Ensemble Learning Based Classification Algorithm Recommendation

GUANGTAO WANG, QINBAO SONG and XIAOYAN ZHU, Xi'an Jiaotong University, China

Recommending appropriate algorithms to a classification problem is one of the most challenging issues in the field of data mining. The existing algorithm recommendation models are generally constructed on only one kind of meta-features by single learners. Considering that i) ensemble learners usually show better performance and ii) different kinds of meta-features characterize the classification problems in different viewpoints independently, and further the models constructed with different sets of meta-features will be complementary with each other and applicable for ensemble. This paper proposes an ensemble learning based algorithm recommendation method. To evaluate the proposed recommendation method, extensive experiments with 13 well-known candidate classification algorithms and five different kinds of meta-features are conducted on 1090 benchmark classification problems. The results show the effectiveness of the proposed ensemble learning based recommendation method.

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Classification algorithm recommendation, Ensemble learning, Multi-label learning

1. INTRODUCTION

As one of the most important and widely-used techniques, classification has been a research hotspot for decades in the field of data mining. A number of classification algorithms have been developed. And various improved or new algorithms are constantly proposed. However, no matter from experimental studies [Brazdil et al. 2003; Bensusan 1998; Ali and Smith 2006b; Smith-Miles 2008; Yang and Jiu 2006; Song et al. 2012; Prudêncio et al. 2011; Ali et al. 2018; Khan et al. 2020] or theoretical analysis [Wolpert 2001], there is no specific classification algorithm being appropriate for all classification problems. The appropriate classification algorithms vary with different classification problems. Therefore, for a given classification problem, users are usually puzzled in facing so many candidate algorithms. In this context, it is necessary to explore an effective classification algorithm automatic recommendation method to assist users picking up the appropriate algorithms for the given problem.

Algorithm recommendation aims at exploring the relationship between the characteristics of the classification problems and the appropriate algorithms on them, and further utilizes this relationship to assist algorithm selection for a new classification problem. To facilitate presentation and understanding algorithm recommendation, we first give some related notations as follows.

- $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ denotes a collection of $N$ historical classification problems;
- $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ denotes a collection of $k$ candidate classification algorithms;
- $F: \mathcal{P} \rightarrow R^m$, $F$ is a function which can extract $m$ characteristics from each classification problem $p_i \in \mathcal{P}$ as the meta-features of $p_i$, where $R^m$ presents the space of meta-features;
- $X = \{X_1, X_2, \ldots, X_N\} \subseteq R^m$ denotes the meta-features of the $N$ classification problems in $\mathcal{P}$ collected by $F$, where $X_i (1 \leq i \leq N)$ is the meta-features of $p_i$, i.e., $X_i = F(p_i)$;
- $Y = \{Y_1, Y_2, \ldots, Y_N\}$ denotes the meta-target of the $N$ classification problems in $\mathcal{P}$, where $Y_i$ represents the appropriate algorithms of $p_i \in \mathcal{P}$ identified from $\mathcal{A}$.

With these notations, algorithm recommendation can be formally defined as a process consisting of two steps: i) finding a function $\phi: X \rightarrow Y$ which maps the characteristics of the classification problems $X$ to the appropriate classification algorithms on...
them \( Y \); ii) for a new problem \( p_{\text{new}} \), recommending the algorithms for it according to 
\( \phi(F(p_{\text{new}})) \).

Thus, the key point to solve the algorithm recommendation problem is constructing an accurate function \( \phi \) (i.e., algorithm recommendation model) exposing the relationship between input \( X \) and output \( Y \). In the field of algorithm recommendation, different representations of the meta-target \( Y \) will lead to different methods to construct the function \( \phi \). There have been different representations of the meta-target for a classification problem in different views. Such as single-label, algorithm ranking and multi-label based meta-targets [Wang et al. 2014]. Where single-label based meta-target assumes that only the algorithm with the best performance is appropriate for a given classification problem, and algorithm ranking based meta-target assumes that each candidate algorithm might be an appropriate one and supplies a rank list of the candidate algorithms according to their performance. Considering that there are usually multiple classification algorithms being statistically equivalent on a given classification problem in terms of a given performance metric [Wang et al. 2014], neither the single-label nor the algorithm ranking based meta-target can fit this situation. Multi-label based meta-target is a natural and more reasonable representation of the appropriate algorithms on a classification problem. Moreover, the extensive experimental results in [Wang et al. 2014] have demonstrated that the recommendation models constructed on the multi-label based meta-data have better performance. In this paper, we will follow the multi-label based meta-target, and attempt to find the function \( \phi \) by the multi-label learning algorithms.

There have been various supervised learning methods used to explore the function \( \phi \) [Peng et al. 2002; Pfahringer et al. 2000; Kalousis 2002; Brazdil et al. 2003; Ali and Smith 2006b; Prudêncio et al. 2011; Song et al. 2012; Wang et al. 2014], including single-label learning, \( k \)-NN (nearest neighbor) and multi-label learning based methods, etc. However, all these methods usually employ a single learner to find \( \phi \). The ensemble learner, which integrates a set of single learners in a special way, has been experimentally proved that it is usually of higher accuracy than the single learners [Dietterich 1997; Dietterich 2002]. As far as we know, there is still no such an ensemble learning based method used for constructing any algorithm recommendation model. Therefore, the paper attempts to explore the function \( \phi \) as the recommendation model by the multi-label ensemble learning method.

Meanwhile, the existing methods usually learn the function \( \phi \) on only a single kind of meta-features. There have been several different kinds of meta-features are extracted from different viewpoints of a classification problem [Brazdil et al. 2003; Gama and Brazdil 2000; Engels and Theusinger 1998; Bensusan 1998; Peng et al. 2002; Pfahringer et al. 2000; Bensusan and Giraud-Carrier 2000; Ho and Basu 2002; Song et al. 2012]. These different kinds of meta-features are complementary with each other in describing the classification problem. Combination of these different kinds of meta-features will get a better understanding of the classification problem. Different combinations will produce multiple different sets of meta-data. So it is possible for us to construct multiple complementary single recommendation models on these meta-data. Furthermore, by integrating these single recommendation models together, we can get an ensemble recommendation model. This will be quite different with the existing algorithm recommendation methods. Moreover, the extensive experimental results indicate the effectiveness of the proposed ensemble learning based recommendation method.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 introduces the preliminary study of ensemble learning based algorithm recommendation. Section 4 presents the ensemble learning based algorithm recommen-
2. RELATED WORK

Algorithm recommendation has been formally described as a meta-level learning problem since 70s last century [Rice 1976; Ali et al. 2018; Khan et al. 2020]. Afterwards, this idea is widely used in the field of algorithm recommendation [Smith-Miles 2008]. The input of this meta-level learning problem is the characteristics of a classification problem (i.e., “meta-features”) and the output is the appropriate candidate algorithms on the classification problem (i.e., “meta-target”). The meta-features and meta-targets constitute the meta-data on which we construct the algorithm recommendation models by specific data mining techniques. From the perspective of data mining, the construction of an accurate recommendation model should depend on effectiveness of the following two aspects.

(1) Meta-data preparation

The recommendation model is induced from the meta-data. The qualities of the meta-features and meta-target will be very critical for model construction. This is because of the famous principle in the field of data mining: “Garbage In, Garbage Out” [Lee et al. 1999]. Thus, improving the quality of meta-data has been a main direction in algorithm recommendation. Most of the research works concern on high-quality meta-data collection.

(2) Model construction

The recommendation model is usually built with a given learning technique (e.g., classification) on the meta-data. Different learning techniques result in different recommendation models. In order to guarantee the generalization ability of the recommendation model, elaborate design of the learning schedule is another important research direction in algorithm recommendation.

Researchers have done a lot of works on constructing more effective recommendation models [Brazdil et al. 2003; Gama and Brazdil 2000; Engels and Theusinger 1998; Bensusan 1998; Peng et al. 2002; Pfahringer et al. 2000; Bensusan and Giraud-Carrier 2000; Jain et al. 2000; Duin et al. 2004; Bernado-Mansilla and Ho 2005; Ho and Basu 2002; Elizondo et al. 2009; Ho 2000; Song et al. 2012]. These works mainly focus on the first aspect, including i) extracting a set of high-quality meta-features to characterize a classification problem and ii) developing an effective form of meta-target to represent the appropriate algorithms on a classification problem.

For meta-feature collection, different kinds of data set characterization methods have been proposed from different perspectives of a classification problem, including i) statistic and information-theory based method, which extracts the statistic (e.g., mean value, standard deviation etc.) and information theory based measures (e.g., entropy, signal to noise ratio, etc.) as the meta-features [Brazdil et al. 2003; Gama and Brazdil 2000; Engels and Theusinger 1998]; 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 [Bensusan 1998; Peng et al. 2002]; 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 [Pfahringer et al. 2000; Bensusan and Giraud-Carrier 2000; Jain et al. 2000; Duin et al. 2004]; 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 [Ho and Basu 2002; Elizondo et al. 2009]; and v) structural information based method, which uses structural information based feature vectors to characterize the classification problems [Song et al. 2012]. All these meta-features have been
employed to construct algorithm recommendation models and given us some useful
guidelines for picking up appropriate algorithms.

Nevertheless, these recommendation models are usually constructed on a single
kind of meta-features by a single learner. As we know, an ensemble learner combining
a set of single learners in a specific way (e.g., weighted or unweighted voting) is usu-
ally much more accurate than the single learners [Dietterich 2002; Dietterich 2000a;
Džeroski and Ženko 2004]. But there are few researches of constructing recommenda-
tion models by the ensemble learners.

For meta-target preparation, the expression form of the meta-target has a great in-
fluence on the single learners used to build a recommendation model. Two most widely
used forms are single-label [Ali and Smith 2006a; Ali and Smith 2006b; Kalousis 2002;
Kalousis et al. 2004] and algorithm ranking [Brazdil et al. 2003; Brazdil and Soares
2000; Song et al. 2012]. The former assumes that there is a single optimal algorithm
for a given classification problem and forms a single-labeled meta-data. Furthermore,
the single-label learners are employed to build the recommendation models. The later
weighs all the candidate algorithms according to their performance on a classifica-
tion problem and further gets a rank list of these algorithms as the meta-target. Con-
strained by the ranking structure, the recommendation models on algorithm ranking
based meta-data are usually constructed by $k$ nearest neighbor method or its variants.
Recently, Wang et al. [Wang et al. 2014] proposes a new and natural multi-label form
to describe the meta-target due to the fact that there would be multiple algorithms
being appropriate for a given classification problem in practice, and further constructs
the recommendation model by multi-label learning methods. The experimental results
show that the multi-label based meta-target is more effective than the single-label and
ranking based ones. But the multi-label learning methods used in [Wang et al. 2014]
are still single learner based. In this paper, we will still employ the multi-label based
meta-target. Different from the existing recommendation model construction meth-
ods, we attempt to construct the recommendation model by the multi-label ensemble
learners.

