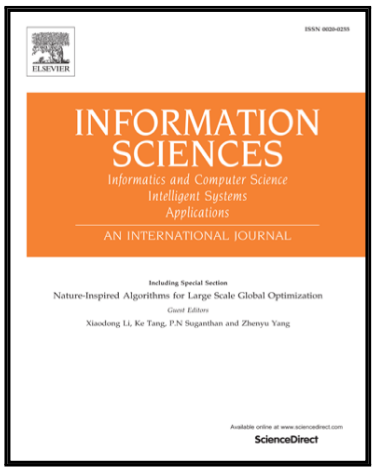

A New Data Characterization for Selecting Clustering Algorithms Using Meta-Learning

Bruno Almeida Pimentel, André C.P.L.F. de Carvalho

PII: S0020-0255(18)30862-4
DOI: https://doi.org/10.1016/j.ins.2018.10.043
Reference: INS 14024

To appear in: *Information Sciences*

Received date: 15 March 2018
Revised date: 24 October 2018
Accepted date: 25 October 2018

Please cite this article as: Bruno Almeida Pimentel, André C.P.L.F. de Carvalho, A New Data Characterization for Selecting Clustering Algorithms Using Meta-Learning, *Information Sciences* (2018), doi: https://doi.org/10.1016/j.ins.2018.10.043

A New Data Characterization for Selecting Clustering Algorithms Using Meta-Learning

Bruno Almeida Pimentel[a], André C. P. L. F. de Carvalho[a]

[a]*Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP), São Carlos, Brazil*

Abstract

Meta-learning has been successfully used for algorithm recommendation tasks. It uses machine learning to induce meta-models able to predict the best algorithms for a new dataset. In this paper, meta-models are applied to a set of meta-features, describing a dataset, to predict the performance of clustering algorithms applied to this dataset. The paper also proposes a new set of meta-features, based on correlation and dissimilarity measures. Experimental results show that these meta-features improve the recommendation. Additionally, this paper evaluates the importance of each meta-feature for the recommendation.

Keywords: Data Characterization, Clustering, Meta-learning

1. Introduction

One of the main Machine Learning (ML) applications is data clustering [15]. With the growing interest in understanding, processing and summarizing data automatically, clustering algorithms have been successfully applied

Email addresses: bapimentel@icmc.usp.br (Bruno Almeida Pimentel), andre@icmc.usp.br (André C. P. L. F. de Carvalho)

to various application domains, such as anomaly detection, gene expression analysis, community detection and object segmentation [24]. Several clustering algorithms have been proposed in the literature.

Selecting a suitable algorithm to deal with a given ML task is fundamental to obtain a model with a good predictive or descriptive performance [41]. This is due to the fact that each dataset has its inherent characteristics [21]. Each algorithm tries to model and solve a problem by extracting information about these characteristics [4], which leads researchers to investigate a large number of algorithms. The selection of the most suitable algorithms among this large number of them is usually based on empirical observation or user's previous experiences, which can be subjective and have a high computational cost [43].

This difficulty has been overcome by the automatic recommendation of ML algorithm(s) for a new dataset. This research approach, known as meta-learning or autoML, investigates the use of ML algorithms for the induction of predictive models able to recommend the most suitable algorithm for a new dataset [44, 4]. Recently, a large number of research groups have investigated new approaches to cover the different aspects involved in meta-learning and the use of meta-learning for recommending techniques for data classification [47, 42], time series analysis [35, 30], optimization [28, 12], noise detection [16, 17], instance selection [31], recommender systems [11, 6] and ML algorithm hyperparameter tuning [32, 22].

Data characteristics (or meta-features) are usually extracted by three main data characterization approaches [18]: Statistics, Landmarking or Model-based properties. As the recommended ranking is based on dataset features,

the characterization of datasets is of crucial importance for the successful use of meta-learning. Most ML algorithm recommendation studies using meta-learning deal with classification tasks. In these tasks, the meta-target is the predictive performance of classification algorithms.

