. From the above formulas, it is concluded that $P\left(i,j\right)$ is inversely proportional to $M\left(x_{i,j}\right)$, that is, the smaller the $P\left(i,j\right)$, the more intense the interaction between the particle i and j.

Based on neighborhood density concept of Reyes et al. [14], the IGC value of particle i is calculated by the formula $m_i = d_i^{w_i}$, where d_i is the neighborhood density or the distribution of the neighbor particles of i, and w_i is the neighborhood weight of particle i. The greater the value of d_i, the more similarity between the label set of another particle and that of particle i. The neighborhood density is computed by the following formula:

$$d_i = 1 + \sum_{j \in N_i} (1 + M(x_{i,j}))/d_F(i,j) \tag{3}$$

Based on the above formula, the neighborhood weight calculation method is improved as below. First, $p_{dY}^i = \frac{\sum_{j \in N_i} I(x_{i,j})}{k}$ and $p_{dF}^i = \frac{\sum_{j \in N_i} d_F(i,j)}{k}$ are identified as the nearest particles of i based on the principle of prior probability and the prior probability of the nearest particles of i in the feature space, respectively. The symbol k is the initial number of nearest neighbors, and N_i is the nearest neighbors set of the particle i. Then, the prior probability of the nearest neighborhood of i with similar label sets can be calculated by $p_{dY|dF}^i = \frac{\sum_{j \in N_i} I(x_{i,j}) \cdot d_F(i,j)}{k}$. Finally, the neighborhood weight (w_i) is obtained by the formula $w_i = \frac{p_{dY|dF}^i p_{dF}^i}{p_{dY}^i} - \frac{(1-p_{dY|dF}^i)p_{dF}^i}{1-p_{dY}^i}$.

Through the above steps, it is possible to get the gravitational force between two instances. Taking an instance from dataset, the k-nearest neighborhood of particle i can be computed as below.

$$\mathrm{F}\left(\mathrm{i,j}\right) = \mathrm{g}\frac{IGC}{d_F(i,j)^2} = \mathrm{g}\frac{m_i}{d_F(i,j)^2} \tag{4}$$

In this experiment, the gravitational constant g is set to 1 for the calculation of the gravitation of instances. Thus, the gravitational force $f\left(i,j\right) = \mathrm{g}\frac{m_i m_j}{r^2}$ is rewritten as $F\left(i,j\right) = \frac{IGC}{d_F(i,j)^2}$, where the IGC replaces the product of the mass of the objects.

After obtaining the gravitational force between instances, it is necessary to determine which labels belong to the test instances. To this end, a simple statistical approach is implemented. The number of the y-th label that belongs to the k-nearest particles is expressed as $n\left(i\right) = \sum\limits_{j \in F_i} F\left(i,j\right), y \in Y_j$, and the number of the y-th label that does not belong to the k-nearest particles is expressed as $n\prime\left(i\right) = \sum\limits_{j \in F_i} F\left(i,j\right), y \notin Y_j$, F_i is the set of the nearest neighbors. If $n\left(i\right) > n\prime\left(i\right)$, the y-th label belongs to the instance; otherwise, the y-th label does not belong to the instance.

The proposed method is divided into three main steps. First, calculate the interaction between instances with the IGC coefficient; second, compute the gravitation of each particle i; third, obtain the k-nearest neighbors of the instance based on gravitation. Figure 1 illustrates the basic procedure of the ITDGM method.

The proposed algorithm seeks for k-nearest neighbors of each multi-label instance, and aims to transform the instances to particles in the learning phase. If linear search strategy is adopted for the k-nearest neighbors of instance i, the optimal time complexity of the learning phase is $O\left(n^2 \cdot d\right)$ for all training instances, i.e. it takes $O\left(n \cdot k \cdot d\right)$ steps at most to classify k instances, where n is the number of instances and d is the cardinality of the feature space.

Algorithm 1 The ITDGM algorithm

Input: $T_r \rightarrow$ multi-label training dataset; $T_s \rightarrow$ multi-label testing dataset; $k \rightarrow$ number of nearest neighbours

Output: label set of instances

Begin

```
Learning phase
for each i $\in T_r$ do
    $N_i \leftarrow k$ Nearest neighbors (i,$T_r, k$)
    $m_i \leftarrow IGC(i, N_i)$
end
Test phase
for each i $\in T_s$ do
    $F_i \leftarrow k$ Nearest neighbors (i,$T_s, k$)
    $Y_i \leftarrow \emptyset$
        For each $y \in Y$ do
            $\text{n}\,(\text{i}) = \sum_{j \in F_i} F\,(i, j)\,, y \in Y_j$
            $\text{n}\prime\,(\text{i}) = \sum_{j \in F_i} F\,(i, j)\,, y \notin Y_j$
            If $\text{n}\,(\text{i}) > \text{n}\prime\,(\text{i})$ then
                $Y_i \leftarrow Y_i \cup y$;
end
```