There have been some research works for multi-label ensemble learning
[G. Tsoumakas 2007; Naisierding et al. 2010; Rokach et al. 2014; Read et al. 2008;
Shi et al. 2011; Zhang and Zhou 2014]. These works can be generally grouped into two
categories: data transformation [G. Tsoumakas 2007; Naisierding et al. 2010; Rokach
et al. 2014] and ensemble adaptation [Shi et al. 2011; Zhu et al. 2020]. For the data
transformation methods, a multi-label problem is usually divided into multiple single-
label problems, then each base model of the ensemble is trained on one of the single-
label problems. The representative method is RAndom k-labELsets (RAKEL) algo-
rithm [G. Tsoumakas 2007], which transforms a multi-label learning problem into an
ensemble of a set of multi-class single-label learning problems. The ensemble adaptation
methods extend the single-label ensemble learning method in order to handle
the multi-label problem directly. For example, Shi et al. [Shi et al. 2011] proposed two
multi-label based criteria to evaluate the accuracy and diversity of multi-label learn-
ing models and then constructed an ensemble by optimizing these two criteria with an
Evolutionary Algorithm (EA). Since different base learners will lead to different ge-
netic representations and operations, and not all the multi-label base learners can be
optimized by EA, the main weakness of their method is that it is mostly tailored for a
specific multi-label learners (e.g., BP-MLL [Zhang 2006] and ML-RBF [Zhang 2009]),
and thus lack generality.

In this paper, we will apply the data transformation method to build the multi-
label learning based algorithm recommendation models since the data transformation
method is easy to be implemented and not limited by a specific learner. That is, at
first, the multi-labeled meta-data is transformed into multiple single-labeled meta-data in both meta-feature and meta-target spaces; then the recommendation models constructed on all the transformed single-labeled meta-data are viewed as the based models of the ensemble.

3. PRELIMINARY STUDY

In this section, we first discuss the rationality and feasibility of constructing algorithm recommendation model by ensemble learning. Then, as the accurate and diverse base models play a critical role in constructing an ensemble recommendation model, we give the definitions of accurate and diverse base learning models from the view of ensemble learning. Finally, based on these definitions, we propose a method to pick up a set of accurate and diverse base recommendation models for algorithm recommendation.

3.1. Rationality and Feasibility

Ensemble learner usually shows better performance since that it can partly overcome the following problems encountering by the single learner: the statistical problem and the representational problem [Dietterich 2002]. These problems will also arise when constructing the recommendation models by the single learners.

For example, most of the published studies in algorithm recommendation usually employed only dozens of classification problems in $P$ to explore the function $\phi$ [Brazdil et al. 2003, King et al. 1995, Song et al. 2012, Brazdil et al. 1994, Ali and Smith 2006a, Ali and Smith 2006b, Prudencio et al. 2011, Peng et al. 2002, Kalousis et al. 2004, Kalousis 2002, Brodley 1993, Pfahringer et al. 2000]. In some studies, comparing to the size of input $X$ (i.e., the number of the classification problems in $P$), its dimension (i.e., the number of meta-features) is usually relatively large. Such as, only 12 classification problems but up to 19 meta-features used in [King et al. 1995], and 32 classification problems and 8 meta-features used in [Prudencio et al. 2011], etc. With limited number of classification problems in $P$, the greater the number of meta-features used, the more difficult to find the true function $\phi$. This is identified as the statistical problem encountered by the single learners. And this problem is usually very significant and serious in the field of algorithm recommendation. In order to overcome this issue and further get recommendation model with better generality ability, ensemble learning will be a good choice.

Moreover, in practice, the performance of a classification algorithm on a given classification problem is related to many factors (or meta-features) of the problem, and different factors play different roles [Wang et al. 2014]. This results in that the true function $\phi$ might be quite complex. To appropriate a complex function $\phi$, the single learners usually might be limited by its representational ability. The ensemble learner can enrich the representational ability of the single learners by combining them in a special way, and further relieve the representational problem. Such as, the Fisher’s linear discriminant algorithm only searches in the linear space. However, ensemble of multiple linear learners can approach a non-linear function. Consequently, in the circumstances without any prior knowledge of the form of $\phi$, ensemble learning will be a sensible candidate.

The discussions above show us that the rationality of constructing the accurate recommendation models by ensemble learning. Next, based on the researches on ensemble learning [Dietterich 2002, Dietterich 2000a, Hansen and Salamon 1990], we first give a corollary to guide the construction of an accurate ensemble learning model, and then discuss the feasibility of constructing an ensemble learning based recommendation model.
COROLLARY 3.1. The efficient and necessary condition to construct an accurate ensemble learning model is that the base (or single) learning models are accurate and diverse/independent with each other.

According to Corollary 3.1, the key point to construct an accurate ensemble recommendation model is to find a set of accurate and diverse base recommendation models. In this paper, we try to build different base recommendation models with respect to different kinds of meta-features, and then assemble these base recommendation models together to form the ensemble recommendation model. Considering Corollary 3.1, the feasibility of this idea can be derived from the following aspects.

1. Accurate base model construction

There have been many different recommendation models constructed with respect to different kinds of meta-features and single learning methods [Brazdil et al. 2003; Gama and Brazdil 2000; Engels and Theusinger 1998; Bensusan 1998; Peng et al. 2002; Pfahringer et al. 2000; Bensusan and Giraud-Carrier 2000; Jain et al. 2000; Duin et al. 2004; Ho and Basu 2002; Elizondo et al. 2009; Song et al. 2012]. And we can view these models as the base recommendation models of the ensemble. Ensemble learning model has one quite good property that it does not require all the base learning models with pretty good accuracy. That is, it is usually achieved by combining a set of weak base learning models [Dietterich 2002; Dietterich 2000a]. Although there exist some differences among the existing recommendation models, all these models can effectively narrow down the choices of the candidate classification algorithms, and have reasonable recommendation accuracy. This provides us the evidence that it is reasonable to construct a set of accurate base recommendation models of ensemble.

![Fig. 1: Correlation coefficients among different kinds of meta-features](image)

---

1 In the field of ensemble learning, the independent base learners is generally called diverse learners.
Diverse base model construction

There have been five different kinds of meta-features in the field of algorithm recommendation (See details in Appendix 6). These meta-features are extracted from different viewpoints of a classification problem independently. So it is reasonable to assume that, different kinds of meta-features are independent with each other. Fig. 1 gives the correlation coefficients among the five kinds of meta-features extracted from 1090 benchmark classification problems. From this figure, we can find that the correlation among different kinds of meta-features is usually quite low. This gives us an experimental evidence that different kinds of meta-features are independent with each other. Furthermore, it is more likely that different recommendation models constructed with these different kinds of meta-features will be independent/diverse.

3.2. Definitions of Accurate and Diverse Models

Let \( M_E \) be an ensemble learning model constructed with \( n \) base learning models \( \{M_1, M_2, \cdots, M_n\} \), and \( pr_{M_i} \) \((1 \leq i \leq n)\) be the probability of \( M_i \) to make an error prediction on a new coming instance. Then, for ensemble learning, an accurate learning model can be defined as follow.

Definition 3.2. Accurate learning model. A base learning model \( M_i \) \((1 \leq i \leq n)\) is accurate if and only if \( pr_{M_i} < 1/2 \).

Definition 3.2 tells us that a base learning model is accurate if and only if its prediction error is less than 1/2. The rationality of this definition can be demonstrated as follows.

\( \diamond \) First, we construct another ensemble learning model \( \hat{M}_E \) over \( n \) other base learning models \( \{M_1, M_2, \cdots, M_n\} \). Where each base model \( M_i \) \((1 \leq i \leq n)\) has the identical probability to make an error prediction. And the probability is equal to the maximum value of \( \{pr_{M_i} : 1 \leq i \leq n\} \), i.e, \( pr_{\hat{M}_i} = \max\{pr_{M_1}, pr_{M_2}, \cdots, pr_{M_n}\} \) \((1 \leq i \leq n)\).

\( \diamond \) Then, by voting the predictions of the base learning models, we can get that \( pr_{M_E} \leq pr_{\hat{M}_E} \) since \( M_E \) is constructed over a set of base models with lower possibility to make an error prediction. Suppose that the base learning models of the ensemble learning model \( (M_E \ or \ \hat{M}_E) \) are independent with each other, we can get a discrete random variable \( X \) following binomial distribution \((n, pr_{\hat{M}_i})\), where \( X \) indicates the number of models in \( \{M_1, M_2, \cdots, M_n\} \) which make an error prediction. According to binomial distribution, we can get:

\[
pr_{M_E} \leq pr_{\hat{M}_E} = Pr(X \geq \lceil n/2 \rceil) = \sum_{i=\lceil n/2 \rceil}^{n} \binom{n}{i} \left(pr_{\hat{M}_i}\right)^i(1-pr_{\hat{M}_i})^{n-i}. \tag{1}
\]

Where \( \lceil n/2 \rceil \) denotes the smallest integer greater than (or equal to) \( n/2 \).

\( \diamond \) Finally, according to Chernoff’s inequality [Arratia and Gordon 1989], for binomial distribution \((n, p)\),

\[
pr_{M_E} \leq Pr(X \geq k) \leq \exp(-n \cdot D(k/n || p)), \text{if and only if } p < \frac{k}{n} < 1. \tag{2}
\]

Where \( D(a||p) \) is the relative entropy between two Bernoulli distributions with parameters \( a \) and \( p \), and defined as \( D(a||p) = a \cdot \log \frac{a}{p} + (1-a) \cdot \log \frac{1-a}{1-p} \). Corresponding to ensemble learning model \( \hat{M}_E, p = pr_{\hat{M}_i}, k = \lceil n/2 \rceil \) and so \( a = \frac{k}{n} \geq 1/2 \). Therefore, if and only if \( pr_{\hat{M}_E} < 1/2 \), by voting the predictions of \( \{\hat{M}_1, \hat{M}_2, \cdots, \hat{M}_n\} \) as the prediction of \( \hat{M}_E \), Eq. 2 will be always true. And \( pr_{M_E} \) will be bounded by \( \exp(-n \cdot D(k/n || p)) \).
Moreover, in the case of $p = p_{\hat{M}_i} < 1/2$, with the increasing of $n$, the value of $\exp(-n \cdot D(\frac{1}{n} \parallel p))$ will approach 0 since $D(\frac{1}{n} \parallel p_{\hat{M}_i}) > 0$. That is, the more base learning models used, the smaller the probability of $M_E$ to make an error prediction. This will be a very good property for ensemble learning. In a word, all these conclusions will be true under $p_{\hat{M}_i} = \max\{p_{M_1}, p_{M_2}, \ldots, p_{M_n}\} < 1/2$. So in order to get an accurate ensemble learning model $M_E$ over $\{M_1, M_2, \ldots, M_n\}$, $p_{\hat{M}_i}$ should be less than 1/2.

Without the independence/diversity among the base models $\{M_1, M_2, \ldots, M_n\}$, the random variable $X$ in Eq. 1 will not follow binomial distribution and further the Eq. 2 might be false. This will result in that $p_{\hat{M}_E}$ might be no-converging or converging too slowly. This phenomenon has been recognized in learner combination [Cunningham and Carney 2000; Lam 2000]. And the diverse ensemble learner has a better potential to improve the accuracy than non-diverse ensemble learner [Peterson and Martinez 2005; Brown et al. 2005; Kuncheva and Whitaker 2003]. Therefore, there will be a notable question: “How to define or evaluate the independence/diversity among the base learning models?”.