As in classification tasks, there are many clustering algorithms and their bias make some more suitable for particular datasets than others. The bias of ML algorithms is usually related with how they search for a model (search bias) and how they represent the model (representation bias). For a good performance in a particular dataset, the bias of an algorithm should match well with the data distribution in the dataset. Thus, the automatic selection of those most suitable among a set of ML algorithms can select algorithms whose bias are more appropriate for particular datasets. This is the main goal of algorithm recommender systems using meta-learning.

The remainder of this paper is organized as follows. Section 2 briefly explains the basic aspects of meta-learning and some of its techniques. Section 3 addresses related works of data characterization for clustering algorithms recommendation and describes the meta-features presented in [8], [13] and [46]. Section 4 shows the main contribution from this work regarding data characterization for selecting clustering algorithms. A comparative study using the proposed data characterization, two baseline recommendation methods, statistical-based meta-features [8], distance-based meta-features [13] and evaluation-based meta-features [46] is presented in Section 5. In order to analyze the importance of the investigated meta-features to recommend clustering algorithms, Section 6 shows the application of the Random Forest algorithm to identify the most relevant meta-features. In Section 7, final

remarks are provided and future work directions are pointed out.

2. Meta-learning

This section briefly explains the main aspects of meta-learning and how it can be used for algorithm recommendations. According to Brazdil et al. (2008) [4], the goal of algorithm recommendation is to save time by reducing the number of algorithms investigated, without a significant reduction in predictive performance. For algorithm recommendations, it is not important to accurately predict the true performance of all considered algorithms, but rather to predict their relative performance. One of the main approaches for algorithm recommendation is the use of meta-learning [4].

Meta-learning investigates the induction of meta-models able to recommend the most suitable algorithm(s) for a new dataset. For such, ML algorithms are applied to a meta-dataset. Similar to a ML dataset, which is formed by a set of instances, a meta-dataset has a set of meta-instances. The predictive attributes of a meta-instance are defined by meta-features. Meta-features are information extracted from the conventional datasets, for example the number of instances in the dataset. The values of the meta-features for a dataset characterize the dataset. The target attributes for a meta-instance are the performances obtained by a set of algorithms when applied to the dataset corresponding to the meta-instance.

After creating the meta-dataset, a ML algorithm, called *meta-learner*, can be applied to the meta-dataset, as in conventional ML, to induce a predictive *meta-model*. This task can be either a simple classification task (where each class is one ML algorithm), or a ranking classification task (where the target

is a ranking of ML algorithms).

An ML algorithm often used to induce a meta-model for ranking prediction is an adapted version of the k-Nearest Neighbors algorithm (k-NN) [14]. As an example, the recommendation of the ranking of the most suitable ML algorithms for a new dataset occurs by applying the adapted k-NN to the vector of meta-feature values extracted from the new dataset. This returns the k meta-instances from the meta-dataset that are mostly similar to this vector. The ranking of algorithms of the returned meta-instances are aggregated using the *Average Ranking* (AR) method [4]. The AR method works as follows. Let $\mathbf{r}_j = (r_{1,j}, \ldots, r_{i,j}, \ldots, r_{k,j})$ be the rank position of algorithm j ($j = 1, \ldots, a$) for k datasets, where a is the number of algorithms. The average rank position $\overline{r}_j$ for an algorithm j is given by:

$$\overline{r}_j = \frac{\sum_{i=1}^{k} r_{i,j}}{k} \tag{1}$$

The recommended ranking is obtained by reassigning the rank positions, with the lowest value of $\overline{r}_j$ in the first position and the highest value in the last position. Fig. 1 illustrates how meta-learning with ranking recommendation by Average Ranking works.