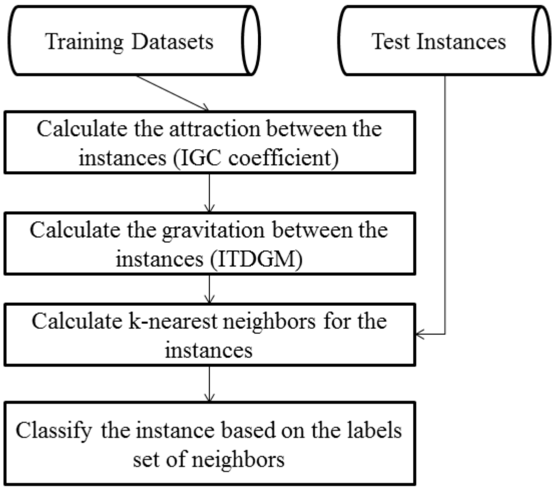
Training Datasets
Test Instances
Calculate the attraction between the instances (IGC coefficient)
Calculate the gravitation between the instances (ITDGM)
Calculate k-nearest neighbors for the instances
Classify the instance based on the labels set of neighbors

Figure 1: Basic procedure of the ITDGM method

4 Experiment

Due to the one-to-many relationship between instances and labels, the evaluation of multi-label classification performance is much more complicated than the traditional supervised single-label learning. The existing evaluation measures for multi-label classification mainly fall into two groups: the example-based measure [7] and the label-based measure [25]. In this research, the experimental results are verified by a string of different public datasets and discussed in details below.

4.1 Datasets

In this paper, five public multi-label datasets of different scales, labels and properties are selected from various domains to validate the proposed algorithm. The "Genbase" datasets [3] store information on the gene functions; the "Birds" contain bird species examples for acoustic classification; Corel16k001, Corel16k002 and Corel16k006 encompass examples of Corel images. The statistics on benchmark datasets are shown in Table 1, where n is the number of instances in the datasets; d is the number of features; q is the number of labels; d_s and l_c are the label cardinality and the label density, respectively.

Table 1: Statistics of the public multi-label benchmark datasets

Dataset	**Domain**	**n**	**d**	**q**	$\boldsymbol{d_s}$	$\boldsymbol{l_c}$
Genbase	biology	662	1186	27	32	1.252
Birds	audio	645	260	19	133	1.014
Corel16k001	image	13766	500	653	4937	2.867
Corel16k002	image	13761	500	654	4937	2.867
Corel16k006	image	13859	500	652	4937	2.867

4.2 Evaluation metrics

The performance of multi-label classification is measured differently from the single-label classification learning system. Let X $\in R_d$ be instances of data sets and express each instance as a d-dimensional vector x $= \{x_1, x_2, \ldots, x_d\}$,x $\in$ X. Let Y $= \{l_1, l_2, \ldots, l_m\}$ be a finite label set. Every instance belongs to a subset of Y and the subset is represented by an m-dimensional vector y $= \{y_1, y_2, \ldots, y_m\}$; if $y_k = 1$, label l_k belongs to instance x; if $y_k = 0$, the label does not belong to the instance. For a given training dataset $\mathrm{T} = \{(x_i, y_i) | 1 \leq \mathrm{i} \leq \mathrm{n}, x_i \in \mathrm{X}, y_i \in \mathrm{Y}\}$ composed of n instances, where x is the instance and y is the label subset of the instance, there is a corresponding test dataset $\mathrm{D} = \{(x_i, y_i) | 1 \leq \mathrm{i} \leq \mathrm{p}, x_i \in \mathrm{X}, y_i \in \mathrm{Y}\}$ composed of p instances based on the evaluation of the classification performance on test instances and the test set that returns the mean value.

In this paper, the instance-based F1-measure ($F1_{Ex}$) is implemented to present the experimental evaluation results:

$$F1_{Ex} = \frac{1}{m} \sum_{i=1}^{m} \frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \tag{5}$$

Where m is the number of test instances, and Z_i is the predicted label set of a given test instance i. The greater the value of $F1_{Ex}$, the better the performance of the multi-label classification algorithm.

Different from the above example-based measure, the label based micro F1-measure relies on the harmonic mean between recall and precision. The precision refers to the relevance of the predicted labels and the recall reflects the percentage of the predicted labels that are relevant. Such a measure is used in this research to present the algorithm performance. The F1-measure and the micro averaging are expressed by the following formulas:

$$\mathrm{F1-measure} = \frac{1}{N} \sum_{i=1}^{N} \frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \tag{6}$$

$$M_{micro} = M\left(\sum_{l=1}^{M} tp_l, \sum_{l=1}^{M} fp_l, \sum_{l=1}^{M} tn_l, \sum_{l=1}^{M} fn_l\right) \tag{7}$$

Where tp_l is the true positives; fp_l is the false positives; tn_l and fn_l are the true and false negatives for l labels after a binary evaluation, separately. The greater the micro F1-measure, the better the classification performance of the multi-label algorithm.