In the field of ensemble learning, it is usually difficult to evaluate the diversity between different learning models directly. Researchers usually resort to the prediction/classification results of the learning models on a given test data. There have been several metrics proposed based on the prediction results to assess the diversity between different models [Kuncheva and Whitaker 2003; Lee and Giraud-Carrier 2013; Cunningham and Carney 2000; Dietterich 2000b]. These metrics can guide us to identify the diverse base models. Meanwhile, Kuncheva et al. [Kuncheva and Whitaker 2003] have stated that, in order to guaranteed the improvement over the performance of base models, there exists a minimum threshold value for each of these diversity metrics to pick up the diverse base models for ensemble learning.

Following these ideas, suppose that $M_i$ and $M_j$ ($1 \leq i \neq j \leq n$) are two different base models, and $R_i$ and $R_j$ are the prediction/classification results of $M_i$ and $M_j$ on a given test data set $D$, we can give the definition of diverse learning model for ensemble learning as follow.

**Definition 3.3.** **Diverse learning model.** Two base models $M_i$ and $M_j$ are diverse with each other if and only if $\psi(R_i, R_j) < \delta$, where $\psi$ is a function which computes the diversity between $M_i$ and $M_j$ based on $R_i$ and $R_j$, and $\delta$ is a given minimum threshold.

In Definition 3.3, the computation of $\psi$ depends on the expression of the prediction results (e.g., $R_i$) of a learning model. There are three general expressions of the prediction results in the field of ensemble learning [Kuncheva and Whitaker 2003].

1. A numeric vector which records the predicted posterior probabilities of all class labels. e.g., for a classification problem with $k$ class labels, the prediction results are a vector with $k$ probability values [Tumer and Ghosh 1996; Tumer and Ghosh 1999].
2. Class label which directly indicates the predicted result [Dietterich 2000a; Dietterich 2000b; Kohavi et al. 1996].
3. Correct/incorrect decision which records whether the predicted label is correct or not [Kuncheva et al. 2003; Ho 1998; Giacinto and Roli 2001].

For the numeric vector based expression, one of the assumptions is that a learning model outputs independent estimates of the posterior probabilities. However, this

---

2In the field of ensemble learning, the independence of base learners is generally called diversity.
is usually not the case since all these posterior probabilities sum up to a constant 1. Moreover, not all the learning models can directly output the posterior probabilities of the class labels. In the field of ensemble learning, the researchers usually define the diversity function $\psi$ in terms of either class label or correct/incorrect decision [Kohavi et al. 1996; Dietterich 2000b; Dietterich 2000a; Kuncheva et al. 2003; Ho 1998; Giacinto and Roli 2001; Kuncheva and Whitaker 2003]. In this paper, we propose a function, which makes full use of the prediction results and considers both of the class label and correct/incorrect decision, to pick up the diverse models for ensemble in next section.

3.3. Base Model Identification for Algorithm Recommendation

The definitions of accurate and diverse learning models in previous section are proposed with respect to single-label learning, and work well on constructing ensemble learning model over the single-label learning problems. However, in this paper, we view algorithm recommendation as a multi-label learning problem and attempt to handle it by ensemble learning method.

Therefore, there will be a question: "How to identify the accurate and diverse base models for algorithm recommendation with respect to multi-label based meta-data?" And this question can be answered by dividing it into the following ones.

- **Question 1**: How to construct base recommendation models for ensemble on multi-label based meta-data?
- **Question 2**: How to identify an accurate recommendation model?
- **Question 3**: How to identify a diverse recommendation model?

3.3.1. Answer to Question 1. This paper employs the frequently-used multi-label ensemble learning method, data transformation, to construct ensemble recommendation model on the multi-label based meta-data. That is, we first transform the multi-labeled meta-data into multiple single-labeled meta-data, and then build recommendation models on these single-labeled meta-data as the base models of the ensemble.

Here, the process of multi-labeled meta-data transformation consists of two steps: i) in meta-feature space, a number of different sets of multi-labeled meta-data are generated with respect to different combinations of the existing meta-features (See details in Section 4.2); ii) in meta-target space, for each multi-labeled meta-data generated in i), multiple different sets of single-labeled meta-data are generated according to different labels of the meta-target in a specific way (See details in Section 4.3).

3.3.2. Answer to Question 2. Once we achieve the base models, it is straightforward to identify the accurate base models according to Definition 3.2. That is, for a given base model, if its classification error rate on test data is less than 1/2, it will be accurate, otherwise not.

3.3.3. Answer to Question 3. In order to identify the diverse models, according to Definition 3.3, two critical dimensions should be considering: one is to find a function $\psi$ to evaluate the diversity between two models, and the other is to set a proper threshold $\delta$ to pick up the diverse models. However, the existing researches usually just supply the function $\psi$ but no any effective approach to preassign the threshold $\delta$ [Kuncheva and Whitaker 2003; Lee and Giraud-Carrier 2013; Cunningham and Carney 2000; Dietterich 2000b].

In this paper, we present a statistical method which can not only quantize the diversity between two models but also adaptively set the threshold $\delta$. Different from the existing diversity evaluation functions acting on the prediction results in terms of either the class labels or the direct/indirect decisions, the proposed diversity evaluation method concerning both of them, which will make full use of prediction results.

ACM Journal Name, Vol. V, No. N, Article A, Publication date: January YYYY.
Suppose there are two different learning models $M_1$ and $M_2$, and a data set $D$ with $K$ class labels $\{C_1, C_2, \ldots, C_K\} (K \geq 2)$, then we can construct a $K \times K$ contingency table (see Table I) based on the class labels and correct/incorrect decisions predicted by $M_1$ and $M_2$ on $D$.

Table I: The $K \times K$ contingency table

| Classified label | $C_1$ | $C_2$ | $\cdots$ | $C_K$ | Total |
|------------------|-------|-------|----------|-------|-------|
| $C_1$            | $N_{1,1}$ | $N_{1,2}$ | $\cdots$ | $N_{1,K}$ | $N_{1,*}$ |
| $C_2$            | $N_{2,1}$ | $N_{2,2}$ | $\cdots$ | $N_{2,K}$ | $N_{2,*}$ |
| $\vdots$         | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ | $\vdots$ |
| $C_K$            | $N_{K,1}$ | $N_{K,2}$ | $\cdots$ | $N_{K,K}$ | $N_{K,*}$ |
| Total            | $N_{*,1}$ | $N_{*,2}$ | $\cdots$ | $N_{*,K}$ | $N$     |

In Table I, $N_{i,j}$ ($1 \leq i, j \leq K$) is the number of test instances incorrectly classified as $C_i$ and $C_j$ by $M_1$ and $M_2$, respectively. $N_{i,*} = \sum_{j=1}^{K} N_{i,j}$ ($1 \leq i \leq K$), $N_{*,j} = \sum_{i=1}^{K} N_{i,j}$ ($1 \leq j \leq K$) and $N = \sum_{i=1}^{K} N_{i,*} = \sum_{j=1}^{K} N_{*,j} = \sum_{i,j=1}^{K} N_{i,j}$. $N$ is the total number of instances of $D$ which are incorrectly predicted by either $M_1$ or $M_2$.

With this $K \times K$ contingency table, we can get the diverse measure $\kappa$ by Eq. 3. The greater the value of $|\kappa|$, the smaller the diversity between $M_1$ and $M_2$.

$$\kappa = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2},$$  

(3)

Where $\Theta_1 = \sum_{i=1}^{K} \frac{N_{i,*}}{N}$ and $\Theta_2 = \sum_{i=1}^{K} \frac{N_{i,*}}{N} \times \frac{N_{i,*}}{N}$.

According to the contingency table analysis, the joint frequency distribution of $C_i$ and $C_j$ predicted by $M_1$ and $M_2$ is $\frac{N_{i,j}}{N}$, and $\frac{N_{i,*}}{N}$ and $\frac{N_{*,j}}{N}$ correspond to the marginal frequency distributions of $C_i$ and $C_j$. The rationality of $\kappa$ being able to evaluate how strong the independence between $M_1$ and $M_2$ is demonstrated as follows.

1. Suppose that $M_1$ and $M_2$ are independent of each other, the expected joint distribution of $C_i$ and $C_j$ would be $\frac{N_{i,j}}{N} \times \frac{N_{i,*}}{N} \times \frac{N_{*,j}}{N}$. In Eq. 3, the numerator $\Theta_1 - \Theta_2$ can also be represented as $\sum_{i=1}^{K} \left( \frac{N_{i,j}}{N} \times \frac{N_{i,*}}{N} \times \frac{N_{*,j}}{N} \right)$. Therefore, in the case that $M_1$ is independent of $M_2$, $\Theta_1 - \Theta_2$ will be quite close to 0 in practice.

2. If the learner $M_1$ is positively related to $M_2$, an instance being predicted as $C_i$ ($1 \leq i \leq K$) by $M_1$ means that it is more likely that $M_2$ classifies the instance to $C_i$ as well. This will increase the value of $\frac{N_{i,j}}{N}$, i.e., the elements on the main diagonal of the contingency table. Otherwise, if $M_1$ is negatively related to or independent with $M_2$, the instance will be predicted as different classes by these two models. This will reduce the value of $\frac{N_{i,j}}{N}$. So we can get that the value of $N_{i,j}$ can reflect the dependence of two different learners. That is why we define the metric $\kappa$ by the elements on the main diagonal of the contingency table.

3. The denominator $1 - \Theta_2$ of $\kappa$ plays a role to limit the value of $\kappa$ into the range $[-1, 1]$.

If the classification results on the test data are always identical, $\sum_{i=1}^{K} N_{i,i} = N$, so $\Theta_1 = 1$ and $\kappa$ achieves its maximum value 1. If $M_1$ and $M_2$ are independence with
each other, $\kappa$ will be 0 or quite near 0. And $\kappa < 0/0 > 0$ means $M_1$ is negative/positive related to $M_2$. The greater the value of $|\kappa|$, the stronger the dependence between $M_1$ and $M_2$.

In practical application, the metric $\kappa$ is estimated via the prediction results of the learning models on only a sample rather than the whole population. This might also be the reason that there needs a threshold $\delta$ in Definition 3.3. Therefore, we need to further understand the statistical significance of $\kappa$, including the statistical significance of $\kappa \neq 0$ and its confidence interval. And the confidence interval will be set as the threshold $\delta$ in Definition 3.3.

For this purpose, we need to find a statistic for significant test of $\kappa$. As we know, the distribution of the class labels predicted by a learner on a $K$-class classification problem would follow either binomial ($K = 2$) or multi-nominal ($K > 2$) distribution. Both of binomial, multi-nominal distributions are derived from exponential family of distributions. Meanwhile, inspiring by the idea that the independence between two variables, which follow the well-known exponential distribution (i.e, normal distribution), is usually statistically tested by a $t$-statistic, we attempt to employ a $t$-statistic in Eq. 4 to test the significance of $\kappa \neq 0$, and further determine whether two learning models $M_1$ and $M_2$ are independent with each other or not.

$$t = \frac{\kappa \sqrt{1 - \kappa^2}}{\sqrt{N - 2}}$$

(4)

The $t$ statistic follows the Student’s $t$-distribution with freedom of degree $N - 2$ under the null hypothesis that $M_1$ and $M_2$ are independent with each other. If the $t$-statistic test accepts the null hypothesis under given significance level $\alpha$, we can conclude that the measure $\kappa$ has no significant difference with 0, i.e., $M_1$ and $M_2$ are independent with each other.