In this study, meta-learning is used to recommend the ranking of the most suitable clustering algorithms for new datasets. As meta-features are crucial for the recommendation process, this work proposes meta-features able to collect more information from data, allowing the recommender to improve its performance regarding existing approaches. Experiments were carried out to evaluate the predictive performance of the recommender system when using the proposed meta-features (described in Section 4) and meta-features

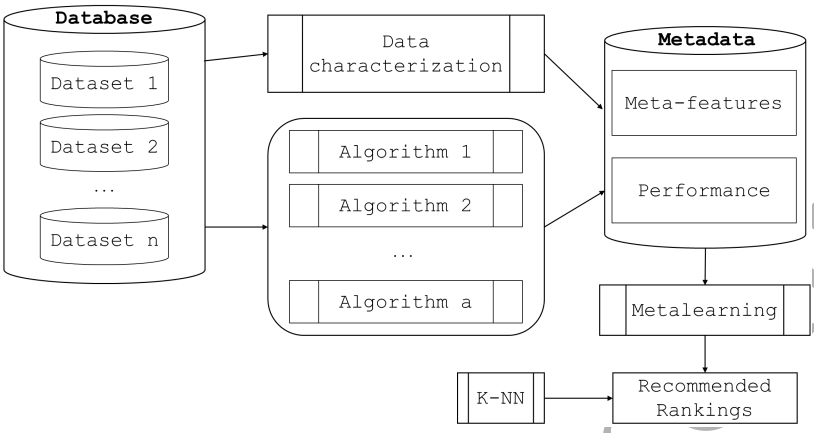

Figure 1: Meta-learning with ranking recommendation [4].

found in the literature (described in Sections 3.1, 3.2 and 3.3).

3. Related Works

There are not as many studies proposing meta-features for the recommendation of clustering algorithms as there are for classification algorithms. The authors found four studies proposing meta-features for clustering algorithm recommendations, which are discussed next.

The first study, Souto et al. (2008) [8] proposed 8 meta-features, based on statistical measures. These meta-features were used to recommend 1 out of 7 clustering algorithms. The meta-dataset was created using 32 datasets related to cancer diagnosis using gene expressions obtained from microarrays. Later, Ferrari and de Castro (2015) [13] proposed a new set of meta-features based on the distance between instances. They also proposed the combination of internal index values to rank a set of selected clustering algorithms. They used 84 datasets from the UCI repository in their experiments. Adam

and Blockeel (2015) [1] introduced meta-features based on measures of must-link and cannot-link constraints, which can quantify the overlapping present in a dataset. The authors showed the benefits of using these constraints for clustering algorithm recommendations. The experiments used 22 synthetic datasets and 14 datasets from the UCI repository. Afterwards, Vukicevic et al. (2016) [46] proposed meta-features based on statistical and internal clustering evaluation measures. The recommender system used these measures to rank the most suitable algorithms for new datasets. The authors also used 32 microarray gene expression datasets in the experiments. It is important to highlight that these other studies did not compare the proposed meta-features with meta-features proposed in other studies.

3.1. Meta-features based on Statistical Measures

The 8 meta-features proposed in De Souto et al. [8] are based on descriptive measures. They were extracted from 32 datasets. The name and a brief description of each meta-feature are presented in Table 1.

The indication of the microarray technology by the "Chip" meta-feature captures relevant information in the context of gene expression microarray datasets. Moreover, the "PFA" meta-feature is related to the application of attribute selection to remove uninformative genes for the case of Affymetrix arrays [8]. Since the datasets used in this study deal with a larger variety of application domains, the meta-features "Chip" and "PFA" are not used. Using these meta-features, meta-learning can recommend, for a new dataset, the most suitable clustering algorithm from a set of 7 clustering algorithms.

Table 1: Statistical-based meta-features and their respective description.

