Appl Intell (2015) 43:892–912
DOI 10.1007/s10489-015-0689-3

An improved data characterization method and its application in classification algorithm recommendation

Guangtao Wang[1] · **Qinbao Song**[1] · **Xiaoyan Zhu**[1]

Published online: 2 July 2015

Abstract Picking up appropriate classification algorithms for a given data set is very important and useful in practice. One of the most challenging issues for algorithm selection is how to characterize different data sets. Recently, we extracted the structural information of a data set to characterize itself. Although these kinds of characteristics work well in identifying similar data sets and recommending appropriate classification algorithms, the extraction method can only be applied to binary data sets and its performance is not high. Thus, in this paper, an improved data set characterization method is proposed to address these problems. For the purpose of evaluating the effectiveness of the improved method on algorithm recommendation, the unsupervised learning method EM is employed to build the algorithm recommendation model. Extensive experiments with 17 different types of classification algorithms are conducted upon 84 public UCI data sets; the results demonstrate the effectiveness of the proposed method.

Keywords Classification algorithm recommendation · Classification · Data set characteristics extraction

1 Introduction

Classification has been a research hotspot in the area of data mining for many years, resulting in a huge number of classification algorithms come forth. Moreover, new and modified algorithms are proposed continuously. Thus, there are many alternative choices for a classification problem. However, both No Free Lunch theorem [49] and reported results [4, 6, 9, 10, 32, 34, 35] state that there does not exist any specific algorithm which performs well on all data sets. Therefore, the question "Which algorithm should be picked up for a data set at hand?" is raised. One solution is to evaluate the classification performance of all candidate algorithms by trying them on the data set. Yet, this is not practicable in many situations since it is time-consuming to run such a huge number of candidate algorithms especially on a large data set.

Fortunately, it has been demonstrated that the optimal algorithm for a data set usually varies with the data set, and there is an intrinsic interaction between the performance of an algorithm and the characteristics of a data set [46]. Thus, an alternative solution is constructing an automatic algorithm recommendation model based on the interaction to directly recommend appropriate algorithms for the data set at hand. This paper focuses on how to capture this kind of interaction and construct an effective recommendation model.

Exploring the interaction between the characteristics of a data set and the performance of candidate algorithms plays a critical role in automatic algorithm recommendation. Generally, it is viewed as a learning problem where the features are the characteristics of data sets and the learning target corresponds to the performance of classification algorithms [8, 10, 25, 29, 45, 46]. From this perspective, algorithm recommendation consists of two steps: i) meta-knowledge collection, including characterization of different data sets and performance evaluation of candidate algorithms, and ii) recommendation model construction with the meta-knowledge.

The performance of candidate algorithms can be easily estimated by the well-known and commonly-used

✉ Guangtao Wang
gtwang@mail.xjtu.edu.cn

[1] Department of Computer Science, Technology, Xi'an Jiaotong University, Xi'an Shaanxi, 710049, China

cross-validation strategy, while the data set characterization is full of challenges [44]. The characteristics of a data set are a group of measures reflecting the properties of the data set. Theoretically, any measure, as long as it is extracted from a data set and is able to portray the intrinsic properties of the data set, can be used as a data set characteristic. However, for algorithm recommendation, the characteristics of a data set not only should be easy to calculate, but also affect the performance of the algorithms [14]. Based on different viewpoints of a data set, several different kinds of data set characterization methods have been proposed, including i) statistic and information-theory based method [10]; ii) model structure based method, which first maps the classification problem into a special data structure (e.g., decision tree) and then extracts the properties of the structure as the meta-features, such as the height of the tree, [6, 37]; iii) land-marking based method, which characterizes a classification problem by the performance metrics of a set of simple learners (also referred to as land-marker) on the problem [7, 19, 38]; iv) problem complexity based method, which extracts a set of measures reflecting the source of the difficulty to solve a classification problem as the meta-features [26, 27]. However, there is still no effective method to predetermine which ones are really relevant.

Recently, we proposed a novel feature vector[1] extraction method for algorithm recommendation [46]. The method uses structural information based feature vectors to characterize the learning problems, which is quite different from the existing ones. Specially, it first calculates the frequencies of the 1-itemsets and 2-itemsets of a data set, then extracts the quantiles of these frequency sequences as the feature vectors of the data set, which shows better performance than two other kinds of characterization methods (e.g., statistic and information-theory and problem complexity based ones) in constructing algorithm recommendation models. However, the feature vector still has room for improvement. First, it can only be applied to binary data sets since the frequency of an itemset is calculated based on parity feature function. This will be quite time-consuming and space-consuming when applying on a high-dimension data set. Second, it does not distinguish the learning target and the other features when searching for the itemsets of a data set. This is because the learning target plays a quite different role from the other features, and they should be treated differently. Third, the extraction of the quantiles is time-consuming especially when the number of itemsets is large since we need to sort these itemsets based on their frequencies at first. Moreover, the quantiles are not robust to the extreme frequency values and the distribution of frequencies.

[1] In this paper and [46], the feature vector of a data set is made of the characteristics of the data set.

In order to overcome these drawbacks, in this paper, we propose an improved data set characterization method which is more effective in feature vector extraction. And the major contributions of the method are: i) using the conjunction feature function instead of the parity feature function to calculate itemset frequencies since these two functions are equivalent, and the resulting feature vector can be employed to characterize non-binary data sets directly; ii) the learning target is treated differently from the other features when searching for the itemsets and more information will be represented in the feature vector; iii) instead of extracting the quantiles, a new itemset frequency unification method is proposed to improve the robustness and the computation efficiency of the feature vector.

The remainder of the paper is organized as follows. In Section 2, we summarize the related work. In Section 3, we present the improved data set characterization method. In Section 4, we introduce the clustering based algorithm recommendation method. In Section 5, with the proposed algorithm recommendation method, we experimentally compare the improved data set characterization method with the existing five different data set characterization methods. Finally, in Section 6, we conclude this paper.

2 Related work

Most of the existing algorithm recommendation methods have been devoted to make use of the knowledge on the performance of algorithms and the characteristics of data sets. These methods can be either of theoretical or experimental origin.

The theoretical ones usually focus on capturing the knowledge of the applicability of certain classification algorithms by analyzing their representational biases [13]. However, not the applicability of all classification algorithms can be theoretically analyzed. Moreover, most algorithms need to preassign values to parameters to get good performance. All these situations make the theoretical analysis more difficult.

Due to the complication of the theoretical analysis, most recommendation methods are from experimental origin [2, 4, 7, 8, 10, 23, 30–32, 34, 35, 37, 38, 42, 43, 46]. These experimental methods aim towards learning the interaction between the data set characteristics and the performance of algorithms upon a set of historical learning problems. The differences mainly lie in their data set characteristics and recommendation procedures. [8] characterized data sets with a set of statistical and information theoretic based measures, and utilized C4.5 to generate recommendation rules. Although these rules were not precise enough, it still narrowed down the choices of candidate algorithms. After that, [10] employed a *k*-Nearest Neighbors based meta-learning

method to recommend algorithms with these characteristics. This time, they provided a ranked list of candidate algorithms, but did not tell users which algorithms should be picked up in practice. [4] used C5.0 to generate recommendation rules, where the characteristics of a data set are measured by following those in [42, 43] that are still statistical and information based. [38] characterized a data set by the performance of a set of simple classifiers (i.e., landmarkers) on the data set, and rule based learners were employed to produce recommendation rules. [37] extracted a set of measures based on the induced decision tree to characterize data sets, and built the recommendation model with a *k*-Nearest Neighbors method that also outputs a ranked list of candidate algorithms. [27] studied a set of measures to characterize the complexity of the classification problems and analyzed how these measures affect the performance of algorithms.

Data set characterization is the most challenging issue [13], algorithm recommendation performance depends heavily on its effectiveness. Therefore, many data set characterization methods have been proposed and the existing data set characteristics can be categorized into four groups. They are: i) the statistical and information-theory based [2, 10, 24, 32, 33, 45], ii) the model structure based [6, 37], iii) the land marking based [7, 19, 38], and iv) the problem complexity based [26, 27]. The brief introduction of these methods is listed as follows (The specific measures extracted by these methods shown in the Appendix A).

i) *Statistical and information-theory based method*

The statistical and information-theory based method is the most widely-used in the filed of algorithm recommendation [10, 24, 32–35, 45]. The prominent examples based on these measures are the projects ESPRIT Statlog (1991-1994) and METAL (1998–2001). The measures extracted by this method generally include the data set characteristics such as, number of features, number of instances, number of target concepts, ratio of missing values, ratio of binary features, information gain between the feature and the target concept, and correlation coefficient between features, etc.

ii) *Model structure based method*

This kind of method represents a data set in a special data structure which can embed the complexity of the data set. Then, it extracts the characteristics of the structure to describe the data set.

In the area of algorithm recommendation, a well-known structure is the induced decision tree which is used to model a data set. Bensusan [6] proposed to capture the information from the induced decision tree for describing the learning complexity. And ten measures are extracted from the decision tree, such as the ratio of the number of nodes to the number of features, the ratio of the number of nodes to the number of instances, etc. Afterwards, Peng et al. [37] re-analyzed the characterization of decision trees, and proposed some new measures to describe the structural properties of decision trees.

iii) *Landmarking based method*

This kind of method falls within the concept of landmarking [7, 19, 38]. It was proposed based on the idea that the performance of the candidate algorithms could be related to the performance of a set of simple learners (also called landmarkers). The performance (e.g., accuracy) of these landmarkers is employed to characterize a data set. Evidently, this kind of measures depends on the choice of landmarkers. In practice, it should be ensured that there exist significant differences among the chosen landmarkers in terms of learning mechanism.

iv) *Problem complexity based measures*

The problem complexity based method is exploited to describe the classification data sets (or problems) in [26, 27]. By analyzing the source of difficulty in solving a classification problem, the method focuses on the description of the geometrical complexity of the problem and emphasizes the geometrical characteristics of the distributions of the classes. The metrics reflecting the way in which different classes are separated or interleaved (and being relevant to learning performance) are identified as the measurement of the problem's complexity. Such as Fisher's discriminant ratio, and the nonlinearity of linear/non-linear learning algorithm, the percentage of instances in the problem that linear the class boundary, etc.