According to Eq. 4, its inverse can be calculated as follow

$$\kappa = \frac{t \sqrt{N - 2}}{\sqrt{N - 2 + t^2}}$$

(5)

Where $t$ statistic follows student distribution with degree of freedom $N - 2$.

Let $t_c$ denote the critical value of student distribution with degree of freedom $N - 2$ under a given significance level $\alpha$ (e.g., $\alpha = 0.05$), then we can get the confidence interval of $\kappa = 0$ as $[-t_c \sqrt{\frac{N - 2}{t_c^2}}, t_c \sqrt{\frac{N - 2}{t_c^2}}]$. If $\kappa$ calculated between two single learning models falls into this interval, we can conclude that these two models are statistically independent with each other under the given significance level $\alpha$. Based on this confidence interval, we can set the minimum threshold $\delta = t_c \sqrt{\frac{N - 2}{t_c^2}}$. And the diverse models can be detected by comparing $\kappa$ with $\delta$ directly. That is, two models $M_1$ and $M_2$ are independent with each other if and only if $|\kappa| < \delta$.

4. ENSEMBLE LEARNING BASED ALGORITHM RECOMMENDATION

This section first shows the general view of the proposed ensemble learning based recommendation method, then describes the process of model construction in details.

4.1. General View

Firstly, different kinds of meta-features and the multi-labeled meta-target are collected over a set of historical classification problems; afterwards, by joining different combinations of these meta-features and the multi-labeled meta-target together,
different sets of multi-labeled meta-data will be generated. Secondly, the base recommendation models are constructed on each of the generated multi-labeled meta-data. Thirdly, a multi-label ensemble learning recommendation model will be achieved by combining these base recommendation models together. Fig. 2 gives the general view of the proposed method which consists of three steps: i) meta-data preparation, ii) base recommendation model construction and iii) ensemble model construction.

1) Meta-data preparation
Meta-data is collected from a set of historical classification problems. i) For meta-feature collection, all the \( q \) different kinds of data characterization methods are utilized on the historical classification problems to get \( q \) groups of meta-features. ii) For meta-target collection, the appropriate algorithms of each historical classification problem are identified by statistically comparing all the candidate algorithms in terms of a given performance metric (such as, classification accuracy). And these appropriate algorithms form the multi-label based meta-target.

Different kinds of meta-features reflect the properties of a classification problem in different viewpoints. Combinations of these meta-features will give us a more comprehensive understanding of the problem. Inspired by this idea, this paper attempts to construct the base recommendation models with respect to different combinations of these meta-features in a specific way. Firstly, \( q \) different sets of meta-features can be combined to make out \( t = 2^q - 1 \) combinations (See the generation process in Section 4.2). Then, by merging these \( t \) combinations and the multi-labeled meta-target together, \( t \) different sets of multi-labeled meta-data can be generated.

2) Base recommendation model construction
For each multi-labeled meta-data, firstly, the data transformation method is performed to transform the multi-labeled meta-data into multiple single-labeled meta-data, and then the base recommendation models will be generated on these single-labeled meta-data. The details of this process will be described in Section 4.3.

3) Ensemble recommendation model construction

Once achieving the base recommendation models, the accurate and diverse base models are identified for constructing an ensemble recommendation model according to Section 3.3. For a new coming classification problem, the recommendations of these identified base models are combined in a specific way to form the recommended algorithms for the new problem. The detailed process of ensemble model construction will be introduced in Section 4.4.

4.2. Meta-data Preparation

Let \( A = \{A_i : i = 1, 2, \cdots, k\} \) be a set of \( k \) candidate classification algorithms, \( P = \{p_i : i = 1, 2, \cdots, n\} \) be a set of \( n \) historical classification problems, and \( F = \{F_i : i = 1, 2, \cdots, q\} \) be \( q \) different data set characterization functions used for meta-feature extraction. \( X_i^j = F_j(p_i) \) denotes the meta-features extracted by \( F_j \) (\( 1 \leq j \leq q \)) on the classification problem \( p_i \) (\( 1 \leq i \leq n \)).

Suppose that \( D = \{(X_i, Y_i) : i = 1, 2, \cdots, n\} \) denotes a set of multi-label meta-instances, where \( X_i \) is the meta-features extracted from the classification problem \( p_i \) by the function(s) in \( F \), that is, \( X_i \in nchoosek((X_1^1, X_2^1, \cdots, X_q^1)) \) and here \( nchoosek(Z) \) outputs all the possible combinations of the elements in \( Z \). For example, let \( Z = \{a, b, c\} \), then \( nchoosek(Z) = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \); and \( Y_i = \{Y_{i,j} : j = 1, 2, \cdots, k\} \) represents the multi-label based meta-target on \( p_i \) and \( Y_{i,j} = 1 \) or 0 indicates the algorithm \( A_j \) is appropriate or inappropriate on \( p_i \).

It is noted that, there are many methods to generate different combinations of meta-features from the given \( q \) kinds of meta-features. The combination function \( nchoosek() \) being chosen is due to the fact that it can not only generate multiple different sets of meta-data for base model construction, but also help to find whether the combinations of different kinds of meta-features are better, and further discover the salient meta-features for algorithm recommendation.

With the combination function \( nchoosek() \) on \( q \) different kinds of meta-features, we can generate \( t = 2^q - 1 \) different sets of meta-features and furthermore \( t \) sets of multi-labeled meta-data.

4.3. Base Recommendation Model Construction

For each one of the \( t \) sets of multi-labeled meta-data, the process to construct the base recommendation models is identical. This section will illustrate this process by taking one given multi-labeled meta-data \( D \) as an example.

![Fig. 3: Data sets transformed by BR method](image-url)
At first, the multi-labeled meta-data \( D \) is transformed into multiple single-labeled data sets in meta-target space. The popular data transformation method, Binary Relevance (BR) \cite{Tsoumakas2010}, is performed. The BR method transforms \( D \) into \( k \) data sets \( D_{A_j, j = 1, \cdots, k} \) which contain all the instances of \( D \) labeled by whether algorithm \( A_j \) is appropriate or not. This means, for each instance \((X_i, Y_i)\) of \( D \), the corresponding instance of \( D_{A_j} \) is \((X_i, Y_{i,j})\). Fig. 3 shows the \( k \) single-labeled data sets produced by BR on the multi-labeled meta-data \( D \).

Afterwards, \( k \) recommendation models will be learned on these \( k \) binary learning data sets by a specific classification algorithm (e.g., Decision Tree). These \( k \) models \( \{M_1, M_2, \cdots, M_k\} \) constitute the base recommendation models on \( D \). Here, \( M_i \) is a learning model which can output the probability value \( pr_i \) (1 \( \leq \) \( i \) \( \leq \) \( k \)) of the candidate algorithm \( A_i \) being appropriate.

4.4 Combination of Base Recommendation Models

By combining the \( q \) different kinds of meta-features, \( t = 2^q - 1 \) different sets of multi-labeled meta-data are achieved. After constructing base recommendation models on each one of the \( t \) multi-labeled meta-data, we will get a learning model matrix \( M_{t \times k} \), where \( M_{i,j} (1 \leq i \leq t, 1 \leq j \leq k) \) denotes the model learned on the \( j \)th data set transformed from \( i \)th multi-labeled meta-data by BR method.

\[
M_{t \times k} = \begin{bmatrix}
M_{1,1} & M_{1,2} & \cdots & M_{1,k} \\
M_{2,1} & M_{2,2} & \cdots & M_{2,k} \\
\vdots & \vdots & \ddots & \vdots \\
M_{t,1} & M_{t,2} & \cdots & M_{t,k}
\end{bmatrix}
\]

(6)

Let \( M_{i,j} (1 \leq j \leq k) \) denote the \( j \)th column of the matrix \( M_{t \times k} \). According to the process of BR transformation method, \( M_{i,j} \) consists of all the \( t \) learning models with respect to the candidate algorithm \( A_j \). Thus, it is possible for us to construct the ensemble learning model improve the prediction of \( A_j \) based on the \( t \) models in \( M_{i,j} \).

According to Corollary 3.1 the necessary and sufficient condition of constructing an accurate ensemble learning model is that the base models are accurate and diverse with each other. Thus, the ensemble model construction is a process to find out the accurate and diverse models in \( M_{i,j} \).

Algorithm 1 gives the filtering method on the \( t \) models in \( M_{i,j} \). In this algorithm, besides \( M_{i,j} \), there are two other input variables \( Accs \) and \( Outs \) with respect to the models in \( M_{i,j} \), where \( acc_i \in Accs \) (1 \( \leq \) \( i \) \( \leq \) \( t \)) denotes the classification accuracy of \( M_{i,j} \) on a validation data 5 and \( out_i \in Outs \) represents the outputs (i.e., predicted labels and correct/incorrect decisions) of \( M_{i,j} \) on the validation data. \( FlagList \) records the filtering results, where \( f_i \in \{0, 1\} \) and \( f_i = 0/1 \) means that \( M_{i,j} \) is filtered out/reserved for ensemble model construction.

Algorithm 1 consists of two filters: i) accuracy based filter (lines 2-3) and ii) diversity based filter (lines 5-14). The first filter is used to find out the accurate models. In this filter, the models whose classification accuracy being smaller than 1/2 are filtered out (i.e., set the corresponding flags in \( FlagList \) as 0) according to Definition 3.2 of accurate model. The second filter aims at finding out the diverse models. In this filter, if the \( \kappa \) between two specified models is greater than the predefined threshold \( \delta \) according to Definition 3.3 of diverse model, the model with lower classification accuracy will be filtered out. Where the threshold \( \delta \) is set according to Eq. 5.