4.3 Experiment setting

The performance of the proposed algorithm ITDGM is compared with that of five classic multi-label classification algorithms: the ML-kNN, the BR, the RAkEL, the BRkNN and the BPMLL. In the RAkEL and BR algorithms, the C4.5 decision tree is taken as the classifier. In this research, the number of the nearest neighbors is 2∼7 for the ML-kNN, the ITDGM and the BRkNN; the smoothing factor of the ML-kNN is 1.0; the learning rate parameter of the BPMLL is 0.05; the Epoch parameter is 100; 20% of the input units are hidden units. The experiments are conducted on an Interl Core-i5 2.3 GHz processor with 8G memory, with the aid of MATLAB 2012 and the Java-based open resource software MULAN, which contains lots of classic and popular multi-label algorithms, evaluation targets and measures. The results are subjected to 5-fold cross validation.

5 Results and discussion

In this research, the example-based F1-measure ($F1_{Ex}$) and micro F1-measure are taken as the multi-label classifier evaluation criteria. In any case, the best results are highlighted and bolded. As mentioned above, the greater the $F1_{Ex}$ and micro F1-measure, the better the performance of the corresponding evaluation method.

Tables 2∼3 show the results of multi-label classification algorithms on test datasets for $F1_{Ex}$ and micro F1-measure. It is seen that the proposed algorithm ITDGM has better performance than the other methods.

Based on the $F1_{Ex}$ values in Table 2, the ITDGM outperforms the other five classic methods on the five public datasets. The $F1_{Ex}$ index of the proposed algorithm surpasses that of the other algorithms more than 3% on the“Bird” dataset, and 2% on the remaining datasets.

Table 3 displays the micro F1-measure values of the proposed algorithm and the other algorithms. The results show that the proposed algorithm boasts the best performance on the Corel16k001, Corel16k002 and Corel16k006 datasets, and keeps a 2% lead over the other algorithms in "Genbase" and "Birds" datasets.

Table 2: Example-based F1-measure ($F1_{Ex}$) values of all algorithms on the five public datasets

Algorithm	**Dataset**				
	Genbase	**Birds**	**Corel16k001**	**Corel16k002**	**Corel16k006**
MLKNN	0.9649±0.0099	0.5330±0.0436	0.0071±0.0027	0.0063±0.0042	0.0109±0.0039
RAkEL	0.9705±0.0169	0.5608±0.0334	0.0459±0.0093	0.0739±0.0059	0.0798±0.0070
BRKNN	0.9565±0.0090	0.5649±0.0385	0.0410±0.0105	0.0522±0.0096	0.0567±0.0071
BPMLL	0.0772±0.0457	0.4436±0.0594	0.0289±0.0011	0.0287±0.0026	0.0282±0.0017
BR	0.9705±0.0169	0.5608±0.0334	0.0535±0.0054	0.0849±0.0088	0.0798±0.0070
ITDGM	**0.9766±0.0121**	**0.5661±0.0544**	**0.0851±0.0076**	**0.0946±0.0125**	**0.0995±0.0066**

Figures 2∼11 show the relationship between the values of $F1_{Ex}$ and micro F1-measure and the number of nearest neighbors in all algorithms. It is demonstrated that the proposed algorithm

Table 3: Micro F1-measure values of all algorithms on the five public datasets

Algorithm	**Dataset**				
	Genbase	**Birds**	**Corel16k001**	**Corel16k002**	**Corel16k006**
MLKNN	0.9447±0.0142	0.2728±0.0300	0.0129±0.0050	0.0094±0.0056	0.0134±0.0039
RAkEL	0.9625±0.0094	0.3456±0.0445	0.0606±0.0062	0.1000±0.0054	0.1077±0.0109
BRKNN	0.9446±0.0123	0.3972±0.0303	0.0629±0.0144	0.0717±0.0087	0.0787±0.0044
BPMLL	0.0870±0.0472	0.1528±0.0518	0.0414±0.0003	0.0408±0.0008	0.0396±0.0008
BR	0.9625±0.0094	0.3456±0.0445	0.0718±0.0066	0.1108±0.0081	0.1077±0.0109
ITDGM	**0.9686±0.0118**	**0.4314±0.0571**	**0.1066±0.0082**	**0.1138±0.0101**	**0.1195±0.0065**

has the best classification effect with the fewest k-nearest neighbors in the testing datasets. In Figures 2 and 3, all algorithms perform well except the BPMLL ($F1_{Ex}$: 0.0772, micro F1: 0.0870). The ITDGM achieves the best overall result (micro F1: 0.9686) in Figure 3, followed closely by the simply transformation methods the RAkEL and the BR (micro F1: 0.9625).