In addition to the above mentioned data set characterization methods, recently, we proposed using a structural information based feature vector to characterize data sets and constructed an algorithm recommendation model with the *k*-Nearest Neighbors learner [46]. The proposed data set characterization method is different from the existing four ones in the ways of characterizing a data set and dealing with similar data sets. And it adopts the frequencies of the itemsets with respect to the parity function to depict a data set. The experimental results show that the feature vector is better than the traditional statistical and information theoretic-based or the problem complexity based features. However, this method can only work on binary data sets due to the fact that the parity function is employed to calculate the frequency of the itemsets, and deals with features and the learning target equally although they play different roles in a data set. Meanwhile, when extracting the quantiles of itemset frequencies, it needs to sort the itemsets, which

is time-consuming especially when number of itemsets is large. On the contrary, the improved method can work on both binary data sets and ordinary data sets; it takes into account the difference between features and the learning target when extracting itemsets, and is more effective since the unification method does not need to sort the itemsets anymore. Moreover, a clustering algorithm is used to find the similar data sets for a new one instead of the *k*-Nearest Neighbors method. This is because that, for different data sets, the optimal number of nearest neighbors can vary. The clustering based method can automatically obtain the appropriate number of nearest data sets, and so it is more flexible than the *k*-Nearest Neighbors method.

3 Improved data set characteristics extraction method

3.1 Basic concepts

To facilitate the understanding of the improvements, we first introduce some basic concepts from Tatti's [47] as the preliminary to state the rationality of the improvements.

Definition 1 Binary data set A data set is a binary data set if and only if all its attribute values are *0* or *1*.

An ordinary data set can be transformed into a binary data set without losing any semantic information. For the binary data set, we only focus on whether an instance has some properties or not. For example, if an attribute value is 1, we say the related instances have the corresponding property, and vice versa. It does not concern what the property is. In this way, the statistics and the structural information of a data set can be evaluated. Next, we introduce the process to transform an ordinary data set into a binary data set.

Let $D = \{d_1, d_2, \cdots, d_n\}$ be an ordinary data set with n instances, and AVSet = $\{AV_1, AV_2, \cdots, AV_m\}$ be its attribute value space, where AV_i is the domain of the ith attribute A_i and m is the number of attributes. The instance d_i $(1 \leq i \leq n)$ can be denoted as a multi-tuple $(x_{i,1}, x_{i,2}, \cdots, x_{i,m})$, where $x_{i,j} \in AV_j$ $(1 \leq j \leq m)$. Suppose $D_B = \{\hat{d}_1, \hat{d}_2, \cdots, \hat{d}_n\}$ is the corresponding binary data set of D, and $\{A_1^B, A_2^B, \cdots, A_k^B\}$ is the attribute set of the binary data set D_B, where $k = \sum_{i=1}^{m} |AV_i|$ and the domain of each attribute A_i^B is $\{0, 1\}$.

For each instance $d_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,m})$ $(1 \leq i \leq n)$ of D, let $p_{i,j}$ be the index of value $x_{i,j}$ $(1 \leq j \leq m)$ in attribute value domain AV_j. For $x_{i,j}$, its corresponding attribute index in D_B can be calculated via $Index(x_{i,j}) = \sum_{t=1}^{j-1} |AV_t| + p_{i,j}$. Therefore, for instance d_i, a set of indexes IndSet = $\{Index(x_{i,1}), Index(x_{i,2}), \cdots, Index(x_{i,m})\}$ can be obtained. With IndSet, d_i can be transformed into the corresponding binary instance $\hat{d}_i = (\hat{x}_{i,1}, \hat{x}_{i,2}, \cdots, \hat{x}_{i,k})$ $(1 \leq i \leq n)$ of D_B via the following mapping function:

$$\hat{x}_{i,j} = \begin{cases} 1, \text{if } j \in \text{IndSet}(1 \leq j \leq k) \\ 0, \text{otherwise} \end{cases}.$$

Definition 2 Itemset For a binary data set D_B, each of its attributes is called an item. If the set of all attributes of D_B is denoted as $ASet_B = \{A_1^B, A_2^B, \cdots, A_k^B\}$, then a non-empty subset of $ASet_B$ is referred to as an itemset.

An itemset with k items is called a k-itemset. Two particular kinds of itemsets have been employed to describe a binary data set [46], they are one-itemset $I = \{\{A_i^B\}|1 \leq i \leq k\}$ and two-itemset $II = \{\{A_i^B, A_j^B\}|1 \leq i < j \leq k\}$. The pattern information of one item set I explicitly tells the information of each individual attributes. In contrast, the two-item set II tells the correlation information of the attribute pairs. They describe the different but complementary aspects of the data set.

Definition 3 Feature function A feature function maps a point in the sample space Ω to a real vector, and it is defined as $S : \Omega \to R^N$, where N represents the dimensionality of the range space of S.

There are many kinds of feature functions. A boolean feature function $S : \Omega \to \{0, 1\}$ can be used to map a binary vector to a binary value. Let $\Omega = \{\omega | \omega = (\omega_1, \omega_2, \cdots, \omega_k)\}$, where ω_i takes the value of attribute A_i^B, be the sample space of the binary data set D_B, and $B = \{A_{i_1}^B, A_{i_2}^B, \cdots, A_{i_L}^B\} \subset \text{ASet}_B$ be an itemset, two particular boolean feature functions can be defined as follows.

Definition 4 Conjunction function A conjunction function S_B over itemset B is defined as:

$$S_B(\omega) = \omega_{i_1} \wedge \omega_{i_1} \wedge \cdots \wedge \omega_{i_L}.$$

This definition indicates that conjunction function S_B results in 1 if and only if all the values in corresponding itemset B are true.

Conjunction functions are frequently used in data mining for frequency itemset mining and association rule mining [1].

Definition 5 Parity function A parity function T_B over itemset B is defined as:

$$T_B(\omega) = \omega_{i_1} \oplus \omega_{i_1} \oplus \cdots \oplus \omega_{i_L},$$

where $\oplus$ is *XOR* operator.

The function T_B results in 1 if and only if the number of the true values corresponding B is odd.

With the definitions of conjunction function S_B and parity function T_B, we can define the frequency θ of itemset B on data set D_B by averaging the S_B (or T_B) values over D_B as follows.

Definition 6 Frequency of an itemset The frequency θ of itemset B on data set D_B in terms of the conjunction function S_B is defined as:

$$\theta(S_B, D_B) = \frac{1}{|D_B|} \sum_{\omega \in D_B} S_B(\omega).$$

Similarly,

$$\theta(T_B, D_B) = \frac{1}{|D_B|} \sum_{\omega \in D_B} T_B(\omega)$$

denotes the frequency θ in terms of the parity function T_B.

Suppose $\mathbb{F} = \{B_1, B_2, \cdots, B_N\}$ is a set of itemsets with N itemsets, where B_i is an itemset. By setting $\mathbb{F}$ to B, conjunction function $S_B(\omega)$ can be extended to $S_{\mathbb{F}}(\omega) = [S_{B_1}(\omega), S_{B_2}(\omega), \cdots, S_{B_N}(\omega)]$. Similarly, parity function $T_B(\omega)$ can be extended to $T_{\mathbb{F}}(\omega) = [T_{B_1}(\omega), T_{B_2}(\omega), \cdots, T_{B_N}(\omega)]$. Both $S_{\mathbb{F}}$ and $T_{\mathbb{F}}$ can be used to compute the constrained minimum (CM) distance between two data sets. And the constrained minimum (CM) distance is defined as follows.

Definition 7 Constrained minimum (CM) distance, let H be a feature function, assume that $Cov(H)$ is invertible, the CM distance between two data sets D_1 and D_2 can be defined as

$$d_{CM}(D_1, D_2|H) = (\theta_1 - \theta_2) \cdot Cov(H)^{-1} \cdot (\theta_1 - \theta_2)^T,$$

where θ_i is the frequency of feature function H over D_i $(i = 1 or 2)$.

Lemma 1 *If the set of itemsets $\mathbb{F}$ is anti-monotonic or downward closed, then the constrained minimum (CM) distances $d_{CM}(D_1, D_2|S_{\mathbb{F}})$ and $d_{CM}(D_1, D_2|T_{\mathbb{F}})$ between two data sets D_1 and D_2, which are calculated based on the frequencies of $\mathbb{F}$ in terms of feature functions $S_{\mathbb{F}}$ and $T_{\mathbb{F}}$ respectively, are equivalent, i.e. $d_{CM}(D_1, D_2|S_{\mathbb{F}}) = d_{CM}(D_1, D_2|T_{\mathbb{F}})$.*

Where the collection $\mathbb{F}$ of itemsets is said to be anti-monotonic or downward closed if each non-empty subset of an itemset included in $\mathbb{F}$ is also included in $\mathbb{F}$. For example, suppose that $\mathbb{F}$ is defined as $\{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}$, it is not anti-monotonic since there exists one subset $\{b, c\} \subseteq \{a, b, c\}$ but not in $\mathbb{F}$. And the set of itemsets $\{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}\}$ is anti-monotonic.

This lemma demonstrates that the CM distances are equal for the frequencies in terms of conjunction and parity feature functions. The reason is that if $\mathbb{F}$ is anti-monotonic, there exists an invertible matrix A such that $S_{\mathbb{F}} = A \times T_{\mathbb{F}}$, i.e., the conjunction function $S_{\mathbb{F}}$ can be achieved from the parity function $T_{\mathbb{F}}$ by an invertible linear transformation that guarantees the equality of distances, please refer to [47] for the detailed proof.

However, when calculating the CM distance, the parity function possesses significant advantages: 1) the covariance matrix $Cov[T_{\mathbb{F}}]$ of the parity function is very simple, it is a diagonal matrix having 0.5 at the diagonal; and 2) the CM distance can be directly calculated when we know the frequencies of $\mathbb{F}$ on D_1 and D_2.

This lemma can be confirmed by the following simple example. Figure 1 gives two example binary data sets D_1 and D_2. According to the data sets, we can get an anti-monotonic set of itemsets $\mathbb{F} = \{\{F_1 = 0\}, \{F_1 = 1\}, \{F_2 = 1\}, \{F_2 = 0\}, \{F_1 = 0, F_2 = 0\}, \{F_1 = 0, F_2 = 1\}, \{F_1 = 1, F_2 = 0\}, \{F_1 = 0, F_2 = 1\}\}$. With $\mathbb{F}$, the frequencies of D_1 with respect to the conjunction and parity functions will be calculated as $\theta(S_1, D_1) = < 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0, 0.5 >$ and $\theta(T_1, D_1) = < 0.5, 0.5, 0.5, 0.5, 0, 0.5, 0.5, 0 >$. Similar, we can get $\theta(S_2, D_2) = < 0.5, 0.5, 0.5, 0.5, 0, 0.5, 0.5, 0 >$ and $\theta(T_2, D_2) = < 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0, 0.5 >$. Therefore, $\theta(S_1, D_1) - \theta(S_2, D_2) =< 0, 0, 0, 0, 0.5, -0.5, -0.5, 0.5 >$ and $\theta(T_1, D_1) - \theta(T_2, D_2) = < 0, 0, 0, 0, -0.5, 0.5, 0.5, -0.5 >$. Moreover, considering that the covariance matrix $Cov[T_{\mathbb{F}}]$ is a diagonal matrix, it is obvious that there exist an identity matrix A resulting in $S_{\mathbb{F}} = A \times T_{\mathbb{F}}$, and further $d_{CM}(D_1, D_2|S_{\mathbb{F}}) = d_{CM}(D_1, D_2|T_{\mathbb{F}}) = 2$.