Meta-feature	**Description**
LgE	A function of the dataset size. It uses equation $LgE = log_{10}(n)$, where n is the number of instances.
LgREA	Estimates the ratio between the number of instances (n) and the number of attributes (p). It is computed as $LgREA = log_{10}(\frac{n}{p})$.
PMV	Measures the percentage of missing values.
MN	Related to multivariate normality. For its estimation, the instances are initially transformed into values of Hotelling's T^2 statistics [26].
SK	It returns the skewness of the T^2 vector.
Chip	Indicates the type of microarray technology used: cDNA or Affymetrix.
PFA	Measures the percentage of attributes that were kept after application of the attribute selection filter.
PO	Percentage of outliers using the T^2 vector. Values of T^2 more distant than two standard deviations from the mean are considered outliers.

3.2. Meta-features based on Distance Measures

This set of 19 meta-features was proposed in Ferrari and De Castro [13]. To understand how they work, consider the following definitions. Let $\Omega = \{1, \ldots, k, \ldots, n\}$ be a set of n instances indexed by k. Each instance k is represented by a vector of quantitative attributes $\mathbf{x}_k = (x_{k,1}, \ldots, x_{k,j}, \ldots, x_{k,p})$ described by p attributes indexed by j where $x_{k,j} \in \Re$.

Initially, it is necessary to compute the Euclidean distance between the instances in the dataset. The distance measured between instances $\mathbf{x}_k$ and $\mathbf{x}_l$ is given by Equation 2:

$$d(\mathbf{x}_k, \mathbf{x}_l) = d_{k,l} = \sqrt{\sum_{j=1}^{p} (x_{k,j} - x_{l,j})^2}. \quad (2)$$

Based on this measure, a vector $\mathbf{d}$, containing the dissimilarity among all instances, is built as follows:

$$\mathbf{d} = [d_{1,2}, d_{1,3}, \ldots, d_{k,l}, \ldots, d_{n-2,n-1}, d_{n-1,n}]. \quad (3)$$

Next, vector $\mathbf{d}$ is normalized in the interval $[0, 1]$, generating a new vector **m'**. Given a value from vector $\mathbf{m}$ indexed by v ($\mathbf{m}[v]$), the corresponding normalized value in vector **m'** for index v (**m'**$[v]$) is given by:

$$\mathbf{m'}[v] = \frac{\mathbf{m}[v] - min(\mathbf{m})}{max(\mathbf{m}) - min(\mathbf{m})}. \quad (4)$$

After obtaining the vector **m'**, the 19 meta-features are extracted from each dataset. Table 2 describes these meta-features (MF_1 to MF_{19}).

According to Kalousis (2002) [27], histograms may provide more information about the data being characterized. In particular, MF_1 to MF_5 extract simple statistical data (measures, variance, standard deviation, skewness, and kurtosis) from vector **m'**. Meta-features MF_6 to MF_{15} capture histogram-based information from vector **m'**. Since the values from vector **m'** are normalized, this histogram ranges in the interval $[0.0, 1.0]$. Meta-features from MF_{16} to MF_{19} are extracted from the histogram of the absolute Z-score.

3.3. Meta-features based on Evaluation Measures

In [46], the authors proposed 19 meta-features based on the characteristics (structure) of clustering algorithms and internal evaluation measures. These

Table 2: Distance-based meta-features and their respective description.

Meta-feature	**Description**
MF_1	Mean of **m'**
MF_2	Variance of **m'**
MF_3	Standard deviation of **m'**
MF_4	Skewness of **m'**
MF_5	Kurtosis of **m'**
MF_6	% of values in the interval $[0, 0.1]$
MF_7	% of values in the interval $(0.1, 0.2]$
MF_8	% of values in the interval $(0.2, 0.3]$
MF_9	% of values in the interval $(0.3, 0.4]$
MF_{10}	% of values in the interval $(0.4, 0.5]$
MF_{11}	% of values in the interval $(0.5, 0.6]$
MF_{12}	% of values in the interval $(0.6, 0.7]$
MF_{13}	% of values in the interval $(0.7, 0.8]$
MF_{14}	% of values in the interval $(0.8, 0.9]$
MF_{15}	% of values in the interval $(0.9, 1.0]$
MF_{16}	% of values with absolute Z-score in the interval $[0, 1)$
MF_{17}	% of values with absolute Z-score in the interval $[1, 2)$
MF_{18}	% of values with absolute Z-score in the interval $[2, 3)$
MF_{19}	% of values with absolute Z-score in the interval $[3, \infty)$

meta-features extend those from [8], where from these 19 meta-features, 8 were statistical measures proposed in [8], 5 were proposed in [34] and 6 are internal evaluation measures. The name and a brief description of each meta-feature are presented as follows.