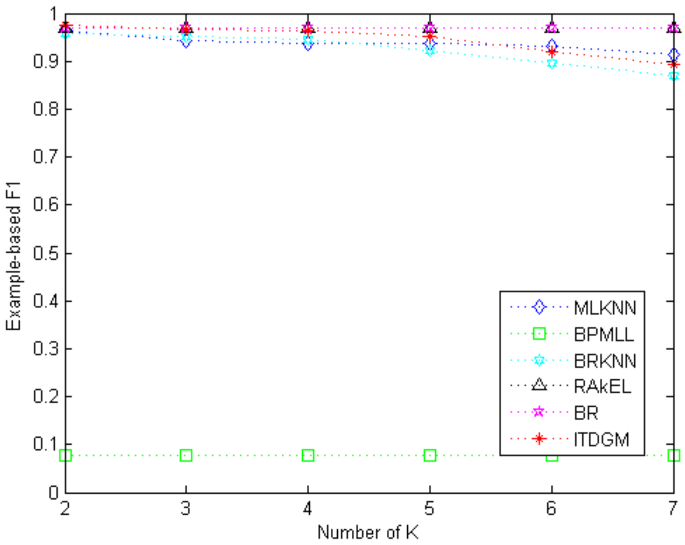

Figure 2: Relationship between $F1_{Ex}$ and the number of nearest neighbors in "Genbase" of all algorithms

The ITDGM is still the best performing algorithm in Figures 4 and 5. Although the ML-kNN and the BRkNN achieve fairly good performance at k=2 ($F1_{Ex}$: 0.5330) and k=3 ($F1_{Ex}$: 0.5649), respectively, the ITDGM has a slight edge over the two algorithms at k=2. As shown in Figure 5, the proposed algorithm has the best performance at almost any number of nearest neighbors on micro F1 metrics. According to Figure 4 and Figure 5, the RAkEL algorithm also has an outstanding performance. This is attributable to the random selection of label sets and unique strategies for training models in the algorithm.

Figures 6∼11 describe the metrics of all six algorithms on three Corel16k datasets. All selective multi-label classification methods perform poorly in these datasets, owing to the low label density and huge amount of distinctive label sets in the three datasets. In this case, the proposed algorithm also outperforms the other algorithms.

On the contrary, the ML-kNN shows the worst performance in the three datasets due to the capacity of its classifier. Despite the good performance in some cases, such as Corel16k001 ($F1_{Ex}$:

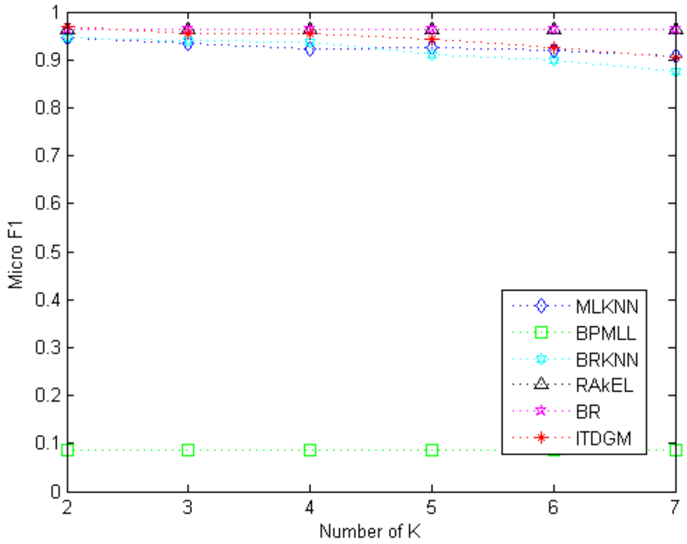

Figure 3: Relationship between micro F1 and the number of nearest neighbors in“Genbase” of all algorithms