3.2 Review of the former data set characteristics extraction method

We proposed a novel data set characteristics extraction method in [46]. The method consists of three steps: data set transformation, data set feature extraction, and data set feature unification.

Step 1: *Data set transformation*
This step transforms a given ordinary data set D into the corresponding binary data set D_B.

Data set D_1

F_1	F_2
0	0
1	1

Data set D_2

F_1	F_2
0	1
1	0

Fig. 1 Example data sets

This makes the computation of the parity feature function T_B over an itemset B, and further the computation of the frequency of the itemset B on D_B is easy.

The drawback of this step is that the transformation will lead to a high-dimensional binary data set D_B, and further result in high space storage and time consumption in the calculation of the frequency of itemsets in Step 2.

The conjunction function S_B is true if and only if all the corresponding binary attribute values of the itemset B in D_B are true. That is, all the corresponding attribute values appear in an instance of D at the same time. In this case, if the parity feature function is equivalent to the conjunction feature function in describing a data set, we do not need to transform the ordinary data set D into a binary data set D_B. This is because, for the conjunction function, we can directly test whether the attribute values of an itemset simultaneously appear in an instance of D or not rather than transforming D into D_B in advance.

Step 2: *Data set feature extraction*

This step extracts the frequency of the parity feature function T_B on the transformed data set D_B with respect to the one-item set I and two-item set II, and these two frequency sequences are viewed as the data set feature.

As we know, there are two kinds of attributes in a data set: the learning target attribute and the ordinary attributes. The learning target attribute is quite different with the ordinary attributes, so they should not be treated the same way. However, in this step, the two-item set II can not distinguish these two kinds of attributes.

Step 3: *Data set feature unification*

For different data sets, the length of the one-item set I or two-item set II might be different as well, so it is difficult to compare different data sets according to the data set feature extracted in Step 2. In order to overcome this problem, the method unifies the frequency sequences by extracting its nine quantiles.

However, extracting the quantiles of a vector to approximate the distribution of the vector is not robust enough, and it is so sketchy to approximate the distribution only by the nine quantile values especially when the distribution is extremely skewed or sparse. Moreover, the quantile extraction needs to sort the elements of the vector in advance, this is time-consuming especially when the number of the elements of the vector is huge.

3.3 Improvements on feature function and itemset

1. *The conjunction function is used to replace parity function*

The conjunction function on a given binary itemset resulting in 1 means that all the items of the itemset are true. This means, according to the data set transformation process, some feature values of the ordinary data set simultaneously appear in the instance. In this view, when calculating the frequency of an itemset in terms of conjunction function, it is not necessary to transform a given ordinary data set into a binary form in advance. That is, the frequency of the itemset is just the proportion of instances including the feature values corresponding to the itemset simultaneously in the ordinary data set. However, the parity feature function is applicable only on the binary data set.

Fortunately, we find that the conjunction feature function is equivalent to the parity feature function in describing a binary data set by the frequencies of one-item sets and two item sets. This can be directly induced from Lemma 1. First, the collection of itemsets consisting all the one-item sets and two-item sets is obviously anti-monotonic. Then, according to Lemma 1, the CM distance is equal for the conjunction function and the parity function. The difference is reflected only on the complexity of the calculation of the distance. However, we just focus on the frequencies of the itemsets rather than the calculation process of the distance. Thus, the conjunction function can be used to replace the parity function in calculating the frequencies of itemsets.

Consequently, with the conjunction function, we can directly calculate the frequency of itemsets for an ordinary data set without transforming it into binary format. This will not only save up the runtime of feature vector extraction, but also reduce the memory consumption especially when the number of feature values of the ordinary data set is large since we do not need to keep the binary data set in memory.

2. *Two-item sets are divided into two groups carrying different information*

Given a classification data set, there are two kinds of attributes, the ordinary attributes used to describe the data and the learning target attribute. They play different roles in training a classifier. However, in the previous work [46], when collecting the two-item sets, all the attributes are equally treated. In order to distinguish these two kinds of attributes, we divide a two-item set into two groups. The first is the two-item set where both of its items are ordinary attributes, which tells us the correlation between ordinary attributes. The second is the two-item set where its one item is the ordinary attribute and another one is the target attribute, which

tells us the correlation between the ordinary attribute and the learning target attribute.

Finally, for a given data set, we can get a set of itemsets consisting in three parts, i) one-item sets (denoted as I) carrying the information of the individual attributes, ii) two-item sets whose items are both of ordinary attribute (denoted as II_A), and iii) two-item sets whose items consists of ordinary attribute and learning target attribute (denoted as II_T). The frequencies of these itemsets form a feature vector V to characterize the data set. Consequently, the feature vector V consists three sub-feature vectors V_I, V_{II_A} and V_{II_T} corresponding to one-item sets I, two-item sets II_A and II_T, respectively.

3.4 Improvement on data set feature unification

For different data sets, the number of distinct attribute values and the number of learning target values are usually different as well. Consequently, the length of feature vector V_I will vary with the data sets. So do the lengths of feature vectors V_{II_A} and V_{II_T}. This makes it impossible to compare the feature vectors of different data sets directly.

In order to overcome this problem, for a given feature vector, in [46], the vector is sorted in ascending order at first, then a specific number of quantiles are extracted from the sorted vector to characterize the original vector. They are the *minimum*, *1/8 quantile*, *2/8 quantile*, *3/8 quantile*, *4/8 quantile*, *5/8 quantile*, *6/8 quantile*, *7/8 quantile* and the *maximum*. This method unifies the vectors with different lengths into a unified form and makes it comparable among different data sets.

However, extracting the quantiles of a vector to approximate the distribution of the vector is not robust enough. The extreme values of the vector, the minimum and the maximum might be sensitive to the data set with small fluctuation of the attribute value distribution. Moreover, it is so sketchy to approximate the distribution only by these nine values especially when the distribution is extremely skewed or sparse.

For handling these problems and achieving robust measures to approximate the distribution of a vector, we extract the distribution of the vector by mapping the elements of the vector into a specified number of containers. Then, we count the number of elements and calculate the proportion of elements in each container as the measures to describe the vector. The details of this process are introduced as follows.

Let $V = \{e_1, e_2, \cdots, e_K\}$ be a given vector with K elements, and $C = \{C_1, C_2, \cdots, C_L\}$ be a set of L containers, where $e_i \in \mathbb{R}$ $(1 \leq i \leq K)$, $C_j \subset \mathbb{R}$ $(1 \leq j \leq L)$ and $C_i \cap C_j = \phi$ $(1 \leq i \neq j \leq L)$. Suppose that the range of values of C_j is $(LB_{C_j}, UB_{C_j}]$, the following function is defined to determine whether or not the element e_i falls in the container C_j:

$$\delta(e_i, C_j) = \begin{cases} 1, \text{if } e_i \in (LB_{C_j}, UB_{C_j}] \\ 0, \text{otherwise} \end{cases}.$$

With this function, the number of elements falling in container C_j is calculated by $N_j = \sum_{i=1}^{K} \delta(e_i, C_j)$. Finally, by normalizing these numbers, we can get a set of measures, i.e., $\{N_1/K, N_2/K, \cdots, N_L/K\}$, to characterize the given vector V, where $K = \sum_{i=1}^{L} N_i$.

It is well known that the frequency of an itemset varies between 0 and 1, and in this paper, we bin the elements of a vector into 10 equally spaced containers. This means $L = 10$ and the value range of C_j $(1 \leq j \leq 10)$ is $(0.1 \times (j-1), 0.1 \times j]$. After applying this process to each of the three vectors V_I, V_{II_A} and V_{II_T}, the length of the final feature vector for any data set is fixed to 30, so different data sets can be uniformly represented and compared.

According to the above process, we can unify a given vector in linear-time since the count in each container can be calculated by scanning the vector once. That is, the time complexity of the improved data set characteristic method is $O(K)$. In contrast, the time complexity of the former method [46] is $O(K \log(K))$ since it must sort the elements of the vector in advance. So the improved method is more effective.

4 Classification algorithm recommendation

4.1 General view

As there is an intrinsic relationship between data set features[2] and the performance of the classification algorithms on the corresponding data set [2, 4, 8, 10, 23, 30, 36, 41, 42], it has been experimentally proven in [9, 10, 46] that if some data sets are similar in terms of their features/characteristics, then the performance of classification algorithms on these data sets is similar as well. Therefore, if an algorithm performs well on a cluster of data sets, it is reasonable to recommend this algorithm for a data set that is similar to the data sets within the cluster, where the similarity between the data set and the given cluster can be calculated on the basis of the features (e.g., the features listed in Appendix A) extracted from the data sets.

Inspired by the above idea, we propose an unsupervised learning based automatic classification algorithm recommendation method. The proposed method consists of i)

[2] As the information we extracted from a data set is quite different from that of the existing work, we use "feature" instead of characteristics hereafter.

data set clustering, ii) appropriate classification algorithms identification for a cluster, and iii) classification algorithm recommendation for a new data set, as shown in Fig. 2.

1. *Data set clustering*
   At first, the data set features are extracted on each of the historical data sets. These features can be either the ones in Appendix A or the improved ones in this paper. Afterwards, the well-known clustering algorithm is performed to cluster the historical data sets into different clusters according to their feature vectors. The similarity between two data sets is calculated based on the distance between the features of these two data sets. Section 4.2 shows the details of this task.
2. *Applicable classification algorithm identification for a cluster*
   A cluster consists of a set of historical data sets with something in common. In this paper, we believe that there would be some classification algorithms performing well on most data sets of the cluster. These classification algorithms can be identified as the applicable ones on the cluster. And this can be accomplished by evaluating the performance of all the candidate classification algorithms on the data sets of the cluster in a specific way. Section 4.3 provides the details of this task.
3. *Classification algorithm recommendation for a new data set*
   Once identifying the applicable classification algorithms for a cluster, the algorithm recommendation for a new data set is straightforward. For the new data set, first, its feature vector is extracted; then, by calculating the distance between the feature vector and the clustering center of each cluster, the new data set is classified into a specific cluster whose clustering center is closer to the feature vector; finally, the applicable algorithms on the cluster identified in the former step are recommended for the new data set.