As in [8], 2 other meta-features, "Chip" and "PFA", related to gene

Table 3: Evaluation-based meta-features and their respective description.

Meta-feature	**Description**
1 - 8	See Section 3.1.
NRE	Normalized relative entropy. An indicator of uniformity distributed of instances among clusters.
SC10	Number of clusters with size smaller than 10.
SC15	Number of clusters with size smaller than 15.
BC	Number of clusters with size larger than 50.
k-NN outliers	Classification error obtained by the k-NN algorithm ($k = 3$).
CM	Sum of distances of items to corresponding cluster representatives.
SI	Global silhouette index. It checks whether the current cluster of every instance is more appropriate than the neighboring cluster.
AIC	Internal cluster evaluation measure.
BIC	Abbreviation from Bayesian Information Criterion, this measure is based on maximized value of the likelihood function of the model.
XB	Ratio of overall deviation to cluster separation.
CN	Measures whether neighboring items are in the same cluster.

expression domain were used in [46]. Since they are domain-specific, they were not used here.

Despite the novelty, relevance and good results found in these studies, the authors believe that there is room for improvement. The first aspect to be improved concerns the low number of datasets used in the experiments. A low number of datasets produces a small meta-dataset, making it difficult to obtain meta-learners with a good generalization. Furthermore, the set of recommended clustering algorithms does not represent the wide variety of clustering approaches, which may limit the benefits of the experimental

results. Besides, more robust results could be obtained using a larger number of internal validation measures to evaluate the performance of the clustering algorithms. Another important point, which has not been well explored in the literature, is the interpretation of results regarding the goal of recommender systems: proposing the most suitable clustering algorithm(s).

Table 4 summarizes the main characteristics found in the related works and how this paper goes one step forward.

Table 4: Comparison of evaluation methodology in related works.

Works	**Datasets**	**Clustering algorithms**	**Internal val. measures**	**External val. measures**	**Meta-learners**
De Souto et al. [8]	30	7	0	1	1
Adam and Blockeel [1]	36	2	0	2	1
Ferrari and De Castro [13]	84	10	7	1	1
Vukicevic et al. [46]	30	7	0	1	5
This paper	219	10	10	3	2

4. Proposed Approach

This work proposes a new approach to address the deficiencies mentioned in the previous section. This new approach uses a new set of meta-features. This set combines correlation and dissimilarity. As a result, it includes positive aspects of both measures. A dissimilarity measure is based on the distance between instances, but does not take into account the behavior among the values of attributes. This set of meta-features does not use any information about the class labels of the instances to extract meta-features.

The main novelty of this work is the correlation measure. This type of measure collects statistical relationships between two instances: the more similar the behavior among the attributes, the stronger the relationship with each other [3]. Therefore, the motivation to use both correlation and dissimilarity measures is to collect more information from the data, thus improving the performance of the recommender system.

Additionally, different from other correlation measures, such as Pearson, the Spearman's rank correlation coefficient is robust to the presence of outliers [9]. In order to analyze how the proposed meta-features affect the predictive performance of the meta-learning, this work carried out an extensive comparison between the proposed data characterization measures and measures proposed in the clustering algorithm recommendation literature. For such, two ranking learning techniques, a number of datasets larger than those used in previous studies, more clustering algorithms and internal and external validation measures were used in the experiments.

The dissimilarity measure is calculated using the Euclidean distance, as described in the previous section. The correlation measure is calculated using the Spearman's rank correlation coefficient. To formally define the proposal, consider the following definition. Let $\mathbf{r}_k = (r_{k,1}, \ldots, r_{k,i}, \ldots, r_{k,p})$ be the ranking of instance $\mathbf{x}_k$.