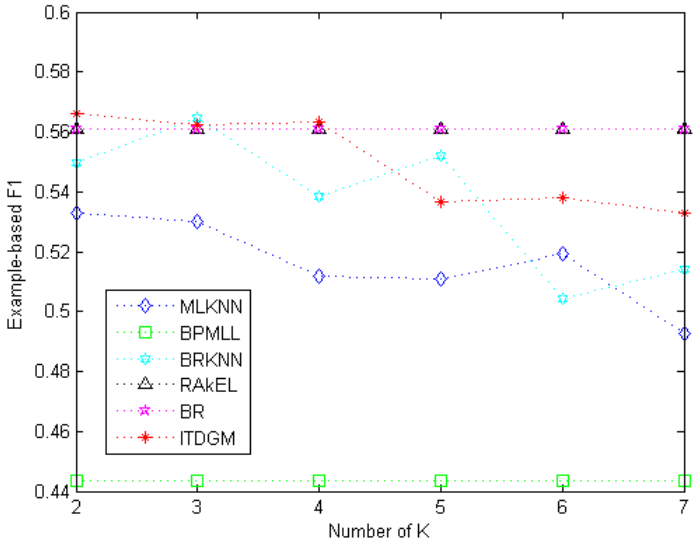

Figure 4: Relationship between $F1_{Ex}$ and the number of nearest neighbors in “Birds” of all algorithms

0.0535, micro F1: 0.0718), Corel16k002 ($F1_{Ex}$: 0.0849, micro F1: 0.1108), Corel16k006 ($F1_{Ex}$: 0.0798, micro F1: 0.1077), the simple transformation method BR costs much more training time than other algorithms. In comparison, the ITDGM, the BRkNN and the ML-kNN consumes less training time thanks to their roots in the lazy learning system.

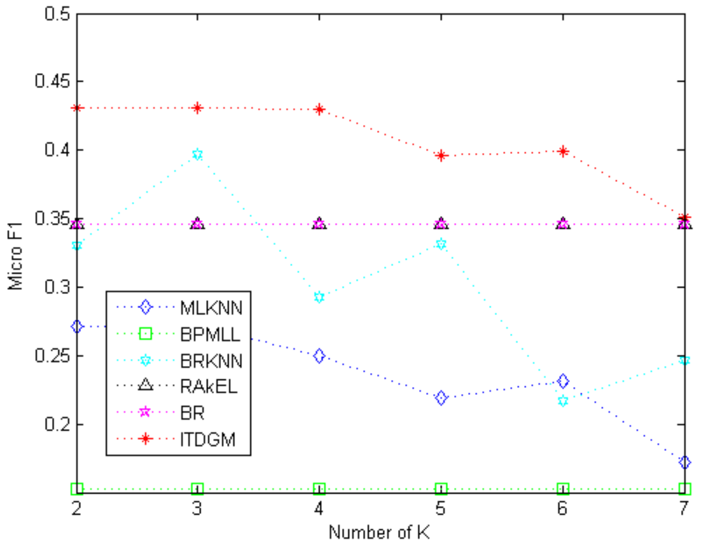

Figure 5: Relationship between micro F1 and the number of nearest neighbors in "Birds" of all algorithms

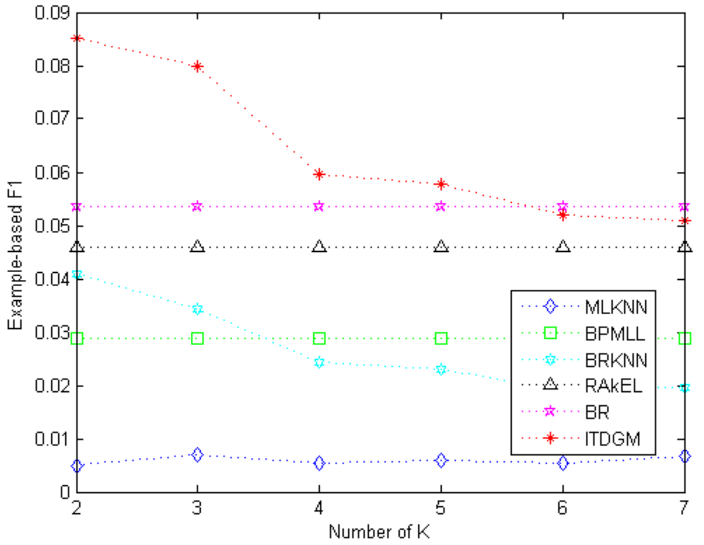

Figure 6: Relationship between $F1_{Ex}$ and the number of nearest neighbors in Corel16k001 of all algorithms

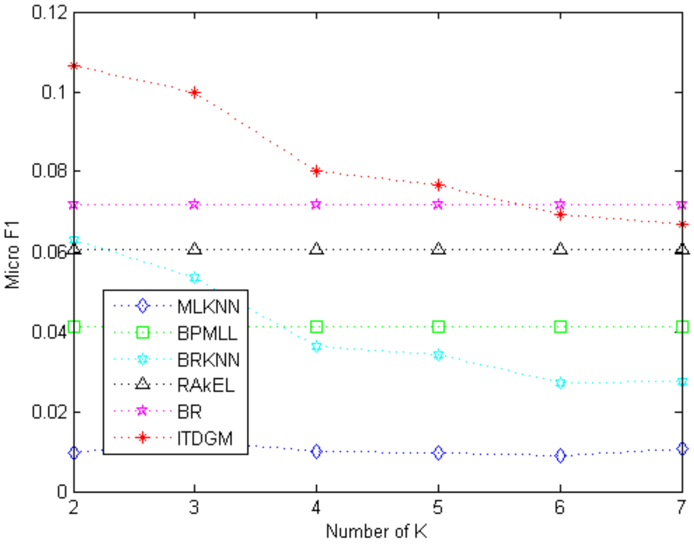

Figure 7: Relationship between micro F1 and the number of nearest neighbors in Corel16k001 of all algorithms