4.2 Data set clustering

Instances can be clustered by an unsupervised learning algorithm according to the features that describe them. However, there are no explicit features for data sets. Moreover, data sets are usually different from each other not only in terms of the number and the types of features depicting instances, but also in the data formations. Therefore, it is very hard to cluster data sets directly.

In order to facilitate data set clustering, data sets have to be preprocessed so that something like features portraying the instances are extracted. The extracted data are referred to as the feature vector of a data set. It should have the following properties: i) it is able to reserve both structural and statistical essentials of a data set, and ii) it is a unified formation of different types of data sets. Such as, with the guideline of each of the five kinds of existing features in Appendix A, each data set can be represented as a feature vector.

In this paper, based on Tatti's work [47] and our previous work [46], an improved feature vector extraction method (which is introduced in Section 3) is proposed. Applying the feature extraction method to each data set, the corresponding feature vector is obtained. As the feature vectors of different data sets have a unified formation, the well-known and commonly-used unsupervised learning algorithms can be used to group the data sets into clusters. The similarities among different data sets are evaluated based on the Euclidean distances among these feature vectors. The smaller the distance, the more similar between two data sets.

After data set clustering, how to measure or characterize each cluster is critical in the subsequent recommendation

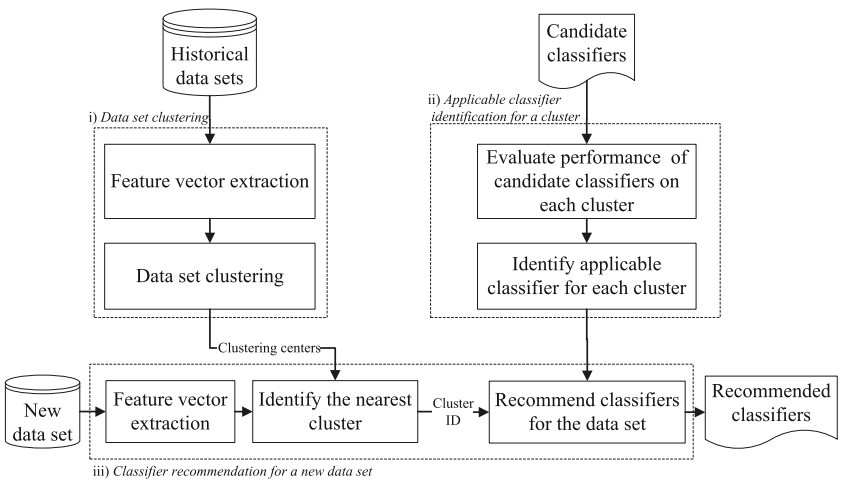

Fig. 2 General view of the proposed recommendation method

process. In this paper, each cluster is characterized by a representative vector which is regarded as the clustering center of the cluster. The clustering center is generated according to the feature vectors of all the data sets of the cluster, and it can be defined as the mean of the data set feature vectors. It contains the general information of the features of these data sets in a cluster, and has the same format with the data set feature vectors. Suppose there are k data sets in a cluster, and the $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k\}$ is a set of feature vectors corresponding to these data sets, the cluster center can be defined as $\mathbf{X}_{center} = \sum_{i=1}^{k} \mathbf{X}_i$.

4.3 Applicable algorithm identification over a cluster

Identifying applicable classification algorithms over a given cluster is straightforward: i) apply the candidate algorithms to each data set of the cluster; ii) evaluate the performance of these algorithms on the data sets; and iii) the candidates with better performance are selected.

When evaluating the performance of a learning algorithm on a given data set, the multi-criteria evaluation measure ARR (*Adjusted Ratio of Ratios*), which combines information about the classification accuracy and total execution time of the learning algorithm, is borrowed from [10] to fit our context.

Let ASet = $\{A_1, A_2, \cdots, A_M\}$ be a set of M candidate classification algorithms, and DSet =$\{D_1, D_2, \cdots, D_N\}$ represent N data sets in the given cluster. Suppose acc_i^j and t_i^j are the classification accuracy and runtime of algorithm A_i on data set D_j $(1 \le i \le M, 1 \le j \le N)$, respectively. Then ARR of algorithm A_i to algorithm A_j over data set D_k is defined as

$$ARR_{A_i,A_j}^{D_k} = \frac{acc_i^k / acc_j^k}{1 + \alpha \cdot \log(t_i^k / t_j^k)} (1 \le i \ne j \le M, 1 \le k \le N), \tag{1}$$

where α denotes the user-defined relative importance of accuracy and runtime. In particular, this parameter represents the amount of accuracy the user is willing to trade for a 10 times speedup or slowdown. For example, $\alpha = 10\ \%$ means that the user is willing to trade 10 % of accuracy for 10 times speedup/slowdown.

It is noted that, differing from the performance evaluation measure used in [46], ARR is dimensionless due to the fact that it is calculated based the ratios of performance metrics (i.e., accuracy/runtime). So the value of ARR is not sensitive to the unit of runtime when comparing with the performance value in [46].

Equation (1) can be directly used to evaluate the performance of two candidate algorithms in ASet. When comparing all these algorithms in ASet when $M > 2$, the performance of any algorithm $A_i \in$ ASet on a given data set D can be evaluated by the metric $\text{ARR}_{A_i}^D$ defined as follow:

$$ARR_{A_i}^D = \frac{1}{M-1} \sum_{j=1 \wedge j \ne i}^{M} ARR_{A_i,A_j}^D \tag{2}$$

To compare two learning algorithms A_1 and A_2 across the N data sets of the cluster, the significant Win/Draw/Loss record [48] and the mean performance are employed.

The significant Win/Draw/Loss record presents three values in terms of a given measure, i.e. the numbers of data sets on which the given measure of algorithm A_1 is statistically significant better than/equal to/worse than that of algorithm A_2, respectively. Here, the given measure can be classification accuracy, runtime, and the multi-criteria evaluation measure $\text{ARR}_{A_i}^D$, etc. In our context, ARR is used.

A win or loss is only counted if the difference between A_1 and A_2 in terms of the given measure is significantly better or worse tested by a paired sign test at a specific significance level. If the test result is significant then it is reasonable to conclude that it is unlikely that the outcome was obtained by chance. Hence the record of wins to losses represents a systematic underlying advantage to one of the algorithms with respect to the type of domains on which they have been tested.

After obtaining the significant Win/Draw/Loss records and the mean performance across all the data sets of a cluster for all candidate algorithms, we are able to identify the r $(r \le M)$ applicable classification algorithms from the candidate algorithm set ASet = $\{A_1, A_2, \cdots, A_M\}$ as follows. It is noted that the parameter r is set as 3 directly. This due to the fact that the first three identified algorithms on a cluster usually lead to a reasonable high hit ratio. That is, it will be more possible to recommend the appropriate algorithm for a data set by narrowing down the choice of the candidate algorithms into the first three algorithms. And this has been also stated in [46].

1. Compare the significant Win/Draw/Loss records between algorithm A_1 and each of $A_i (i = 2, 3, \cdots, M)$. If Count(Win, $A_1 : A_i$) > Count(Loss, $A_1 : A_i$) holds[3] for each pair of $\{A_1, A_i\}$, then algorithm A_1 is selected.
2. Otherwise, the comparison results over Win/Draw/Loss records are conflicting, so candidates A_1, A_i and A_j are not defeated by each other. That is, the case of Count(Win, $A_1 : A_i$) > Count(Loss, $A_1 : A_i$) $\wedge$ Count(Win, $A_j : A_1$) > Count(Loss, $A_j : A_1$) $\wedge$ Count(Win, $A_i : A_j$) > Count(Loss, $A_i : A_j$) appeared. In this situation, the corresponding mean performance over all data sets is calculated. The algorithm with the highest mean performance is selected.
3. Move the selected algorithm from the candidate algorithm set into the appropriate algorithm list AppL.

Repeat the above two steps r times. The r appropriate algorithms in AppL = $\{A'_1, A'_2, \cdots, A'_r\}$ can be viewed as the appropriate candidates on the cluster and further used to recommend to a new classification problem.

Procedure *Appropriate_Alg_Identification*

Inputs :
ASet: $\{A_1, A_2, \cdots, A_M\}$, i.e. the candidate algorithm list;
Cluster: A given cluster consisting a number of data sets;
Output:
AppAlgList: Appropriate algorithms list

```
AppAlgList = φ;
r = 3; //Number of appropriate algorithms being identified
Apply algorithms in ASet on the data sets of Cluster;
WDL = Significant Win/Draw/Loss records for ASet over Cluster;
for i = 1 to r do
    Algorithm list AL = identify algorithm(s) ∈ ASet who lost the least according
    to WDL;
    if |AL| = 1 then
        AppAlg = AL [1];
        else
            AL = identify algorithm(s) ∈ AL who won the most according to
            WDL;
            if |AL| = 1 then
                AppAlg = AL [1];
                else
                    AppAlg = the algorithm ∈ AL with the best mean
                    performance;
    ASet = ASet − {AppAlg };
    AppAlgList = AppAlgList ∪ {AppAlg };
    update WDL by eliminating the row and column corresponding to AppAlg;
return AppAlgList
```

The detailed identification process of appropriate classification algorithms is given by procedure "Appropriate_Alg_Identification".

4.4 Complexity analysis of the recommendation procedure

In this section, we give the complexity analysis of the proposed recommendation method. According to the general view of the algorithm recommendation method in Fig. 2, the recommendation procedure consists of three steps: i) data set clustering, ii) applicable algorithm identification over a cluster and iii) algorithm recommendation for a new data set.

Let N be the number of historical data sets, n and k denote the average numbers of instances and features in each data set, v denote the average number of feature values in each feature, then for step i) of "data set clustering", the time complexity of feature vector extraction will be $O(N \cdot F(n, k, v))$, where $F(n, k, v)$ denotes the time complexity of feature extraction over a data set and depends on n and k. $F(n, k, v)$ varies with different feature vector extraction method and is usually polynomial. For the proposed improved feature vector extraction method in this paper, $F(n, k, v)$ can be represented as $\max(O(n), O(k^2 \cdot v^2))$. Afterwards, a specific clustering algorithm is performed to cluster these N data sets into K clusters on the basis of N feature vectors, in this paper, we employ the iterative clustering method whose time complexity will be $O(N, K, T)$, where T denotes the number of iterations. The time complexity of clustering is usually polynomial as well. Such as for expectation-maximization method, if we employed the cross-validation method to automatically determine the number of clusters K, the time complexity of clustering will be $O(N^2 \cdot T)$.