The correlation between instances $\mathbf{x}_k$ and $\mathbf{x}_l$, using the Spearman's rank correlation, is given by the following in Equation 5.

$$c(\mathbf{x}_k, \mathbf{x}_l) = c_{k,l} = 1 - \frac{6 \sum_{i=1}^{p} (r_{k,i} - r_{l,i})^2}{p^3 - p} \tag{5}$$

where p is the number of attributes. This equation returns a vector $\mathbf{c}$ con-

taining the correlation among all instances, which is shown in Equation 6.

$$\mathbf{c} = [c_{1,2}, c_{1,3}, \ldots, c_{k,l}, \ldots, c_{n-2,n-1}, c_{n-1,n}]. \quad (6)$$

The dissimilarity between instances $\mathbf{x}_k$ and $\mathbf{x}_l$ is calculated by Equation 2. Both vectors $\mathbf{c}$ and $\mathbf{d}$ have a size equal to $n(n-1)/2$, where n is the number of instances. The vectors $\mathbf{c}$ and $\mathbf{d}$ are concatenated to create a new vector $\mathbf{m}$, of size $n(n-1)$, with both types of measures.

$$\mathbf{m} = [\mathbf{c}, \mathbf{d}]. \quad (7)$$

To reduce the influence of the different ranges of values of the meta-features, vector $\mathbf{m}$ is normalized in the interval $[0, 1]$, generating a new vector **m'**, as shown in Equation 4. After obtaining vector **m'**, the meta-features proposed in [13], described in Table 2, are used. Since meta-features are calculated from the vector with correlation and dissimilarity values, different measures of correlation and dissimilarity can change the characterization of a dataset and meta-features values extracted. The following algorithm summarizes how the meta-features are computed. Algorithm 1 shows how the meta-features are extracted.

The proposed meta-features are experimentally compared with the meta-features proposed in De Souto et al. [8] (based on statistics), in Ferrari and De Castro [13] (based on the distance between instances) and in Vukicevic et al. [46] (based on evaluation measures).

Algorithm 1 Meta-feature extraction

Require: Dataset D with n instances

Let **m**, **m'**, **c** and **d** be empty vectors
$i = 1$
for $1 \leq k \leq n$ **do**
    for $k < l \leq n$ **do**
        $\mathbf{c}[i] = c_{k,l}$ (Equation 5)
        $\mathbf{d}[i] = d_{k,l}$ (Equation 2)
        $i = i + 1$
    end for
end for
Aggregate **c** and **d** into vector **m**.
for $1 \leq v \leq n(n-1)$ **do**
    $\mathbf{m'}[v] = \frac{\mathbf{m}[v] - min(\mathbf{m})}{max(\mathbf{m}) - min(\mathbf{m})}$
end for
Compute meta-features (MF_1 to MF_{19}) based on **m'** (Table 2)

return Meta-features.

5. Experiments

This section describes the methodology used in the experiments and the evaluation of the proposed set of meta-features. Initially, the datasets, internal indices, clustering algorithms, baseline methods and evaluation measures used in the experiments are described. Afterwards, the experimental results are presented and discussed.

5.1. Datasets

The experiments used 219 datasets collected from Open Machine Learning (OpenML), an online ML platform that allows access to a large number of datasets and sharing ML experiments. These datasets cover different domains, such as engineering, biology, medicine, physics, robotics. These datasets can be found in `https://www.openml.org/s/88/data`. Thus, unlike the datasets used in the previous studies, the datasets used here cover a wider variety of application domains. For each dataset, attributes were normalized, as described in Equation 4, to the interval $[0, 1]$. Table 5 describes all datasets according to number of classes, attributes and instances using the minimal, maximal and average values (standard deviation in parenthesis).