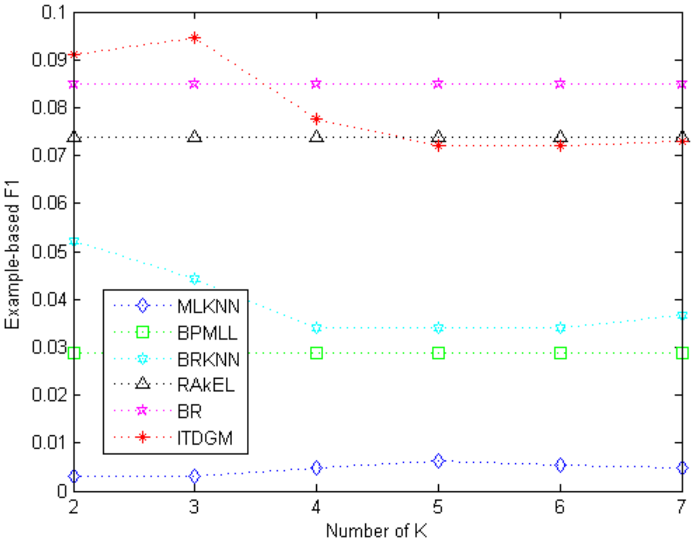

Figure 8: Relationship between $F1_{Ex}$ and the number of nearest neighbors in Corel16k002 of all algorithms

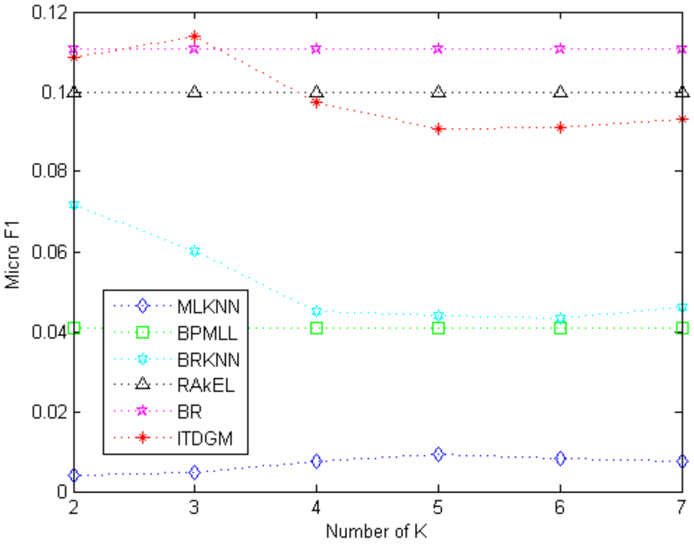

Figure 9: Relationship between micro F1 and the number of nearest neighbors in Corel16k002 of all algorithms

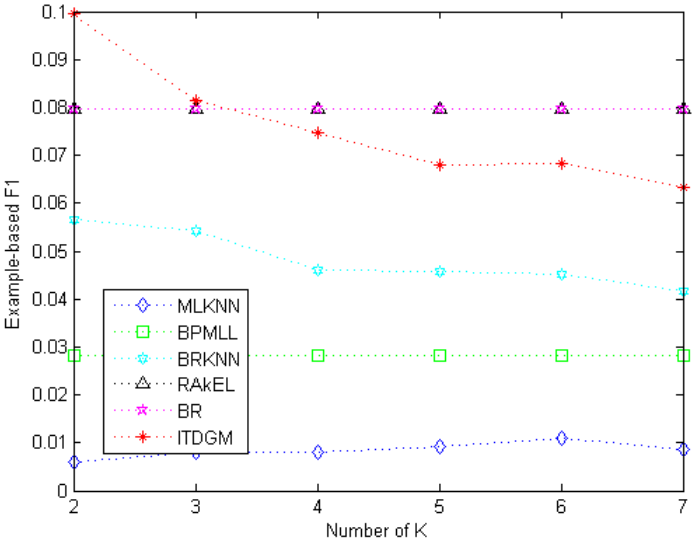

Figure 10: Relationship between $F1_{Ex}$ and the number of nearest neighbors in Corel16k006 of all algorithms