After data set clustering, suppose there are M candidate classification algorithm, the time complexity of step ii) of applicable algorithm identification will be $O(N \cdot M \cdot C(n, m)) + K \cdot O(M^2 \cot N/K)$, where the first part $O(N \cdot M \cdot C(n, m))$ denotes the complexity of performance calculation of M candidate classification algorithm over N data sets, and $C(n, k)$ represents the average time complexity of each classification algorithm and depends on n and k; the second part $K \cdot O(M^2 \cot N/K)$ denotes the time

[3] Count(Win/Loss, A_1:A_i) denotes the total number of data sets that algorithm A_1 won/lost against A_i.

complexity of identifying the applicable algorithms all the K clusters, and depends on the Win/Draw/Loss collection on each cluster (i.e., $O(M^2 \cot N/K)$) where M algorithms will be compared with each other on N/K data sets.

The step iii) to recommend algorithms for a new data set is straightforward, that is, recommending the applicable algorithms on a specific cluster as the appropriate ones on the new data set. Since that the computation of the distance between the feature vector and each clustering center is quite fast comparing with the process of feature vector extraction, the time complexity of this step will depend on the feature vector extraction on the new data set and can be represented as $O(F(n,k,v))$.

In summary, the time complexity of the proposed algorithm recommendation method is $O(N \cdot F(n,k,v)) + O(N^2 \cdot T) + O(N \cdot M \cdot C(n,m)) + K \cdot O(M^2 \cot N/K) + O(F(n,k,v)) = O(N \cdot F(n,k,v)) + O(N^2 \cdot T) + O(N \cdot M \cdot C(n,m)) + K \cdot O(M^2 \cot N/K)$. It is noted that, once the first two steps are accomplished, the method does not need to carry out the first steps anymore for each new coming data sets, and can recommend the appropriate algorithms directly based on the third step. And the time complexity of the recommendation method will be $O(F(n,k,v))$.

5 Experimental study

In this section, we evaluate the improved data set characteristics for algorithm recommendation by comparing them with the other different kinds of data set characteristics in terms of recommendation performance on a set of benchmark data sets.

5.1 Experimental setup

For the purposes of evaluating the effectiveness of the improved data set characteristic method in classification algorithm recommendation, confirming whether or not the method is useful in practice, and guaranteeing the reproducibility of the results, we set up our experiments as follows.

1) *Benchmark data set*
   84 extensively-used public data sets from UCI repository [5] are employed in the experiments. The statistical summary of these data sets is shown in Table 1 in terms of the number of instances, the number of features and the number of classes.
2) *Characteristics of data set*
   Five different kinds of data set characteristics are employed to be compared with our proposed new feature vector. They are i) the statistical and information-theory based, ii) the model structure based, iii) the land marking based, iv) the problem complexity based and v) the structural information based. The Appendix A shows the details of these characteristics.
3) *Candidate classification algorithms*
   In order to guarantee the generality of the experimental results, 17 different types of classification algorithms are selected as the candidate algorithms.
   They are two probability-based Bayes Network and Naive Bayes Updateable [28]; the tree-based C4.5; two rule-based Ripper [16] and PART [21]; the instance-based algorithm KStar [15]; and the support vector based algorithm Sequential Minimal Optimization (SMO) [40].
   Besides the above seven single learning algorithms, we also employ the ensemble classification algorithms which usually show better classification performance. They are tree based RandomForest [12] and RandomTree; Bagging [11] and Boosting [22] with the four simple classifiers Naive Bayes, instance based IB1 [3], C4.5 and PART, respectively.
4) *Performance evaluation for the candidate learning algorithms*
   In our experiment, the multi-criteria metric ARR (see (2)) is employed to evaluate the performance of the candidate algorithms. The tradeoff coefficient α used in ARR is set to 0.1 %, 1 % and 10 % following the suggestions in [10] to balance the effect between classification accuracy and runtime, respectively. This setting permits us to test the effectiveness of the proposed method in different situations.
5) *Data set clustering*
   When clustering the historical data sets, the well-known unsupervised learning algorithm EM (Expectation Maximization) [17] is employed. One advantage of the EM algorithm is that the appropriate number of clusters is determined by itself via the cross-validation strategy. That is why we employ the clustering algorithm to find the similar data sets rather than k-NN algorithm. Since that we should preassign a parameter, number of nearest neighbors k, for k-NN algorithm, but the optimal setting of this parameter would vary with data sets, and there is no effective method to preassign this parameter.

5.1.1 Experiment Process

The experiment consists of two parts: i) the identification of appropriate classification algorithms for each of the 84 data sets, and ii) the recommendation of classification algorithm for each of the 84 data sets.

Part 1: Identifying the appropriate algorithms for each data set

Table 1 Description of the 84 data sets

ID	Name	Attributes	Instances	Classes	ID	Name	Attributes	Instances	Classes
1	anneal	38	898	6	43	liver-disorders	6	345	2
2	anneal.ORIG	38	898	6	44	lung-cancer	56	32	3
3	arrhythmia	279	452	16	45	lymph	18	148	4
4	audiology	69	226	24	46	mfeat-fourier	76	2000	10
5	australian	14	690	2	47	mfeat-karhunen	64	2000	10
6	autos	25	205	7	48	mfeat-morphological	6	2000	10
7	balance-scale	4	625	3	49	mfeat-zernike	47	2000	10
8	breast-cancer	9	286	2	50	molecular-biology_promoters	58	106	2
9	breast-w	9	699	2	51	monks-problems-1	6	556	2
10	car	6	1728	4	52	monks-problems-2	6	601	2
11	cleve	11	303	2	53	monks-problems-3	6	554	2
12	cmc	9	1473	3	54	mushroom	22	8124	2
13	colic	22	368	2	55	nursery	8	12960	5
14	connect-4	42	13512	3	56	optdigits	64	5620	10
15	credit-a	15	690	2	57	page-blocks	10	5473	5
16	credit-g	20	1000	2	58	pendigits	16	10992	10
17	crx	15	690	2	59	pima	6	768	2
18	cylinder-bands	39	540	2	60	postoperative-patient-data	8	90	3
19	dermatology	34	366	6	61	primary-tumor	17	339	22
20	diabetes	8	768	2	62	segment	19	2310	7
21	ecoli	7	336	8	63	shuttle-landing-control	6	15	2
22	flags	29	194	8	64	sick	29	3772	2
23	german	15	1000	2	65	solar-flare_1	12	323	2
24	glass	9	214	7	66	solar-flare_2	12	1066	3
25	haberman	3	306	2	67	sonar	60	208	2
26	hayes-roth	4	132	3	68	soybean	35	683	19
27	heart-c	13	303	5	69	spambase	57	4601	2
28	heart-h	13	294	5	70	spect	22	267	2
29	heart-statlog	13	270	2	71	spectrometer	102	531	48
30	hepatitis	19	155	2	72	splice	61	3190	3
31	horse-colic.ORIG	21	368	2	73	sponge	45	76	3
32	hypo	23	3163	2	74	tae	5	151	3
33	hypothyroid	29	3772	4	75	tic-tac-toe	9	958	2
34	ionosphere	34	351	2	76	trains	32	10	2
35	iris	4	150	3	77	transfusion	3	748	2
36	kdd_JapaneseVowels_1	14	5687	9	78	vehicle	18	846	4
37	kdd_JapaneseVowels_2	13	4274	9	79	vote	16	435	2
38	kdd_synthetic_control	61	600	6	80	vowel	13	990	11
39	kr-vs-kp	36	3196	2	81	waveform-5000	40	5000	3
40	labor	16	57	2	82	wine	13	178	3
41	led7	7	3200	10	83	yeast	7	1484	10
42	letter	16	20000	26	84	zoo	17	101	7

In order to get stable performance and make best use of the data, the 5×10 folds cross-validation procedure is used to estimate the performance measure ARR. That is, for each data set and each classification algorithm, the 10 folds cross-validation is repeated five times. Each time we randomize the order of the instances of the data set. This is because many of the algorithms exhibit order effects, in that certain orderings dramatically improve or degrade performance

[20]. Randomizing the order of the inputs can help diminish the order effects.

After the 5 × 10 folds cross-validation procedure, we can achieve 50 ARR values of each of the 17 candidate algorithms on a data set. In order to identify the superior algorithms from three or more candidate algorithms (e.g., 17 candidate algorithms in the experiments), the traditional statistical methods usually resort to multiple paired parametric *t*-tests or non-parametric Mann-Whitney tests. However, it has been proven that this approach usually leads to high Type I error[4] [18, 39]. That means the probability that we falsely reject the algorithms with no significant differences to the best one(s) will be high. In order to solve this problem and identify the appropriate algorithms based on these estimated ARRs, we resort to the *Multiple Comparison Procedure* which helps us compare three or more groups of metrics (e.g., ARR) while controlling the probability to make the statistical Type I error. Thus, the non-parametric multiple comparison procedure, Friedman test followed by Holm procedure test which is suggested in [18], is conducted at the significance level 0.05. The non-parametric test being performed is due to the fact that the distribution of the estimated ARRs is usually unknown and it is difficult to guarantee these ARRs following normality distribution and homogeneity of variance, and the non-parametric tests are distribution free methods. In this case, the non-parametric tests are more powerful in detecting the differences among the candidate algorithms.

At first, if the result of Friedman test shows that there is no significant performance differences among these 17 candidate algorithms, all the 17 algorithms are viewed as the appropriate ones. Otherwise, the classification algorithm with the highest ARR is viewed as a reference, and the Holm's procedure test is performed to identify the algorithms from the rest. The algorithms that have no significant differences with the reference are viewed as the appropriate algorithms. Of course, the reference is an appropriate algorithm as well.

Part 2: Recommending classification algorithm for each data set

First, we obtain VECTORS = $\{V_1, V_2, \cdots, V_{84}\}$ by computing the feature vector V_i for each data set D_i ($i = 1, 2, \cdots, 84$). Then we classify each data set into one of the clusters. Procedure "DataSetClassification" shows the details.