Four different meta-datasets were used in the experiments. They differ in the set of meta-features used. Three of them were those used in the previous works, called here Statistical, Distance and Evaluation. The fourth is the dataset proposed here, namely CaD. In all these meta-datasets, the target attribute was the same, which was a ranking of 10 clustering algorithms.

The experiments evaluate the predictive performance of meta-learning in the recommendation of a ranking of the most suitable clustering algorithms

Table 5: Description of all datasets according to number of classes, attributes and instances.

	Classes	**Attributes**	**Instances**
Min	2	2	100
Max	250	168	10^6
Avg. (Std.)	5.8499 (22.1540)	19.9636 (23.9922)	4799.8545 (67249.5303)

for a new dataset. The rank position of each clustering algorithm is based on the quality of the partitions created. The better the quality of the partition found by the clustering algorithm, the lower its ranking position.

For each meta-dataset, ranking classification algorithms can be applied to induce a meta-model, which can be later used to predict the ranking of clustering algorithms for new datasets. The next section describes how the clustering algorithms are ranked.

5.2. Internal Indices

Since clustering is an unsupervised task, 10 internal indices from different approaches were used to assess the quality of the partitions. Table 6 shows its name, interval and objective for each index.

Initially, a ranking is obtained for each internal index and dataset. Each ranking is computed using the average ranking of all 10 rankings. The average ranking is obtained by calculating the mean position for each algorithm on all indices and then reassigning the rank positions.

Table 6: Interval indices including their intervals and objectives.

Index	**Interval**	**Objective**
Calinski-Harabasz [5]	$[0; +\infty)$	Max
Silhouette [38]	$[-1; +1]$	Max
Dunn [10]	$[0; +\infty)$	Max
Gamma [2]	$[-1; +1]$	Max
Tau [19]	$[0; +\infty)$	Max
Davies-Bouldin [7]	$[0; +\infty)$	Min
Xie-Beni [48]	$[0; +\infty)$	Min
SD-Scat [20]	$[0; +\infty)$	Min
SD-Dis [20]	$[0; +\infty)$	Min
Ray-Turi [37]	$[0; +\infty)$	Min

5.3. Evaluation Methodology

In order to assess the predictive performance of the recommender system, two rankings are compared: the true ranking and the ranking predicted by a meta-model. To evaluate the similarity between the true and the recommended rankings, the Spearman's rank correlation coefficient (SRC) is used. For such, let $\boldsymbol{r} = (r_1, \ldots, r_i, \ldots, r_a)$ and $\boldsymbol{s} = (s_1, \ldots, s_i, \ldots, s_a)$ be, respectively, the true and the recommended rankings, where a is the number of candidate algorithms. The Spearman's rank correlation coefficient is calculated using the following equation:

$$SRC(\boldsymbol{r}, \boldsymbol{s}) = 1 - \frac{6 \sum_{i=1}^{a} (r_i - s_i)^2}{a^3 - a} \tag{8}$$

The value of this coefficient ranges from $[-1; +1]$, whereby the larger the value of SRC, the higher the similarity between the true and the rec-

ommended rankings [39]. This coefficient has been often used for ranking comparisons in meta-learning [4, 8, 13].

Additionally, the most commonly used clustering evaluation index is measured: the Adjusted Rand's Index (ARI) [23]. This index falls within the $[-1, +1]$ interval, where 1 represents a perfect correspondence between the *a priori* partition and the partition obtained by a clustering algorithm. Values near 0 indicate an approximately random solution and negative values determine an insufficient ability of a clustering algorithm to find good partitions [36]. Let $P = \{p_1, \ldots, p_i, \ldots, p_C\}$ be a partition with C clusters obtained after running a clustering algorithm and $Q = \{q_1, \ldots, q_j, \ldots, q_D\}$ an *a priori* partition with D clusters. Equation 9 shows how the ARI value is defined.