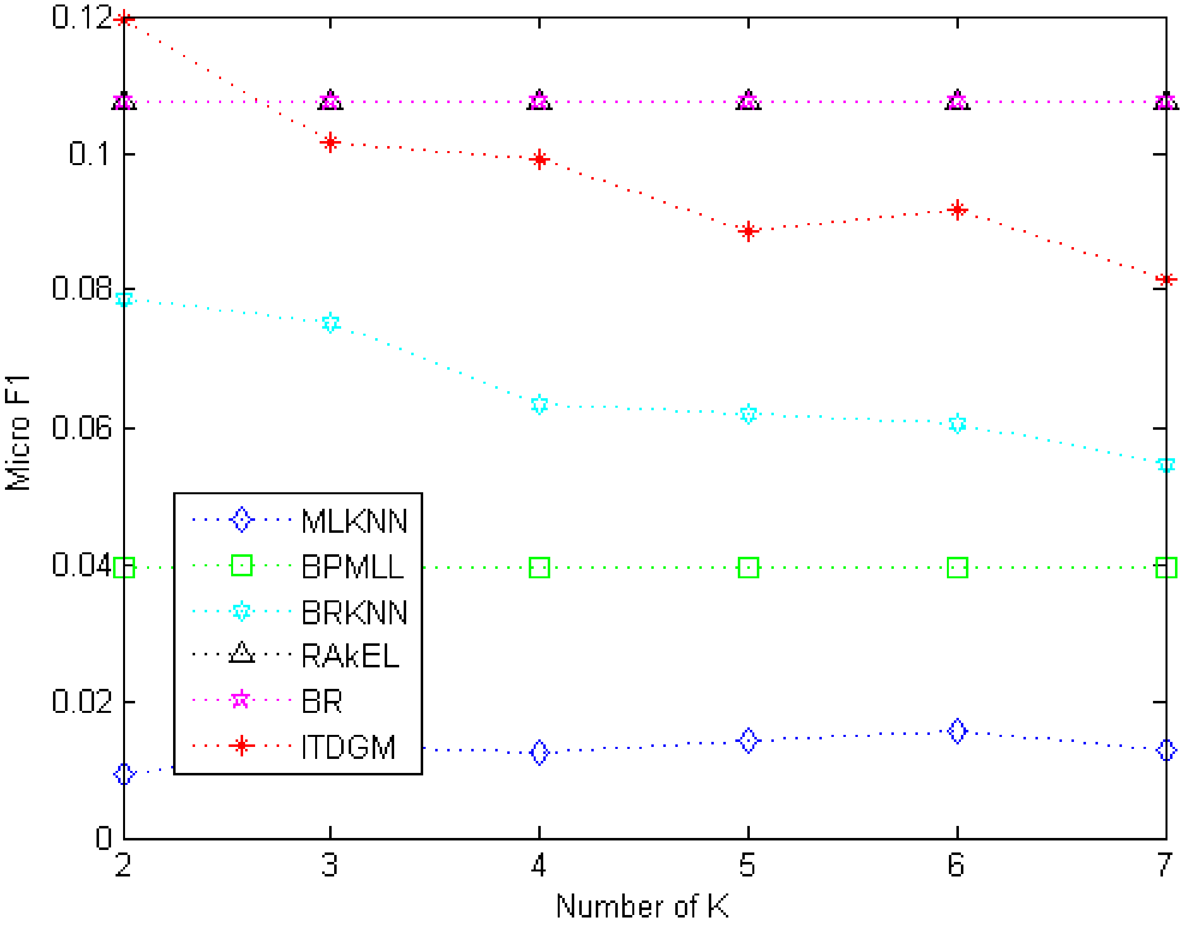

Figure 11: Relationship between micro F1 and the number of nearest neighbors in Corel16k006 of all algorithms

6 Conclusion

This paper presents a new multi-label classification algorithm based on data gravitation theory, introduces the interaction-based gravitation coefficient (IGC) to the proposed algorithm, and uses the algorithm to calculate the gravitational force instead of the mass of the particles. The average classification accuracy of the proposed method is evaluated by experiments on five public open datasets, in comparison with five classic multi-label algorithms: the BR, the RAkEL, the ML-kNN, the BPMLL and the BRkNN. The six algorithms are applied, under the example-based F1-measure ($F1_{Ex}$) and micro F1-measure, to the five public multi-label datasets at different number of nearest neighbors. In the experiment, the proposed algorithm outperforms all the other five algorithms, showcasing ideal capability and rationality. This research creates several new topics for future studies: the proposal of new multi-label classification method based on data gravitation model, integration with other fields [5] [10], and the improvement of the classification effect.