Procedure *DataSetClassification*

```
1 DATA = {D_1, D_2, ..., D_84};
2 for each D_i ∈ DATA do
3 |   Compute V_i;
4 |   VECTORS = VECTORS ∪ {V_i};
5 for each V_i ∈ VECTORS do
6 |   VECTORS' = VECTORS − {V_i};
7 |   Clusters = EM (VECTORS');
8 |   ClusterID = classify (V_i, Clusters);
```

In the procedure, function *EM* clusters the feature vectors into several groups, and function *classify* classifies a data set to a cluster whose center is the nearest to the feature vector of the data set.

By far we have classified each of the 84 data sets into a cluster denoted by ClusterID. Thus the applicable classification algorithms of ClusterID are recommended to the data set (refer to Section 4.3 for the method of identifying the applicable classification algorithm for a cluster). And in our experiment, *three* algorithms with better performance on the cluster are recommended.

5.2 Measures to evaluate recommendation performance

To compare the performance of our proposed classification algorithm recommendation methodunder different kinds of data set characteristics, two metrics, *Recommendation Performance Ratio* and *Hit Rate*, are defined as follows.

Definition 8 *Recommendation Performance Ratio* (RPR). For a given data set D, let A_{rec} be the recommended classification algorithm, and A_{opt} be the optimal classification algorithm, i.e., the algorithm with the greatest ARR on D. RPR over data set D is defined as

$$RPR(D) = \frac{ARR^D_{A_{rec}}}{ARR^D_{A_{opt}}} \times 100\ \%, \tag{3}$$

Note that RPR only describes the performance of each individual recommendation, so we define average recommendation performance ratio (ARPR) as the mean recommendation performance over all test data sets as follows:

$$ARPR = \frac{\sum_{i=1}^{N} RPR(D_i)}{N} \times 100\ \%, \tag{4}$$

[4] The probability that we make a mistake to reject the null hypothesis, i.e., a misjudgement to say there exists significant difference but actually does not.

where RPR(D_i) is the RPR on data set D_i and N denotes the number of the test data sets.

Definition 9 *Hit* . For a given data set D, suppose that *AppSet*(D) is the set of real applicable algorithms on D, which is identified by the method introduced in *Part 1* of Section 5.1.1, and *RecSet*(D) is the set of recommended algorithms by the proposed algorithm recommendation method. Then *Hit* is defined as,

$$Hit(D) = \begin{cases} 1, if\ AppSet(D) \cap RecSet(D) \neq \phi \\ 0, otherwise \end{cases}. \quad (5)$$

Based on the definition of Hit on a data set, *Hit Rate* (HR) over all test data sets is defined as follows:

$$\text{HR} = \frac{\sum_{i=1}^{N} Hit(D_i)}{N} \times 100\ \%, \quad (6)$$

where N denotes the number of the test data sets.

5.3 Experiment results and analysis

In the experiment, for each data set, we extract the six different kinds of data set characteristics (i.e., five existing methods plus our improved one). For each kind of data set characteristics, we compute the Recommendation Performance Ratio (RPR) and Hit of the algorithms recommended by the proposed clustering leaning based recommendation method under each of the three α values 0.1 %, 1 % and 10 % in ARR. Then, we achieve the Average Recommendation Performance Ratio (ARPR) and Hit Rate (HR) over the 84 data sets, respectively.

Afterwards, we compare the recommendation results under our improved feature vector with those under the existing data set characteristics in terms of the Recommendation Performance Ratio (RPR) and Hit Rate, respectively. Meanwhile, we compare the different data set characterization methods in terms of the runtime of characteristic extraction.

5.3.1 Comparisons among different kinds of data set characteristics for algorithm recommendation

Tables 2 and 3 compare the recommendation performance under six different kinds of data set characteristics in terms of average recommendation performance ratio ARPR (see Definition 8) and Hit Rate, respectively. In these tables, the values in the parentheses are the ranks of the average recommendation performance (i.e., RPR and Hit Rate). These ranks are calculated as follows. The algorithm with the best performance getting the rank of 1, the second best rank 2 and so forth. In the case of ties (like for $\alpha = 0.1$ % with Alg_{1st} in Table 3), average ranks are assigned. Moreover, the recommendation with the best performance is marked in bold.

From these tables, we can observe that:

1. Under each of the three different settings (i.e., α) of the compromise between classification accuracy and runtime and each of the six different kinds of data set characteristics, the first recommended algorithm Alg_{1st} usually keeps the best average RPR (see Table 2) and Hit Rate (see Table 3). Alg_{2nd} is the second best and Alg_{3rd} is the third. This means that the first clustering based recommended algorithm should be in favor in practice.
2. When comparing different kinds of data set characteristics, the first recommended algorithm Alg_{1st} under $V_{Improved}$ achieves the highest RPR when α = 0.1 % and 1 % and the second highest RPR when α = 10 %.

Table 2 Comparisons of the recommendations based on different kinds of data set characteristics in terms of average RPR (%)

α	Recommendation	$V_{S\&I}$	V_{Model}	$V_{LandMark}$	$V_{Complexity}$	$V_{I\&II}$	$V_{Improved}$
$\alpha = 0.1$ %	Alg_{1st}	94.14 (3)	94.43 (2)	93.86 (5)	93.90 (4)	93.72 (6)	**97.02 (1)**
	Alg_{2nd}	93.60 (5)	91.96 (6)	93.67 (4)	94.82 (1)	94.51 (3)	94.62 (2)
	Alg_{3rd}	94.13 (2)	91.29 (6)	94.14 (1)	93.04 (3)	92.03 (5)	92.36 (4)
$\alpha = 1$ %	Alg_{1st}	95.08 (3.5)	94.78 (5)	95.08 (3.5)	94.49 (6)	95.22 (2)	**96.22 (1)**
	Alg_{2nd}	94.33 (4.5)	92.75 (6)	94.33 (4.5)	95.63 (1)	94.70 (2)	94.34 (3)
	Alg_{3rd}	93.82 (4)	93.48 (5)	93.83 (3)	93.43 (6)	94.21 (1)	94.08 (2)
$\alpha = 10$ %	Alg_{1st}	97.68 (4.5)	97.68 (4.5)	97.68 (4.5)	97.68 (4.5)	**97.79 (1)**	97.71 (2)
	Alg_{2nd}	93.36 (5)	93.27 (6)	93.96 (2.5)	93.96 (2.5)	93.63 (4)	94.17 (1)
	Alg_{3rd}	90.29 (1)	85.03 (4)	80.82 (6)	85.32 (3)	86.59 (2)	84.91 (5)

*Alg_x denotes the x-th recommended algorithm. $V_{S\&I}$ denotes the statistical and information-theory based data set characteristics, $V_{I\&II}$ is the characteristics extracted from one- and two-itemsets and $V_{Improved}$ denotes the characteristics proposed in this paper.

Table 3 Comparisons of the recommendations based on different kinds of data set characteristics in terms of Hit Rate (%)

α	Recommendation	$V_{S\&I}$	V_{Model}	$V_{LandMark}$	$V_{Complexity}$	$V_{I\&II}$	$V_{Improved}$
$\alpha = 0.1$ %	Alg_{1st}	57.14 (5.5)	63.10 (3)	58.33 (4)	57.14 (5.5)	**65.48 (1)**	64.29 (2)
	Alg_{2nd}	40.48 (6)	50.00 (3.5)	50.00 (3.5)	45.24 (5)	58.33 (2)	63.10 (1)
	Alg_{3rd}	14.29 (4.5)	13.10 (6)	20.24 (1.5)	19.05 (3)	14.29 (4.5)	20.24 (1.5)
	$Alg_{\{1st,\ 2nd,\ 3rd\}}$	75.00 (2.5)	71.43 (4.5)	63.10 (6)	71.43 (4.5)	75.00 (2.5)	**85.71 (1)**
$\alpha = 1$ %	Alg_{1st}	63.10 (3)	55.95 (6)	63.10 (3)	61.90 (5)	63.10 (3)	**71.43 (1)**
	Alg_{2nd}	51.19 (2)	41.67 (5.5)	47.62 (4)	41.67 (5.5)	53.57 (1)	48.81 (3)
	Alg_{3rd}	21.43 (5)	30.95 (2)	40.48 (1)	29.76 (3)	28.57 (4)	17.86 (6)
	$Alg_{\{1st,\ 2nd,\ 3rd\}}$	73.8 (4)	78.57 (2)	72.62 (5)	66.67 (6)	75.00 (3)	**80.95 (1)**
$\alpha = 10$ %	Alg_{1st}	89.29 (4)	89.29 (4)	89.29 4)	89.29 4)	89.29 (4)	**90.48 (1)**
	Alg_{2nd}	70.24 (5)	67.86 (6)	76.19 (2)	76.19 (2)	72.62 (4)	76.19 (2)
	Alg_{3rd}	46.43 (1)	20.24 (2.5)	10.71 (6)	19.05 (4)	20.24 (2.5)	14.29 (5.0)
	$Alg_{\{1st,\ 2nd,\ 3rd\}}$	**95.24 (2)**	94.05 (4)	**95.24 (2)**	91.67 (5.5)	91.67 (5.5)	**95.24 (2)**

*$Alg_{\{1st,\ 2nd,\ 3rd\}}$ denotes the top three recommended algorithms

When $\alpha = 10$ %, the highest RPR of Alg_{1st} corresponds to $V_{I\&II}$ (See Table 2).The first recommended algorithm Alg_{1st} under $V_{Improved}$ gets the highest Hit Rate when $\alpha = 1$ % and 10 %, and second highest Hit Rate when $\alpha = 0.1$ %. When $\alpha = 0.1$ %, the highest Hit Rate of Alg_{1st} corresponds to $V_{I\&II}$ (See Table 3). It is noted that the data set characteristics $V_{I\&II}$ and $V_{Improved}$ are both of frequencies of itemsets based. This means that this kind of data set characteristics is superior to the other ones in algorithm recommendation. Moreover, although the RPR of Alg_{1st} under $V_{Improved}$ is not the highest when $\alpha = 10$ %, the gap to that under $V_{I\&II}$ is quite small. The similar phenomenon can be found for Hit Rate when $\alpha = 0.1$ %.

3. The Hit Rate of the top three recommended algorithms $Alg_{1st,\ 2nd,\ 3rd}$ achieves the highest value under $V_{Improved}$ for all the three different α settings. Moreover, the highest Hit Rate is evidently better than those under other five kinds of data set characteristics when $\alpha = 0.1$ % and 1 %. According to the definition of Hit Rate in Section 5.2, the higher the Hit Rate of a set of recommended algorithms, the more possible that the recommended algorithms include a real applicable algorithm. Thus, from Table 3, we can conclude that if the user needs to narrow down the choice of the candidate algorithms, the improved data set characteristics should be in favor.