5The validation data is drawn from the original meta-data and never used for model construction.
ALGORITHM 1: ModelFilter()

Inputs: \( M_{i,j} = \{M_{1,j}, M_{2,j}, \ldots, M_{t,j}\} \);
\( \text{Accs} = \{\text{acc}_1, \text{acc}_2, \ldots, \text{acc}_t\} \);
\( \text{Outs} = \{\text{out}_1, \text{out}_2, \ldots, \text{out}_t\} \);

Output: FlagList = \( \{f_1, f_2, \ldots, f_t\} \);

1. Initialization: \( f_i = 1 : (1 \leq i \leq t) \);

// Part 1: Accuracy based filter
2. for \( i = 1 \) to \( t \) do
3.  if \( \text{acc}_i < 0.5 \) then
4.    \( f_i = 0 \);

// Part 2: Diversity based filter
5.     IX = sort(\( \text{Accs} \), "descend") // IX is the indices of the models sorted by \( \text{Accs} \) in a descending order
6. for \( i = 1 \) to \( t - 1 \) do
7.   \( Id_i = \text{IX}_i \); // The \( i \)th element of IX
8.   if \( f_{Id_i} == 1 \) then
9.     for \( j = (i + 1) \) to \( t \) do
10.    \( Id_j = \text{IX}_j \); // The \( j \)th element of IX
11.    if \( f_{Id_j} == 1 \) then
12.       \( \kappa = \text{DiversityComp}() \) // According to Eq. 3 of \( \kappa \)
13.       if \( \kappa \geq \delta \) then
14.          \( f_{Id_j} = 0 \); // \( \delta \) is set according to Eq. 5

With the help of Algorithm 1, we can get a flag matrix \( G_{t \times k} \), where the column \( f_{i,j} \) \((1 \leq j \leq k)\) is achieved by applying Algorithm 1 on the models \( M_{i,j} \) of the \( j \)th column in the model matrix \( M_{t \times k} \), \( f_{i,j} \in \{0, 1\} \) \((1 \leq i \leq t)\), and \( f_{i,j} = 1 \) means the model \( M_{i,j} \) is chosen for ensemble model construction.

\[
G_{t \times k} = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,k} \\
 f_{2,1} & f_{2,2} & \cdots & f_{2,k} \\
 \vdots & \vdots & \ddots & \vdots \\
 f_{t,1} & f_{t,2} & \cdots & f_{t,k} \end{bmatrix} \quad (7)
\]

Meanwhile, let \( M_{i,\cdot} \) \((1 \leq i \leq t)\) be the \( i \)th row of the model matrix \( M_{t \times k} \), which consists of the models learned from the \( i \)th multi-labeled meta-data. For a new coming classification problem \( p_{new} \), each model \( M_{i,j} \in M_{i,\cdot} \) \((1 \leq j \leq k)\) predicts the probability \( pr_{i,j} \) to recommend the algorithm \( A_j \) to \( p_{new} \). So we can further get a matrix \( P_{t \times k} \) of probabilities predicted by the model matrix \( M_{t \times k} \) on \( p_{new} \) as follow.

\[
P_{t \times k} = \begin{bmatrix} pr_{1,1} & pr_{1,2} & \cdots & pr_{1,k} \\
 pr_{2,1} & pr_{2,2} & \cdots & pr_{2,k} \\
 \vdots & \vdots & \ddots & \vdots \\
 pr_{t,1} & pr_{t,2} & \cdots & pr_{t,k} \end{bmatrix} \quad (8)
\]

With the predicted probability value matrix \( P_{t \times k} \) and the flag matrix \( G_{t \times k} \), the ensemble recommendation model will estimate the probability value of the candidate algorithm \( A_j \) \((1 \leq j \leq k)\) being appropriate on \( p_{new} \) by the Eq. 9.
In Eq. 9, the denominator $\sum_{i=1}^{t} f_{i,j}$ denotes the number of models chosen by Algorithm $A_j$ on $M_{i,j}$, i.e., the number of based models used for ensemble learning with respect to algorithm $A_j$. The molecule of Eq. 9 can be viewed as a kind of weighted voting ensemble, where $pr_{i,j}$ denotes the weight of model $M_{i,j}$ and it works if and only if $f_{i,j} = 1$, i.e., $M_{i,j}$ is picked up as a base model for ensemble learning.

Afterwards, the ensemble recommendation model can rank the $k$ candidate algorithms according to the estimated probability values $\{pr_{en}(A_1), pr_{en}(A_2), \cdots, pr_{en}(A_k)\}$. Meanwhile, the algorithm $A_j$ will be recommended as an appropriate one for $p_{new}$ if and only if $pr_{en}(A_j)$ is greater than a specific threshold (e.g., 1/2).

Table II: Example for ranking strategy

| Algorithm | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ |
|-----------|-------|-------|-------|-------|-------|
| Probability value | 0.7 | 0.4 | 0.8 | 0.5 | 0.7 |
| Ranks | 2 | 5 | 1 | 4 | 3 |
| Ranks (considering ties) | 2.5 | 5 | 1 | 4 | 2.5 |

Meanwhile, these $k$ probability values can be further used to learn a ranking list of the candidate algorithms in $k$ on $p_{new}$. The algorithm with the highest probability will be ranked first, the algorithm with the second highest probability will be ranked second, and so on. In case of ties, average ranks are assigned. This strategy can be illustrated by the example in Table II in which we give the estimated probability values of five candidate algorithms and corresponding ranks.

5. EXPERIMENTAL STUDY

In this section, we experimentally evaluate the performance of the proposed ensemble learning based algorithm recommendation method, including the experimental setup and analyses of the results.

5.1. Experimental Setup

In order to evaluate the effectiveness of the ensemble learning based algorithm recommendation method, confirm the applicability in practice, and guarantee the reproducibility of the results, we set up our experiments as follows.

5.1.1. Benchmark Classification Problem. 84 extensively-used public classification problems being available on UCI repository are employed in the experiments. Table III shows the statistical summary of these problems in terms of the number of attributes, the number of instances and the number of classes.

Moreover, in order to guarantee the reliability and soundness of the conclusion, more classification problems should be employed. Thus, with the help of the problem generation method, Datasetoids, which was proposed in [Soares 2009] and aimed to obtain

[^4]: http://archive.ics.uci.edu/ml/datasets.html.

ACM Journal Name, Vol. V, No. N, Article A, Publication date: January YYYY.
Table III: Description of the 84 classification problems

| ID | Name                  | Attributes | Instances | Classes |
|----|-----------------------|------------|-----------|---------|
| 1  | anneal                | 38         | 898       | 6       |
| 2  | coral                 | 17         | 101       | 7       |
| 3  | arrhythmias           | 279        | 452       | 16      |
| 4  | audiology             | 20         | 225       | 24      |
| 5  | australian            | 14         | 690       | 2       |
| 6  | avar                   | 25         | 205       | 7       |
| 7  | balance-scale         | 4          | 625       | 3       |
| 8  | breast-cancer         | 9          | 286       | 2       |
| 9  | breast-w              | 9          | 699       | 2       |
| 10 | car                   | 6          | 1728      | 4       |
| 11 | clever                | 11         | 303       | 2       |
| 12 | cnote                 | 9          | 1473      | 3       |
| 13 | colic                 | 22         | 368       | 2       |
| 14 | connect-4             | 42         | 13512     | 3       |
| 15 | credit-a              | 15         | 690       | 2       |
| 16 | credit-g              | 20         | 1600      | 2       |
| 17 | crx                   | 15         | 690       | 2       |
| 18 | cylinder-bands        | 39         | 540       | 2       |
| 19 | dermatology           | 34         | 366       | 6       |
| 20 | diabetes              | 9          | 796       | 2       |
| 21 | soil                  | 7          | 336       | 8       |
| 22 | flags                 | 29         | 194       | 8       |
| 23 | german                | 15         | 1090      | 2       |
| 24 | glass                 | 9          | 214       | 7       |
| 25 | haberman              | 3          | 306       | 2       |
| 26 | hayes-kivl            | 4          | 132       | 3       |
| 27 | heart-c               | 13         | 301       | 5       |
| 28 | heart-h               | 13         | 294       | 5       |
| 29 | heart-statlog         | 13         | 270       | 2       |
| 30 | hepatitis             | 19         | 155       | 2       |
| 31 | horse-colic-ORIG      | 21         | 368       | 2       |
| 32 | hypothyroid           | 23         | 3163      | 2       |
| 33 | hypothyroid-dimension | 29         | 3772      | 4       |
| 34 | hypersphere           | 34         | 351       | 2       |
| 35 | icra                  | 4          | 150       | 3       |
| 36 | kid_nipponeses1       | 14         | 5687      | 9       |
| 37 | kid_nipponeses2       | 13         | 4274      | 9       |
| 38 | kid-predictive-control| 61         | 600       | 6       |
| 39 | kn-nlp                | 36         | 3196      | 2       |
| 40 | labor                 | 16         | 57        | 2       |
| 41 | led                   | 7          | 3200      | 10      |
| 42 | letter                | 16         | 20000     | 26      |

a large number of classification problems for algorithm recommendation [Prudência et al. 2011][Prudencio et al. 2011][Halabi Echeverry et al. 2012][Prudêncio et al. 2011], we extend the 84 publicly UCI classification problems into 1090 (84 data sets and 1006 datasetoids) different classification problems. The Datasetoids method is quite simple, it achieves the new classification problems by exchanging role of each nominal attribute with that of a target concept, i.e., viewing the nominal attribute as the new target concept.

5.1.2. Meta-feature Collection. Five different types of meta-features are extracted from the 1090 classification problems. They are i) the statistical and information-theory based, ii) the model structure based, iii) the landmarking based, iv) the problem complexity based and v) the structural information based. See Appendix [6] for the details.

5.1.3. Meta-target Collection. Meta-target tells us the appropriate candidate algorithms for each of the 1090 classification problems. Next, we introduce the candidate classification algorithms in the experiments and how to identify the appropriate algorithms for a given classification problem.

(1) Candidate classification algorithms
In order to guarantee the generality of the experimental results, 13 different types of classification algorithms are picked up as the candidates.

These algorithms include i) the probability based algorithm Bayes Network; ii) tree based algorithms C4.5, RandomForest and RandomTree; iii) rule-based algorithms PART, Ripper and NNge; iv) Gaussian function based algorithm RBFNetwork, and v) support vector machine based algorithm SMO.

Besides the above nine single classification algorithms, we also pick up two kinds of well-known ensemble classification algorithms: Boosting and Bagging. They are applied with the base classifiers Naive Bayes (NB) and C4.5, respectively.

(2) Appropriate algorithm identification
The multi-label based meta-target indicating the appropriate algorithms for a classification problem $D$ can be expressed in terms of a binary value vector $B_D = \langle b_1, b_2, \cdots, b_{13} \rangle$, where $b_i = 1$ means that the corresponding candidate algorithm $A_i$ ($1 \leq i \leq 13$) is appropriate. The appropriate algorithms are identified by their performance metrics (i.e., classification accuracy) on $D$ as follows.

(a) Process of classification accuracy estimation on $D$

In order to get a stable estimation of classification accuracy of the candidate algorithms on $D$, $5 \times 10$-fold stratified cross-validation is performed as the following steps. i) The problem $D$ is randomly split into ten mutually exclusive subsets $D_1, D_2, \cdots, D_{10}$ of equal size, and $D = \bigcup_{j=1}^{10} D_j$. ii) $D - D_j$ and $D_j$ ($j \in \{1, 2, \cdots, 10\}$) are used as the training and test sets, respectively. Each algorithm $A_i$ ($1 \leq i \leq 13$) is trained on $D - D_j$, and its classification accuracy is estimated on $D_j$. iii) Repeat i) and ii) five times on $D$ whose instances are randomly re-ordered. Afterwards, for each candidate algorithm $A_i$ ($1 \leq i \leq 13$), we will get a vector $Acc_i = \langle acc_{i,1}, acc_{i,2}, \cdots, acc_{i,50} \rangle$ with 50 classification accuracies.

(b) Binary-valued based meta-target $B_D$ identification

In order to identify the real appropriate algorithms from 13 candidate algorithms according to from the collected performance sets $\{Acc_1, Acc_2, \cdots, Acc_{13}\}$ on $D$, the statistical algorithm selection is a reasonable and commonly-used approach [Pizarro et al. 2002].