Acknowledgement

This research was supported in part by the National Natural Science Foundation of China (NSFC) under grants 60903074, the National High Technology Research and Development Program of China (863 Program) under grant 2008AA01Z119.

Bibliography

[1] Boser B.E., Guyon I.M., Vapnik V.N. (1992); A training algorithm for optimal margin classifiers, *Proce. of the 5th Annual ACM Conf. on Computational Learning Theory*, 144–152, 1992.

[2] Boutell M.R., Luo J., Shen X., Brown C.M. (2004); Learning multi-label scene classification, *Pattern Recognition*, 37(9), 1757–1771, 2004.

[3] Diplarisa S., Tsoumakas G., Mitkas P., Vlahavas I. (2005); Protein classification with multiple algorithms, *Proc. of the 10th Panhellenic Conference on Informatics*, 3746, 448–456, 2005.

[4] Freund Y. (1995); Boosting a weak learning algorithm by majority, *Information and Computation*, 121(2), 256–285, 1995.

[5] Fichera A., Fortuna L., Frasca M., Volpe R. (2015); Integration of complex networks for urban energy mapping, *International Journal of Heat and Technology*, 33(4), 181–184, 2015.

[6] Gomez J., Dasgupta D., Nasraoui O. (2013); A new gravitational clustering algorithm, *Proc. of the SIAM Intl. Conf. on Data Mining*, 2013.

[7] Ghamrawi N., McCallum A. (2005); Collective multi-label classification, *Proc. of the 14th ACM Intl. Conf. Information and Knowledge Management*, 195–200, 2005.

[8] Lu H., Rudy S., Liu H. (1996); Effect data mining using neural networks, *IEEE Transaction on Knowledge and Data Engineering*, 8(6), 957–961, 1996.

[9] Li J., Fu H. (2009); Data classification based on supporting data gravity. *Proc. of the IEEE Intl. Conf. on Intelligent Computing and Intelligent Systems*, 1, 22–28, 2009.

[10] Lin T., Wu P., Gao F.G., Yu Y., Wang L.H. (2015); Study on SVM temperature compensation of liquid ammonia volumetric flowmeter based on variable weight PSO, *International Journal of Heat and Technology*, 33(2), 151–156, 2015.

[11] Pawlak A. (1991); *Rough Sets: Theoretical Aspects of Reasoning about Data*, Dordrecht, 1991.

[12] Peng L., Zhang H., Yang B., Chen Y. (2014); A new approach for imbalanced data classification based on data gravitation, *Information Sciences*, 288(C), 347–373, 2014.

[13] Read J., Pfahringer B., Holmes G., Frank E. (2009); Classifier chains for multi-label classification, *Machine Learning and Knowledge Discovery in Databases*, 254–269, 2009.

[14] Reyes O., Morell C., Ventura S. (2016); Effective lazy learning algorithm based on a data gravitation model for multi-label learning, *Information Sciences*, 2016, s340–341, 159–174, 2016.

[15] Spyromitros E., Tsoumakas G., Vlahavas I. (2008); An empirical study of lazy multilabel classification algorithms, *Proc. of the 5th Hellenic Conference on Artificial Intelligence*, 401–406, 2008.

[16] Shafigh P., Hadi S.Y., Sohrab E. (2013); Gravitation based classification. *Information Sciences*, 220(1), 319–330, 2013.

[17] Tsoumakas G., Vlahavas I. (2007); Random k-Labelsets: An ensemble method for multilabel classification, *Machine learning: ECML 2007*, 406–417, 2007.

[18] Wright W. E. (1977); Gravitational clustering, *Pattern Recognition*, 9(3), 151–166, 1977.

[19] Wang C., Chen Y.Q. (2005); Improving nearest neighbor classification with simulated gravitational collapse, *Proc. of the Intl. Conf. on Advances in Natural Computation*, 3612, 845–854, 2005.

[20] Wena G., Wei J., Wang J., Zhou T., Chen L. (2013); Cognitive gravitation model for classification on small noisy data, *Neurocomputing*, 118(11), 245–252, 2013.

[21] Younes Z., Abdallah F., Denceux T. (2008); Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies, *Proc. of the 16th Eropean Signal Processing Conference*, 297–308, 2008.

[22] Yang B., Peng L., Chen Y., Liu H., Yuan R. (2006); A DGC-based data classification method used for abnormal network intrusion detection, *Proc. of the Intl. Conf. on Neural Information Processing*, 4234, 209–216, 2006

[23] Zhang M.L., Zhou Z.H. (2006); Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10), 1338–1351, 2006.

[24] Zhang M.L., Zhou Z.H. (2007); ML-KNN: A lazy learning approach to multi-label learning, *Pattern Recognition*, 40(7), 2038–2048, 2007.

[25] Zhang M.L., Zhou Z.H. (2014); A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8), 1819–1837, 2014.