5.3.2 Comparisons among the extraction time of different kinds of data set characteristics

For algorithm recommendation, a set of good data set characteristics should not only show better performance in constructing the recommendation model, but also have the advantages of simple operation and easy calculation. The data set characterization methods which spend long time to calculate the characteristics are not practical. Thus, the runtime of a data set characterization method is an important metric to evaluate the effectiveness of the method.

Table 4 shows the runtime of each data set characterization method over the 84 data sets. For each data set, the extraction time, marked in bold, corresponds to the fastest data set characterization method. From this table, we can observe that for most data sets, the improved data set characterization method is the fastest (only *127.33ms* on average). Its average extraction runtime is only 2.86 % of that of $V_{S\&I}$, 37.69 % of that of V_{Model}, 0.56 % of that of $V_{LandMark}$, 0.05 % of that of $V_{Complexity}$, and 9.01 % of that of $V_{I\&II}$.

According to the extraction process of the improved data set characteristics $V_{Improved}$ introduced in Section 3, there are two steps, collecting the frequencies of one- and two-itemsets and mapping these frequencies into some specific intervals. The former can be done by scanning the data set one pass without any extra operation, and the later can be finished by scanning these frequencies in one pass without any extra operation as well. Comparing to the other extraction methods, they usually need to do some further computation to collect the data set characteristics. Such as, for problem complexity based $V_{Complexity}$, it needs to compute the distances between all instances and find the classification boundary; for landmarking based $V_{LandMark}$, it needs to train multi-classifiers; and for $V_{I\&II}$, it needs to sort the frequencies of the itemsets.

Moreover, according to the analysis in Section 5.3.1, the recommendation model constructed based on the improved

Table 4 Comparisons of different kinds of data set characteristics in terms of extraction time (ms)

Data ID	$V_{S\&I}$	V_{Model}	$V_{LandMark}$	$V_{Complexity}$	$V_{I\&II}$	$V_{Improved}$	Data ID	$V_{S\&I}$	V_{Model}	$V_{LandMark}$	$V_{Complexity}$	$V_{I\&II}$	$V_{Improved}$
1	891.59	417.69	2195.54	89271.79	605.13	**146.71**	43	269.05	11.92	138.91	1008.22	25.79	**2.59**
2	1437.19	87.76	1359.96	49927.44	377.14	**42.15**	44	231.52	**7.60**	26.40	4379.35	57.33	10.77
3	4873.54	1297.54	6170.69	522066.85	2491.42	**346.48**	45	54.99	**2.53**	63.89	6247.15	27.80	2.70
4	506.54	**19.31**	305.25	193694.96	345.94	36.31	46	2528.55	2262.92	25751.09	842622.74	4877.09	**968.79**
5	299.72	22.38	566.78	5931.78	90.48	**12.66**	47	2012.83	1924.47	21807.53	697592.38	3812.06	**778.45**
6	70.36	**15.32**	158.12	18108.44	137.78	16.57	48	424.09	120.22	2411.52	69735.71	213.73	**39.82**
7	43.99	17.29	183.51	4830.68	21.62	**4.84**	49	1340.08	1637.66	16144.00	507589.38	2979.70	**554.68**
8	71.52	3.08	79.84	1679.42	41.73	**1.60**	50	155.30	**2.38**	76.07	3525.21	265.26	32.99
9	89.33	18.13	426.11	3858.28	54.24	**9.74**	51	167.34	6.51	165.69	2248.74	18.67	**1.26**
10	1319.42	11.05	1191.32	26865.31	66.21	**6.54**	52	182.24	4.77	179.03	2469.28	20.48	**1.34**
11	88.87	4.16	95.43	2053.53	27.41	**1.70**	53	182.67	1.29	166.35	2164.82	19.50	**1.23**
12	1139.53	74.22	1410.16	24656.30	92.84	**9.35**	54	84696.02	**59.48**	86214.24	128599.19	3403.47	149.49
13	158.38	14.33	263.84	3789.79	112.28	**12.28**	55	13730.16	70.13	82224.30	329260.14	686.68	**55.32**
14	53480.69	1103.22	420817.72	950090.74	7486.84	**759.97**	56	6153.44	2532.70	145864.20	2758275.11	13007.75	**999.54**
15	665.66	**22.96**	621.42	6844.95	86.74	24.31	57	2653.72	571.16	24772.66	470830.92	1124.91	**131.61**
16	1044.46	64.10	1442.82	9435.52	151.90	**26.37**	58	6610.82	1711.78	159179.37	1481810.39	10272.57	**500.82**
17	528.01	8.04	465.31	6069.49	87.69	**6.89**	59	590.31	3.96	274.47	3598.71	17.70	**1.67**
18	584.30	**13.39**	930.89	6726.68	9178.04	64.72	60	34.51	0.72	19.83	1502.51	5.85	**0.46**
19	384.54	**7.60**	325.51	39941.03	256.90	27.31	61	211.69	12.73	173.77	69306.29	45.70	**3.22**
20	78.18	24.60	476.14	3113.13	28.61	**12.42**	62	751.81	218.74	8638.16	166176.48	2236.75	**130.11**
21	29.91	13.91	122.60	9486.30	25.46	**4.07**	63	7.60	**0.15**	7.20	124.65	2.38	0.70
22	121.08	**12.80**	124.49	22344.88	129.39	14.07	64	17447.04	218.89	24477.58	70915.54	607.94	**118.07**
23	1090.29	27.07	974.99	9167.16	139.56	**11.63**	65	122.39	**1.38**	108.67	2264.95	22.07	1.89
24	20.74	12.43	77.22	6405.48	20.72	**4.87**	66	910.94	**4.07**	878.45	20442.58	83.48	5.32
25	20.81	4.72	51.07	470.96	22.61	**1.70**	67	162.07	48.51	351.88	2549.76	49.34	**14.92**

Table 4 (continued)

Data ID	$V_{S\&I}$	V_{Model}	$V_{LandMark}$	$V_{Complexity}$	$V_{I\&II}$	$V_{Improved}$	Data ID	$V_{S\&I}$	V_{Model}	$V_{LandMark}$	$V_{Complexity}$	$V_{I\&II}$	$V_{Improved}$
26	13.86	0.61	21.72	1158.85	19.53	**0.38**	68	1069.01	**25.04**	1010.48	229020.48	473.94	26.08
27	70.38	8.18	142.67	9626.78	26.64	**4.07**	69	4326.25	2075.00	87449.98	205352.18	3624.72	**698.56**
28	57.11	9.80	119.85	7279.47	33.31	**6.24**	70	443.75	6.98	126.18	3431.94	25.52	**4.16**
29	34.40	9.64	120.06	1468.62	26.13	**4.47**	71	1830.02	1199.78	5149.11	1249689.96	9611.10	**953.53**
30	62.23	**4.39**	60.92	1414.10	27.38	7.72	72	35955.11	**183.77**	33775.22	345770.28	7651.33	578.38
31	203.94	8.66	203.21	4144.74	222.26	**8.03**	73	90.22	**1.19**	46.86	6260.26	112.15	9.68
32	13261.66	**49.25**	11547.75	49878.32	415.04	52.05	74	22.54	4.20	28.93	836.01	**3.15**	3.89
33	17687.14	**70.18**	23562.41	210757.78	727.30	111.71	75	548.18	8.65	589.91	5580.46	36.98	**3.37**
34	103.48	68.98	427.46	6196.74	331.32	**49.99**	76	62.38	**0.21**	12.96	316.50	9.17	1.33
35	11.31	**1.26**	37.83	1201.97	13.63	1.43	77	146.96	**0.79**	158.88	1691.10	13.86	0.86
36	29892.98	90.38	25057.66	407731.05	2356.38	**53.13**	78	150.80	95.89	1224.73	33982.95	181.18	**29.86**
37	15586.33	69.81	12963.91	277097.03	1314.40	**28.72**	79	214.79	3.78	223.28	4065.88	40.98	**3.34**
38	738.47	**266.14**	2159.29	106112.81	2216.06	276.89	80	315.97	130.06	1234.61	80999.47	203.31	**45.63**
39	22064.49	**67.68**	19370.22	79119.99	613.65	109.68	81	3765.25	1920.20	82589.62	199597.38	2289.64	**339.86**
40	40.40	**1.17**	17.53	488.39	30.22	1.65	82	269.12	7.48	87.28	4115.63	19.37	**4.96**
41	4901.82	32.12	4361.71	133904.06	101.43	**9.46**	83	1288.25	9.77	1050.86	63014.24	64.45	**3.84**
42	8230.49	7196.47	553949.50	6606260.33	19933.11	**1146.99**	84	53.00	**1.00**	32.51	7064.35	10.79	2.97
Average	4454.55(4.17)	337.88(1.71)	22736.51(4.46)	238175.01(5.99)	1413.60(3.35)	**127.33(1.32)**							

data set characteristics $V_{Improved}$ shows better performance in terms of RPR and Hit Rate. Thus, to sum up, both the recommendations and extraction time show that our improved data set characteristics $V_{Improved}$ is very effective.

6 Conclusion

In this paper, we propose an improved data set characterization method, which can be directly applied on ordinary data sets and is more robust and more efficient.

The first computes the frequencies of the one-item sets and two-item sets in a data set; then these frequencies are unified into a fixed length of metrics as the data set characteristics. When calculating the frequencies of itemsets, the conjunction function is employed instead of the parity function. This means the improved method can be directly applied to an ordinary data set. When unifying the frequencies of itemsets, we map these frequencies into specific intervals and define the ratios of the number of values in these intervals as the metrics, rather than extract the fractiles of these frequencies. This makes the improved method is more robust and more effective.

In order to validate the improved data set characteristics extraction method for algorithm recommendation, a clustering based algorithm recommendation method is proposed. In this method, at first, the data set features are extracted, and the data sets are clustered and their applicable classification algorithms are identified. Then the relationships between data set clusters and classification algorithms' performance are built up, upon which classification algorithms can be recommended to a new classification problem. In particular, a new data set is classified into a certain cluster after working out its feature, and the applicable classification algorithms of that cluster are recommended. Finally, 84 public UCI data sets, 17 different types of classifiers and five different kinds of data set characteristics have been used in the experiment. The experimental results show that the improved data set characterization method is effective.

Acknowledgments This work is supported by China Postdoctoral Science Foundation under grant NO. 2014M562417 and the National Natural Science Foundation of China under grant 61402355.