To find out the superior algorithms from three or more candidate algorithms, the traditional statistical methods usually resort to multiple paired $t$-tests. However, it has been proved that this approach usually leads to high Type I error.\footnote{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.}

For solving this problem, we turn to the \textit{multiple comparison procedure}. The multiple comparison procedure is a statistical test technique which helps us compare three or more groups of metrics (e.g., classification accuracy) while controlling the probability to make the statistical Type I error [Pizarro et al. 2002]. Moreover, it allows us to concern with a set of candidate algorithms not significantly different from the best one rather than a single algorithm. Therefore, the multiple comparison procedure is an effective method for multi-labeled meta-target collection.

Therefore, in our experiment, as suggested in [Demšar 2006], we employ the non-parametric multiple comparison procedure, Friedman followed by Holm’s procedure test, to obtain the binary-value based meta-target $B_D$ for the problem $D$ according to $\{Acc_1, Acc_2, \cdots, Acc_{13}\}$ as follows.

i. Applying Friedman test on $\{Acc_1, Acc_2, \cdots, Acc_{13}\}$, the null hypothesis of Friedman test is there does not exist significant difference among these 13 algorithms. If the result of the test support the null hypothesis, all these 13 algorithms will be viewed as the appropriate ones. This means that $\forall b_i$ of $B_D$ ($1 \leq i \leq 13$), $b_i = 1$. And the multiple comparison procedure is over.

ii. Otherwise, there will exist significant difference among these candidate algorithms. In this case, we should apply the post-hoc Holm’s procedure test to further find the real appropriate algorithms. At first, the algorithm with the highest average classification accuracy is picked up as a reference. And the Holm’s procedure test is performed to identify the appropriate algorithms from the rest ones. The algorithms that have no significant differences with
the reference are viewed as the appropriate algorithms. Of course, the refer-
ence is an appropriate one as well. Afterwards, for each value $b_i (1 \leq i \leq 13)$
in $B_D$, if the corresponding algorithm $A_i$ is identified as an appropriate one,
$b_i = 1$, otherwise, $b_i = 0$.

5.1.4. Recommendation model construction. The proposed ensemble multi-label learning
based recommendation model combines a set of base models constructed on different
sets of meta-data. In order to demonstrate whether the proposed ensemble method is
competitive in constructing the recommendation model, we compare the performance
of the ensemble recommendation model with those of the base models.

When constructing the base model on a given multi-labeled meta-data, the data
transformation method BR first transforms the multi-labeled meta-data into multiple
single-labeled meta-data, and then the well-known classification algorithm, decision
tree, is applied on these single-labeled meta-data to get the base recommendation mod-
els. The tree-based learner being used is due to the fact that the it is quite effective to
be explored and has good explanation.

Moreover, one critical factor affecting the performance of ensemble learning is that
whether the base models are accurate and diverse. In order to testify how the accu-
rate and diverse base models affect the recommendation performance of the ensemble
recommendation method in our experiment, we compare the recommendations of the
ensemble models constructed with respect to four different sets of base models, in-
cluding i) all learned base models ii) only accurate base models, iii) only diverse base
models and iv) both of accurate and diverse models.

5.1.5. Metrics to Evaluate Recommendation Models. In order to measure the performance
of the recommendation model, two metrics which have been used to evaluate the multi-
label methods are defined as follows.

For a given classification problem $p \in \mathbb{P}$, let $RR_p = \langle r_{r_1}, r_{r_2}, \ldots, r_{r_k} \rangle$ represent
the recommended rank list of $k$ candidate algorithms $\&$ on $p$. Meanwhile, suppose that
$TB_p = \langle t_{b_1}, t_{b_2}, \ldots, t_{b_k} \rangle$ ($t_{b_i} \in \{0, 1\}$) indicates whether a candidate algorithm $A_i$
is true appropriate (i.e., $t_{b_i} = 1$) or not (i.e., $t_{b_i} = 0$) on $p$, $Y$ be the set of indexes of true
appropriate algorithms and $\hat{Y}$ be the set of indexes of the unappropriate algorithms on
$p$.

Ranking Loss represents the number of times that unappropriate algorithms are
ranked higher than the true appropriate algorithms. Ranking loss of the recommended
rank list $RR_p$ is defined as follow.

**Definition 5.1. Ranking Loss**

\[
R\text{-Loss}(RR_p) = \frac{1}{|Y| \cdot |\hat{Y}|} |(i_a, i_b) : r_{r_i_a} > r_{r_i_b}, (i_a, i_b) \in Y \times \hat{Y}| \tag{10}
\]

For algorithm recommendation results in the form of ranking, in a practical appli-
cation, the 1st ranked algorithm is usually in favor, then the 2nd ranked one, and so
forth. Therefore, it is natural for the users to ask that whether the top ranked algo-
rithms are true appropriate or not. In this case, precision of ranking results, which
has been widely-used in the field of information retrieval to measure whether the top
ranked records are true relevant [Baeza-yates and Ribeiro-neto 1999], is employed as
a measure to evaluate how well the algorithm ranking based recommendation results.

Precision at $m$ to measure the accuracy of the top $m$ recommended algorithms on
problem $D$ is calculated by Eq. [11]

\[
\text{Precision}(m) = \frac{\text{number of real appropriate algorithms within top } m}{m}. \tag{11}
\]
With precision at \( m \), average precision \cite{Baeza-yates and Ribeiro-neto 1999} to measure the accuracy of the recommendation result \( RR_D \) on problems \( D \) is defined as follows.

**Definition 5.2. Average Precision**

\[
AP(RR_D) = \frac{k}{\sum_{i=1}^k tb_i} \sum_{m=1}^k \text{Precision}(m) \times \delta(m).
\] (12)

Where \( k \) denotes the number of the candidate algorithms, and \( \delta(m) \) is a binary function to indicating whether the \( m \)th ranked algorithm in \( RR_D \) is real appropriate (\( \delta(m) = 1 \)) or not (\( \delta(m) = 0 \)). The numerator \( \sum_{i=1}^k tb_i \) represents the number of the real appropriate algorithms on \( D \).

5.1.6. Recommendation method validation. After the multi-labeled meta-data \( D_M \) with 1090 instances is acquired. The 5 \times 10-fold cross-validation procedure is applied on \( D_M \) to empirically evaluate the proposed algorithm recommendation method as follows.

1. \( D_M \) is randomly divided into 10 sub data sets in the same size \( \{D_M_i : 1 \leq i \leq 10\} \), \( D_M = \bigcup_{i=1}^{10} D_M_i \), and \( D_{M_i} \cap D_{M_j} = \emptyset \) (\( 1 \leq i \neq j \leq 10 \)).
2. Each sub data set \( D_{M_i} \) is viewed as the test data \( D_{te} \), and the union of rest sub data sets \( \bigcup_{j=1 \wedge j \neq i}^{10} D_{M_j} \) are randomly divided into two equal-size parts: training data \( D_{tr} \) and valid data \( D_{va} \).
3. Construct the base recommendation models on the training data \( D_{tr} \), and filter the base models by their predictions on the valid data \( D_{va} \) according to Algorithm 1 in Section 4.4.
4. Combine the filtered based recommendation models to form the ensemble recommendation model, and evaluate the ensemble model in terms of Ranking Loss and Precision on the test data \( D_{te} \).
5. Repeat the above four steps five times, for each time, the order of the 1090 instances in \( D_M \) is rearranged randomly.

5.2. Results and Analysis

In this section, we give the results of the comparison between the proposed ensemble recommendation model and the base recommendation models in terms of Ranking Loss and Precision, respectively.

For the sake of understanding the results, we notes the different combinations of the five different kinds of meta-features in Table [IV] where numbers “1”, “2”, “3”, “4” and “5” appearing in column “Comment” represent five different kinds of meta-features, respectively.

| Notation | Comment | Notation | Comment | Notation | Comment | Notation | Comment |
|----------|---------|----------|---------|----------|---------|----------|---------|
| 1        | {1}     | 9        | {1,2,3} | 17       | {1,2,4} | 26       | {1,2,4,5} |
| 2        | {2}     | 10       | {2,3}   | 18       | {1,2,5} | 26       | {1,2,3,4} |
| 3        | {3}     | 11       | {2,4}   | 19       | {1,3,4} | 27       | {1,2,3,5} |
| 4        | {4}     | 12       | {2,5}   | 20       | {1,3,5} | 28       | {1,2,4,5} |
| 5        | {5}     | 13       | {3,4}   | 21       | {1,4,5} | 29       | {1,3,4,5} |
| 6        | {1,2}   | 14       | {3,5}   | 22       | {2,3,4} | 30       | {2,3,4,5} |
| 7        | {1,3}   | 15       | {4,5}   | 23       | {2,3,5} | 31       | {1,2,3,4,5} |
| 8        | {1,4}   | 16       | {1,2,3} | 24       | {2,4,5} |           |         |

\*“1” = statistical and information-theory based meta-features; “2” = model structure based meta-features; “3” = Land-marking Based meta-features; “4” = problem complexity based meta-features and “5” = structural information based meta-features.
5.2.1. Comparison on Ranking Loss. Fig. 4 compares the ensemble learning based recommendation model with the models constructed on different combinations of meta-features in terms of Ranking Loss. The smaller the Hamming Loss, the better the corresponding recommendation model. In this figure, a separate box is produced by the “box plot” for each recommendation model according to its Ranking Loss values evaluated on the meta-data. The notch of each box denotes the comparison intervals of the median value of Ranking Loss estimated on the corresponding recommendation model. Two medians are significantly different at the 5% significance level if their intervals do not overlap. And the box marked as “En” denotes the ensemble learning based recommendation model, and the $i$th box denotes the recommendation model constructed on the $i$th combination of meta-features in Table IV. The same representation can be found in Figs. 5 and 6. From Fig. 4 we can observe that:

1. The Ranking Losses of different recommendation models are different. And the differences among some models are significant. This means that the recommended rankings of candidate algorithms vary with different recommendation models. Meanwhile, no matter under which kind of combinations of meta-features in Table IV the recommendation model constructed on the combined meta-features performs equally or better than the single kind of meta-features. The reason is that different kinds of meta-features characterize the classification problems in different viewpoints and will be relatively complemented, so the combinations can give us more comprehensive understanding of the problem. Furthermore, it is possible to construct more precise decision tree model to distinguish the appropriate and inappropriate candidate algorithms.

2. The Ranking Loss of ensemble learning based model (i.e., the last box) is the lowest. And it is significantly smaller than that of any other recommendation model (i.e., any box numbered by 1, 2, …, 31). For the other 31 recommendation models, the smallest/greatest median value of Ranking Loss is 0.2483/0.3047. However, by combining these 31 recommendation models together to form the ensemble recommendation model, the median value of Ranking Loss is only 0.17, and outperforms the best base recommendation model by 31.53%. This means that the proposed ensemble learning method is more effective to estimate the ranking of the candidate algorithms.

5.2.2. Comparison on Precision. Fig. 5 shows the comparison results of different recommendations in terms of average precision. From this figure, we can get that:

1. The Average Precision varies with different recommendation models. For single kind of meta-features corresponding to the first five recommendation models, there exists distinctly significant difference among the Average Precision, such as the Av-
Fig. 5: Comparison among different recommendation models in terms of Average Precision

The average Precision of the models constructed on the statistic and information theory and structural information based meta-features is significantly greater than that of the models constructed on other three kinds of meta-features.