Appendix A: The five different kinds of meta-features

1. Statistical and Information-Theory Based Measures (See Table 5)
2. Model Structure Based Measures (See Table 6)

Table 5 Statistical and Information-Theory Based Measures

Measures	Definitions
Ins.Num	Number of instances
Attr.Num	Number of Attributes
Target.Num	Number of target concept values
Target.Min	Proportion of minority target
Target.Max	Proportion of majority target
Pro.Bin	Proportion of binary attributes
Pro.Nom	Proportion of nominal attributes
Pro.Num	Proportion of numeric attributes
Pro.MissIns	Proportion of instances with missing values
Pro.MissValues	Proportion of missing values
Mean.Geo	Geometric mean
Mean.Harm	Harmonic mean
Mean.Trim	Trim mean excluding the highest and lowest 5 %
Mad	Mean absolute deviation
Var	Variance
Std	Standard deviation
Prcitile	Percentile 75 %
Int.Range	Interquartile range
Prop.AttrWithOutlier	Proportion of numerical attributes with outliers over all numerical attributes
Skewness	Skewness of data based on numerical attributes
Kurtosis	Kurtosis of data based on numerical attributes
Max.eig	Maximum eigenvalue

Table 5 (continued)

Measures	Definitions
Min.eig	Minimum eigenvalue
Can.corr	Canonical correlation
Grav.cent	Center of gravity
MeanAbsCoef	Mean absolute coefficient of attribute pairs
$H(C)$	Entropy of classes
$\bar{H}(X)$	Mean entropy of nominal attributes
$\bar{M}(C, X)$	Mean mutual information of classes and attributes based on nominal attributes
En.attr	Equivalent number of attributes $H(C)/\bar{M}(C, X)$
Ns.ratio	Noise-signal ratio $\bar{H}(X)/\bar{M}(C, X) - 1$

Table 6 Model Structure Based Measures

Measures	Definitions
Tree.Height	Height of tree (also referred as to number of levels in tree)
Tree.Wdith	Width of tree
Node.Num	Number of nodes in tree
Leaf.Num	Number of leaves in tree
Level.Max	Maximum number of nodes at one level
Level.Mean	Mean of the number of nodes on levels
Level.Dev	Standard deviation of the number of nodes on levels
Branch.Long	Length of the longest branch
Branch.Short	Length of the shortest branch
Branch.Mean	Mean of the branch lengths
Branch.Dev	Standard deviation of the branch lengths
Attr.Min	Minimum occurrence of attributes
Attr.Max	Maximum occurrence of attributes
Attr.Mean	Mean of the number of occurrences of attributes
Attr.Dev	Standard deviation of the number of occurrences of attributes

Table 7 Problem Complexity Based Measures

Measures	Definitions
Bound.Len	Length of class boundary
Adherence.Prop	Proportion of retained adherence subsets
Intra/Inter.Ratio	Ratio of average intra/interclass nearest neighbors
NN.Nonlinerity	Nonlinearity of Nearest Neighbors classifier
Linear.Nonlinerity	Nonlinearity of linear classifier
Fisher.Ratio	Maximum Fisher's discriminant ratio
Ins/Attr	Training set size relative to feature space dimensionality

3. Land Marking Based Measures Following the suggestions in [7, 38], the following six classifiers are selected as the landmark learners: i) Naive Bayes, ii) 1-NN (Nearest Neighbor), iii) Elite 1-NN, iv) a decision node learner, v) a random chosen node learner and vi) the worst node learner. Where the last three learners can be achieved based on the well-known learning algorithm C4.5.
4. Problem Complexity Based Measures (See Table 7)
5. Structural Information Based Measures First, the two feature vectors, one-item feature vector and two-item feature vector, are extracted from the given problem. These two vectors consists of the frequencies of one-item sets and two-item sets, respectively. Afterward, the *minimum*, *1/8 quantile*, *2/8 quantile*, *3/8 quantile*, *4/8 quantile*, *5/8 quantile*, *6/8 quantile*, *7/8 quantile* and *maximum* are computed for these two vectors and form the final set of data set characteristics.

References

1. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol 22. ACM, pp 207–216
2. Aha DW (1992) Generalizing from case studies: A case study. In: Proceedings of the Ninth International Conference on Machine Learning. Citeseer, pp 1–10
3. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Mach Learn 6(1):37–66
4. Ali S, Smith KA (2006) On learning algorithm selection for classification. Appl Soft Comput 6(2):119–138
5. Asuncion A NDJ. UCI machine learning repository (2007). http://www.ics.uci.edu/∼mlearn/{MLR}epository.html
6. Bensusan H (1998) God doesn't always shave with occam's razor - learning when and how to prune. In: Proceedigs of the 10th European Conference on Machine Learning. Springer, pp 119–124
7. Bensusan H, Giraud-Carrier C (2000) Casa batlo is in passeig de gracia or landmarking the expertise space. In: Proceedings of the ECML'2000 workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination, pp 29–47
8. Brazdil P, Gama J, Henery B (1994) Characterizing the applicability of classification algorithms using meta-level learning. In: Proceedings of the European conference on Machine Learning. Springer, pp 83–102
9. Brazdil P, Soares C (2000) A comparison of ranking methods for classification algorithm selection. In: 11th European Conference on Machine Learning. Springer, pp 63–75
10. Brazdil PB, Soares C, Da Costa JP (2003) Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Mach Learn 50(3):251–277
11. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
12. Breiman L (2001) Random Forests. Mach Learn 45:5–32
13. Brodley CE (1993) Addressing the selective superiority problem: Automatic algorithm/model class selection. In: Proceedings of the Tenth International Conference on Machine Learning. Citeseer, pp 17–24
14. Castiello C, Castellano G, Fanelli A (2005) Meta-data: Characterization of input features for meta-learning. Modeling Decisions for Artificial Intelligence pp. 457–468
15. Cleary JG, Trigg LEK* (1995) An Instance-based Learner Using and Entropic Distance Measure. In: International Conference on Machine Learning, pp 108–114
16. Cohen WW (1995) Fast effective rule induction. In: Proceedings of the International Conference on Machine Learning, pp 115–123
17. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. J Royal Stat Soc: 1–38
18. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
19. Duin RPW, Pekalska E, Tax DMJ (2004) The characterization of classification problems by classifier disagreements. In: Proceedings of the 17th International Conference on Pattern Recognition, vol 1. IEEE, pp 140–143
20. Fisher D, Xu L, Zard N (1992) Ordering effects in clustering. In: Proceedings of the Ninth International Conference on Machine Learning
21. Frank E, Witten IH (1998) Generating accurate rule sets without global optimization. In: Proceedings of the 15th International Conference on Machine Learning. Citeseer
22. Freund Y, Schapire R (1996) Experiments with a new boosting algorithm. In: Proceeding of the Thirteenth International Conference on Machine Learning. Citeseer, pp 148–156
23. Gama J, Brazdil P (1995) Characterization of classification algorithms. Progress in Artificial Intelligence pp. 189–200
24. Henery RJ (1994) Methods for comparison. Ellis Horwood, Upper Saddle River, NJ, USA, pp 107–124. http://dl.acm.org/citation.cfm?id=212782.212789
25. Hilario M, Kalousis A (2001) Fusion of meta-knowledge and meta-data for case-based model selection. Principles of Data Mining and Knowledge Discovery pp. 180–191
26. Ho TK (2000) Complexity of classification problems and comparative advantages of combined classifiers. Multiple Classifier Systems pp. 97–106
27. Ho TK, Basu M (2002) Complexity measures of supervised classification problems. IEEE Trans Pattern Anal Mach Intell 24(3):289–300
28. John GH, Langley P (1995) Estimating continuous distributions in bayesian classifiers. In: Proceedings of the eleventh conference on uncertainty in artificial intelligence, vol 1. Citeseer, pp 338–345
29. Kalousis A (2002) Algorithm selection via meta-learning. Ph.D. thesis, University of Geneve
30. Kalousis A, Gama J, Hilario M (2004) On data and algorithms: Understanding inductive performance. Mach Learn 54(3): 275–312
31. Kalousis A, Hilario M (2000) Model selection via meta-learning: a comparative study. In: Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence. IEEE, pp 406–413
32. Kalousis A, Theoharis T (1999) NOEMON: Design, implementation and performance results of an intelligent assistant for classifier selection. Intell Data Anal 3(5):319–337
33. King RD, Feng C, Sutherland A (1995) Statlog: comparison of classification algorithms on large real-world problems. Appl Artif Intell Int J 9(3):289–333
34. Lindner G, Studer R (1999) AST: Support for algorithm selection with a CBR approach. Principles of Data Mining and Knowledge Discovery pp. 418–423
35. Michie D, Spiegelhalter DJ, Taylor CC (1994) Machine learning, neural and statistical classification. Citeseer

36. Michie D, Spiegelhalter DJ, Taylor CC (1994) Machine learning. neural and statistical classification
37. Peng Y, Flach P, Soares C, Brazdil P (2002) Improved dataset characterisation for meta-learning. In: Discovery Science. Springer, pp 193–208
38. Pfahringer B, Bensusan H, Giraud-Carrier C (2000) Meta-learning by landmarking various learning algorithms. Morgan Kaufmann, pp 743–750
39. Pizarro J, Guerrero E, Galindo PL (2002) Multiple comparison procedures applied to model selection. Neurocomputing 48: 155–173
40. Platt J (1998) Machines using sequential minimal optimization
41. Quinlan JR (1994) Comparing connectionist and symbolic learning methods. In: Computational Learning Theory and Natural Learning Systems: Constraints and Prospects. Citeseer
42. Smith KA, Woo F, Ciesielski V, Ibrahim R (2001) Modelling the relationship between problem characteristics and data mining algorithm performance using neural networks. Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming. Data Mining, and Complex Systems pp. 357–362
43. Smith KA, Woo F, Ciesielski V, Ibrahim R (2002) Matching data mining algorithm suitability to data characteristics using a self-organising map. Hybrid Information Systems pp. 169–180
44. Smith-Miles KA (2008) Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Comput Surv 41(1):1–25
45. Sohn SY (1999) Meta analysis of classification algorithms for pattern recognition. IEEE Trans Pattern Anal Mach Intell 21(11):1137–1144
46. Song Q, Wang G, Wang C (2012) Automatic recommendation of classification algorithms based on data set characteristics. Pattern Recognition
47. Tatti N (2007) Distances between data sets based on summary statistics. J Mach Learn Res 8:131–154
48. Webb GI (2000) Multiboosting: A technique for combining boosting and wagging. Mach Learn 40(2):159–196
49. Wolpert DH (2001) The supervised learning no-free-lunch theorems. In: Proceedings of 6th Online World Conference on Soft Computing in Industrial Applications. Citeseer, pp 25–42