For the models constructed on the different combinations of the five kinds of meta-features, their Average Precisions are either statistically equal to or greater than that of model constructed on the corresponding single kind of meta-features. For example, the 10th box corresponds to the Average Precision of the model constructed on the combinations of 2nd and 3rd meta-features. And its median is statistically greater than that of either the 2nd or 3rd box.

(2) The average precision of ensemble learning based model, which is achieved by integrating the 31 recommendation models together, is the highest and statistically better than that of any of the other 31 recommendation models. For the 31 base recommendation models, the greatest/smallest median value of Average Precision is 0.7097/0.6637. However, by combining these 31 base recommendation models together to form the ensemble recommendation model, the median value of Average Precision can be up to 0.7785, and outperforms the best base recommendation model by 9.69%. This indicates that combining the recommendation models constructed on different sets of meta-features together is an effective way to construct the more accurate recommendation model.

Fig. 6: Comparison among different recommendation models in terms of Precision(1)

Besides the average precision, the user might be interested on the precision of the top ranked algorithm. That is, whether the first recommended algorithm is one of the real appropriate algorithms. This can be measured by the metric Precision(1) of the recommendation model. Fig. 6 shows the Precision(1) of different recommendation
models. From Fig. 6, we can observe that, the Precision of the top ranked algorithm recommended by different recommendation models is different. For the 31 base recommendation models, the greatest/smallest median value of Precision(1) is 0.6697/0.5871. However, by combining these 31 base recommendation models together to form the ensemble recommendation model, the median value of Precision(1) can be up to 0.7615, and outperforms the best base recommendation model by 13.71%.

In summary, no matter in terms of either Average Precision or Precision(1), the proposed ensemble learning based algorithm recommendation method is significantly than the existing recommendation models.

In summary, no matter in terms of either Average Precision or Precision(1), the proposed ensemble learning based algorithm recommendation method is significantly than the existing recommendation models.

![Fig. 7: Sensitive Analysis of Accurate and Diverse Models on Ensemble Recommendation Model](image)

5.2.3. Sensitive Analysis of Ensemble Recommendation Model. The necessary and sufficient condition to construct an accurate ensemble recommendation model is that the base models must be accurate and diverse. This paper gives the methods to identify the accurate and diverse learning models. In this section, we give the sensitive analysis of how the accurate and diverse models act on constructing accurate ensemble recommendation model. Fig. 7 shows the sensitive analyses of these two aspects on the recommendation models in terms of Ranking Loss, Average Precision, Precision(1) and the number of the base models using for ensemble learning, respectively. In this figure, i) “all” represents that all the 31 base recommendation models are used for ensemble model construction; ii) “accurate” denotes the 31 base models are only filtered by the Definition 3.2 of accurate learning model; iii) “diverse” denotes the 31 base models
are only filtered by the Definition 3.3 of diverse learning model and iv) “accurate & diverse” means that the 31 base models are filtered by both Definitions 3.2 and 3.3.

From Fig. 7, we can get that:

(1) For each sub-figure, the four kinds of recommendation models can be grouped into two categories according to the box plots of their performance metrics. The models marked as “all” and “accurate” have statistically equal performance, and the other two models perform statistically equally as well. This is because that during the base recommendation model construction, for most of the single-label learning problems transformed from the multi-labeled meta-data by BR transformation method, the classification accuracy of decision tree on them is greater than 0.5. Therefore, the classification accuracy based filter can just filter out quite a few decision trees. That mean, for recommendation models marked as “all” and “accurate”, the decision trees used for model construction are quite in common. This leads to that the performance of recommendation models marked as “all” and “accurate” are quite similar. Similarly, since the difference among the construction of models marked as “diverse” and “accurate & diverse” derives from the classification accuracy based filter, their performance is similar. This can be also confirmed by the Fig. 7(d) showing the average number of trees used for ensemble learning based algorithm recommendation. The recommendation models marked as “all” and “accurate” are constructed based on almost the same number of decision trees. So the same as the recommendation models marked as “diverse” and “accurate & diverse”.

(2) Since that the most decision trees used for algorithm recommendation model constructed are accurate according to Definition 3.2 of accurate learning model, the proposed ensemble learning based algorithm recommendation method focuses on finding out the diverse base learning models. According to Fig. 7, after filtering out the decision trees by Definition 3.3 of diverse base model, we get better ensemble learning based recommendation models with less number of decision trees. Such as, the model marked as “diverse” with diversity based filter outperforms the model marked as “all” in terms of all the performance metrics Ranking Loss, Average Precision and Precision(1); and the model marked as “accurate & diverse” is better than the model marked as “accurate”. This indicates that $\kappa$ in Eq. 3 is a good choice to evaluate the diversity between different base models and can be used to detect diverse base models for ensemble learning based algorithm recommendation.

In summary, by the sensitive analysis of two important aspects (including accurate and diverse base learning models) in ensemble learning model construction, we can conclude that, for classification algorithm recommendation, the proposed definitions of accurate and diverse learning models are effective to find out a good set of base recommendation models to construct better ensemble recommendation model.

6. CONCLUSIONS

In this paper, we have proposed a novel multi-label ensemble learning based recommendation method with the aim to support the automatic recommendation of appropriate classification algorithms for a new classification problem from a number of candidates.

The proposed method first viewed the algorithm recommendation as a multi-label learning problem due to the fact that there would be multiple algorithms being appropriate for a classification problem. Then, different from the existing recommendation methods which usually construct the recommendation model on only one kind of meta-features by a single learner, we constructed the recommendation model with an ensem-
Ensemble Learning Based Classification Algorithm Recommendation

Able learner which combines a set of base recommendation models constructed on the combinations of different kinds of meta-features by tree-based multi-label learners.

Finally, we have thoroughly tested the recommendation method 1090 benchmark classification problems, 13 different classification algorithms and five different kinds of meta-features. The experimental results show that the proposed ensemble learning based recommendation model is more effective.

APPENDIX: Meta-features

Meta-features are a set of measures extracted from and reflected the properties of a classification problem. These measures map each classification problem into a real-value vector of the domain $X = \mathbb{R}^m$, and are extracted by function $F$.

Meta-feature extraction is one of the most challenging aspects of algorithm recommendation. Theoretically, any measure reflecting the property of a classification problem in a way can be viewed as a meta-feature. Yet, in the filed of algorithm recommendation, the meta-features should be i) related to the performance of the classification algorithms, ii) easy to calculate and iii) calculable for different classification problems.

Aiming at different viewpoints of the classification problems, researchers in algorithm recommendation have put forward several significant data set characterization methods to extract meta-features from a given classification problem. These meta-features can generally grouped into five different categories as follows.

Table V: Statistical and Information-Theory Based Measures

| Measure       | Definition                                                                 |
|---------------|---------------------------------------------------------------------------|
| 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.MinValue  | Proportion of instances with missing values                               |
| Pro.MaxValue  | Proportion of instances with missing values                               |

Table VI: Model Structure Based Measures

| Measure       | Definition                                                                 |
|---------------|---------------------------------------------------------------------------|
| Tree.Height   | Height of tree (also referred as to number of levels in tree)             |
| Tree.Width    | Width of tree                                                             |
| Node.Num      | Number of nodes in tree                                                   |
| Leaf.Num      | Number of leaves in tree                                                  |
| Level.Max     | Maximum number of nodes on the level                                      |
| Level.Min     | Minimum number of nodes on the level                                      |
| Level.Mean    | Mean of the number of nodes on levels                                     |
| Level.Std     | 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 length                                                 |
| Branch.Dev    | Standard deviation of the branch lengths                                  |
| Attr.Max      | Maximum occurrences of attributes                                         |
| Attr.Mean     | Mean of the number of occurrences of attributes                           |
| Attr.Dev      | Standard deviation of the number of occurrences of attributes             |

Table VII: Problem Complexity Based Measures

| Measure       | Definition                                                                 |
|---------------|---------------------------------------------------------------------------|
| Adherence Prep| Proportion of retained adherence                                          |
| Inter.ClassRatio| Ratio of average interclass/interclass distances                           |
| Linear.ClassRatio| Ratio of average interclass/intra class distances                         |
| Linear.Discrim| Linear discriminant                                                       |
| Fisher.Ratio  | Maximum Fisher's discriminant ratio                                        |
| IncAttr       | Training set size related to feature space dimensionality                 |
(1) **Statistical and Information-theory Based Measures**

The statistical and information-theory based measures are the most widely-used in the field of classification algorithm recommendation [Brazdil et al. 2003; Sohn 1999; Henery 1994; Aha 1992; Gama and Brazdil 2000; Engels and Theusinger 1998]. The prominent examples based on these measures are the projects ESPRIT Statlog (1991-1994) and METAL (1998-2001). These measures generally include the data set characteristics such as, number of features, number of instances, number of target concepts, ratio of instances to features, ratio of missing values, ratio of binary features, entropy of the target concept, information gain between the feature and the target concept, and correlation coefficient between features, etc. See Table V for details.

(2) **Model Structure Based Measures**

Firstly, a classification problem is represented in a special data structure embedding the complexity of the problem. Then, the characteristics of the structure are exploited to describe the classification problem.

In the field of algorithm recommendation, the induced decision tree is a well-known and commonly-used structure to model a classification problem. Bensusan [Bensusan 1998] proposed to capture the information from the induced decision tree for describing the classification complexity. He extracted ten measures 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. [Peng et al. 2002] re-analyzed the characterization of decision trees, and proposed some new measures to characterize the structural properties of decision trees. See Table VI for details.

(3) **Landmarking Based Measures**

This kind of measures falls within the concept of landmarking [Pfahringer et al. 2000; Bensusan and Giraud-Carrier 2000; Jain et al. 2000; Duin et al. 2004]. This idea was proposed based on the assumption that the performance of the candidate algorithms could be predicted by the performance of a set of simple classifiers (also called landmarkers). So the performance (e.g., accuracy) of these landmarkers is used to describe a classification problem. Evidently, this kind of measures is closely related to the choice of landmarkers. In practice, it should be ensured that the chosen landmarkers have significant differences in terms of learning mechanism. Following the suggestions in [Bensusan and Giraud-Carrier 2000; Pfahringer et al. 2000], 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 tree, v) a random chosen node tree and vi) the worst node tree. Where the last three classifiers can be achieved based on the well-known classification algorithm C4.5.

(4) **Problem Complexity Based Measures**

The problem complexity based measures focus on the description of the geometrical complexity of the classification problem and emphasize the geometrical characteristics of the distributions of the classes by analyzing the source of difficulty in solving a classification problem [Bernabé-Mansilla and Ho 2005; Ho and Basu 2002; Elizondo et al. 2009; Ho 2000]. The measures reflecting the way in which different classes are separated or interleaved (and being relevant to classification performance) are identified as the measurement of the problem's complexity. Such as Fisher's discriminant ratio, the percentage of instances in the problem that linear the class boundary, and the nonlinearity of linear/non-linear classification algorithm, etc. See Table VII for details.

(5) **Structural Information Based Measures**

Recently, Song et al. proposed a novel data characterization method to facilitate the algorithm recommendation [Song et al. 2012]. The method utilizes structural infor-
mation based feature vectors to characterize the classification problems, which is quite different from the existing ones. Specially, the two feature vectors, one-item feature vector and two-item feature vector, are extracted from a given classification problem. These two vectors consists 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
Aha, D. W. 1992. Generalizing from case studies: A case study. In Proceedings of the Ninth International Conference on Machine Learning. Citeseer, 1–10.
Ali, R., Khatak, A. M., Chow, F., and Lee, S. 2018. A case-based meta-learning and reasoning framework for classifiers selection. In Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication. 1–6.
Ali, S. and Smith, K. A. 2006a. A meta-learning approach to automatic kernel selection for support vector machines. Neurocomputing 70, 1, 173–186.
Ali, S. and Smith, K. A. 2006b. On learning algorithm selection for classification. Applied Soft Computing 6, 2, 119–138.
Arratia, R. and Gordon, L. 1989. Tutorial on large deviations for the binomial distribution. Bulletin of Mathematical Biology 51, 1, 125–131.
Baeza-Yates, R. A. and Ribeiro-Neto, B. A. 1999. Modern information retrieval. ACM press New York.
Bensusan, H. 1998. God doesn’t always shave with oceam’s razor - learning when and how to prune. In Proceedings of the 10th European Conference on Machine Learning. Springer, 119–124.
Bensusan, H. and Giraud-Carrier, C. 2000. Casa batlo is in passeig de gracia or landmarking the expertise space. In Proceedings of the ECML2000 workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination. 29–47.
Bernadó-Mansilla, E. and Ho, T. K. 2005. Domain of competence of XCS classifier system in complexity measurement space. IEEE Transactions on Evolutionary Computation 9, 1, 82–104.
Brazdil, P., Gama, J., and Horey, B. 1994. Characterizing the applicability of classification algorithms using meta-level learning. In Proceedings of European Conference on Machine Learning. Springer, 83–102.
Brazdil, P. and Soares, C. 2000. A comparison of ranking methods for classification algorithm selection. Proceedings of 11th European Conference on Machine Learning, 63–75.
Brazdil, P. B., Soares, C., and Da Costa, J. P. 2003. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning 50, 3, 251–277.
Brodley, C. E. 1993. Addressing the selective superiority problem: Automatic algorithm/model class selection. In Proceedings of the Tenth International Conference on Machine Learning. Citeseer, 17–24.
Brown, G., Wyatt, J., Harris, R., and Yao, X. 2005. Diversity creation methods: a survey and categorization. Information Fusion 6, 1, 5–20.
Cunningham, P. and Carney, J. 2000. Diversity versus quality in classification ensembles based on feature selection. In The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Springer, 109–116.
Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.
Dieterich, T. G. 1997. Machine learning research: Four current directions. AI Magazine 18, 4, 97–136.
Dieterich, T. G. 2000a. Ensemble methods in machine learning. In Multiple Classifier Systems. Springer, 1–15.
Dieterich, T. G. 2000b. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning 40, 2, 139–157.
Dieterich, T. G. 2002. Ensemble learning. The Handbook of Brain Theory and Neural Networks, 405–408.
Duin, R. P. W., Pekalska, E., and Tax, D. M. J. 2004. The characterization of classification problems by classifier disagreements. In Proceedings of the 17th International Conference on Pattern Recognition. Vol. 1. IEEE, 140–143.
Džeroski, S. and Ženko, B. 2004. Is combining classifiers with stacking better than selecting the best one? Machine Learning 54, 3, 255–273.
ELIZONDO, D. A., BIRKENHEAD, R., GAMEZ, M., GARCIA, N., AND ALFARO, E. 2009. Estimation of classification complexity. In Proceedings of International Joint Conference on Neural Networks. IEEE, 764–770.

ENGELS, R. AND THEUSINGER, C. 1998. Using a data metric for preprocessing advice for data mining applications. In Proceedings of the European Conference on Artificial Intelligence. 430–434.

G. TSOUKAKIS, I. V. 2007. Random k-labelsets: An ensemble method for multilabel classification. In 18th European Conference on Machine Learning (ECML 2007). Springer Berlin Heidelberg, 406–417.

GAMA, J. AND BRAZDIL, P. 2000. Cascade generalization. Machine Learning 41, 3, 315–343.

Giacinto, G. AND Roli, F. 2001. Design of effective neural network ensembles for image classification processes. Image and Vision Computing 19, 699–707.

HALABI ECHEVERRY, A., RICHARDS, D., AND BILGIN, A. 2012. Identifying characteristics of seaports for environmental benchmarks based on meta-learning. Knowledge Management and Acquisition for Intelligent Systems, 350–363.

HANSEN, L. K. AND SALAMON, P. 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001.

HENERY, R. J. 1994. Methods for comparison. Ellis Horwood, 107–124.

Ho, T. K. 1998. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 8, 832–844.

Ho, T. K. 2000. Complexity of classification problems and comparative advantages of combined classifiers. Multiple Classifier Systems, 97–106.

Ho, T. K. AND BASU, M. 2002. Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 3, 289–300.

JAIN, A. K., DUNN, R. P. W., AND MAO, J. 2000. Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1, 4–37.

KALOUSIS, A. 2002. Algorithm selection via meta-learning. Ph.D. thesis.

KALOUSIS, A., GAMA, J., AND HILARIO, M. 2004. On data and algorithms: Understanding inductive performance. Machine Learning 54, 3, 275–312.

KHAN, I., ZHANG, X., REHMAN, M., AND ALI, R. 2020. A literature survey and empirical study of meta-learning for classifier selection. IEEE Access 8, 10262–10281.

KING, R. D., FENG, C., AND SUTHERLAND, A. 1995. Statlog: comparison of classification algorithms on large real-world problems. Applied Artificial Intelligence an International Journal 9, 3, 289–333.

KOHAVI, R., WOLPERT, D. H., ET AL. 1996. Bias plus variance decomposition for zero-one loss functions. In International Conference on Machine Learning. 275–283.

KUNCEVA, L. I. AND WHITAKER, C. J. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 51, 2, 181–207.

KUNCEVA, L. I., WHITAKER, C. J., SHIPP, C. A., AND DUNN, R. P. W. 2003. Limits on the majority vote accuracy in classifier fusion. Pattern Analysis and Applications 6, 22–31.

LAM, L. 2000. Classifier combinations: implementations and theoretical issues. In Multiple Classifier Systems. Springer, 77–86.

LEE, J. W. AND GIRAUD-CARRIER, C. 2013. Automatic selection of classification learning algorithms for data mining practitioners. Intelligent Data Analysis 17, 4, 665–678.

LEE, M., LU, H., LING, T., AND KO, Y. 1999. Cleansing data for mining and warehousing. In Proceedings of the 10th International Conference on Database and Expert Systems Applications. Springer, 751–760.

NASIERDING, G., KOUZANI, A. Z., AND TSOUKAKIS, G. 2010. A Triple-Random Ensemble Classification Method for Mining Multi-label Data. In IEEE International Conference on Data Mining. 49–56.

PENG, Y., FLACH, P., SOARES, C., AND BRAZDIL, P. 2002. Improved dataset characterisation for meta-learning. In Discovery Science. Springer, 193–208.

PETERTON, A. H. AND MARTINEZ, T. 2005. Estimating the potential for combining learning models. In Proceedings of the ICML workshop on meta-learning, 68–75.

PFahlringer, B., BENSUSAN, H., AND GIRAUD-CARRIER, C. 2000. Meta-learning by landmarking various learning algorithms. In Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann, 743–750.

PIZARRO, J., GUERRERO, E., AND GALINDO, P. L. 2002. Multiple comparison procedures applied to model selection. Neurocomputing 48, 155–173.

PRUDÊNCIO, R., DE SOUTO, M., AND LUDERMEIR, T. 2011. Selecting machine learning algorithms using the ranking meta-learning approach. Meta-Learning in Computational Intelligence, 225–243.

PRUDÊNCIO, R., SOARES, C., AND LUDERMEIR, T. 2011. Combining meta-learning and active selection of datasetoids for algorithm selection. Hybrid Artificial Intelligent Systems, 164–171.
Ensemble Learning Based Classification Algorithm Recommendation

Prudencio, R. B. C., Soares, C., and Ludermir, T. B. 2011. Uncertainty sampling methods for selecting datasets in active meta-learning. In The 2011 International Joint Conference on Neural Networks (IJCNN). IEEE, 1082–1089.

Read, J., Pfahringer, B., and Holmes, G. 2008. Multi-label Classification Using Ensembles of Pruned Sets. In IEEE International Conference on Data Mining. 995–1000.

Rice, J. R. 1976. The Algorithm Selection Problem. Advances in Computers 15, 65–118.

Rokach, L., Schclar, A., and Itach, E. 2014. Ensemble methods for multi-label classification. Expert Systems with Applications 41, 16, 7507–7523.

Shi, C., Kong, X., Yu, P., and Wang, B. 2011. Multi-label ensemble learning. In Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 223–239.

Smith-Miles, K. A. 2008. Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Computing Surveys 41, 1, 1–25.

Soares, C. 2009. UCI++: Improved support for algorithm selection using datasetoids. In Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 499–506.

Sohn, S. Y. 1999. Meta analysis of classification algorithms for pattern recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 11, 1137–1144.

Song, Q. B., Wang, G. T., and Wang, C. 2012. Automatic recommendation of classification algorithm based on data set characteristics. Pattern Recognition 45, 7, 2672–2689.

Tsoumakas, G., Katakis, I., and Vlahavas, I. 2010. Mining multi-label data. In Data Mining and Knowledge Discovery Handbook. Springer, 667–685.

Tumer, K. and Ghosh, J. 1996. Error Correlation and Error Reduction in Ensemble Classifiers. Connection Science 8, 385–404.

Tumer, K. and Ghosh, J. 1999. Linear and Order Statistics Combiners for Pattern Classification. Computing Research Repository cs.NE/9905.

Wang, G. T., Song, Q. B., Zhang, X. Y., and Zhang, K. Y. 2014. A generic multilabel learning-based classification algorithm recommendation method. ACM Transactions on Knowledge Discovery from Data, 9, 1, 1–31.

Wolfert, D. H. 2001. The supervised learning no-free-lunch theorems. In Proceedings of 6th Online World Conference on Soft Computing in Industrial Applications. Citeseer, 25–42.

Yang, J. and Ji, B. 2006. Algorithm selection: A quantitative approach. Algorithmic Trading II: Precision, Control, Execution. Institutional Investor Inc, 26–34.

Zhang, M.-L. 2009. Mi-كف: Rbf neural networks for multi-label learning. Neural Processing Letters 29, 2, 61–74.

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

Zhang, Min-Ling, Z. Z.-H. 2006. Multilabel neural networks with applications to functional genomics and text. IEEE Transactions on Knowledge and Data Engineering, 1338–1351.

Zhu, X., Li, Y., Wang, J., Zheng, T., and Fu, J. 2020. Automatic recommendation of a distance measure for clustering algorithms. ACM Transactions on Knowledge Discovery from Data (TKDD) 15, 1, 1–22.