$$ARI = \frac{\sum_{i=1}^{C}\sum_{j=1}^{D}\binom{n_{ij}}{2} - \binom{n}{2}^{-1}\sum_{i=1}^{C}\binom{n_{i\cdot}}{2}\sum_{j=1}^{D}\binom{n_{\cdot j}}{2}}{\frac{1}{2}\left[\sum_{i=1}^{C}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{D}\binom{n_{\cdot j}}{2}\right] - \binom{n}{2}^{-1}\sum_{i=1}^{C}\binom{n_{i\cdot}}{2}\sum_{j=1}^{D}\binom{n_{\cdot j}}{2}}, \quad (9)$$

where n_{ij} is the number of instances that belong to groups p_i and q_j, $n_{i\cdot}$ is the number of instances that belong to group p_i, $n_{\cdot j}$ is the number of instances that belong to group q_j and n is the total number of instances. In this work, partition P is obtained by a clustering algorithm whose rank is equal to 1.

To confirm the ARI evaluation, another clustering evaluation index, the Adjusted Mutual Information (AMI) [45] is also used. This measure falls within the $[0, 1]$ interval, where 1 represents a perfect correspondence between the *a priori* partition and the partition obtained by a clustering algorithm and 0 represents an insufficient ability of a clustering algorithm to find good partitions. To formally define the AMI index, let $H(P) = -\sum_{i=1}^{C}\frac{n_{i\cdot}}{n}log\frac{n_{i\cdot}}{n}$ be the average amount of information of P (similarly given for $H(Q)$) and

$I(P,Q) = \sum_{i=1}^{C}\sum_{j=1}^{D}\frac{n_{ij}}{n}log\frac{n_{ij}/n}{n_{i.}n_{.j}/n^2}$ be the mutual information between P and Q. The value of AMI is defined using Equation 10.

$$AMI = \frac{I(P,Q) - E\{I(P,Q)\}}{max\{H(P),H(Q)\} - E\{I(P,Q)\}} \tag{10}$$

where $E\{I(P,Q)\}$ is given by:

$$E\{I(P,Q)\} = \sum_{i=1}^{C}\sum_{j=1}^{D}\sum_{n_{ij}=max(n_{i.}+n_{.j}-n,0)}^{min(n_{i.},n_{.j})}\frac{n_{ij}}{n}log\left(\frac{nn_{ij}}{n_{i.}n_{.j}}\right)\times \frac{n_{i.}!\,n_{.j}!\,(n-n_{i.})!\,(n-n_{.j})!}{n!\,n_{ij}!\,(n_{i.}-n_{ij})!\,(n_{.j}-n_{ij})!\,(n-n_{i.}-n_{.j}+n_{ij})!} \tag{11}$$

The recommendation process was evaluated using a 10-fold cross-validation. Given a fold, a dataset is selected to be the test set and the remaining 9 folds are used as training set. After selecting all the datasets in a fold, the average of SRC, ARI and AMI values obtained for 10 runs are calculated. This process is repeated 30 times, with the 10 folds randomly defined at each time. Afterwards, the mean and standard deviation of the 30 SRC, ARI and AMI values are computed.

5.4. Clustering Algorithms

In order to validate the proposed method, 10 widely used clustering algorithms with different learning bias were selected: Average Agglomerative Clustering (AA), Complete Agglomerative Clustering (CA), Fuzzy C-Means (FCM), Gaussian Mixture with diagonal matrix (GMd), Gaussian Mixture with full matrix (GMf), Kernel K-Means (KKM), K-Means (KM), K-Medoids (KMD), Mini Batch K-Means (MK) and Ward Agglomerative

Clustering (WA). Table 7 shows the mean, standard deviation and rank position of each clustering algorithm (standard ranking) for all datasets. Table 8 shows how many times each algorithm was in each rank position.

Table 7: Mean, standard deviation and rank of each clustering algorithm.

Algorithm	**Mean**	**STD**	**Rank**
AA	2.8173	2.2459	1
CA	5.6940	2.6625	6
FCM	5.9566	2.1425	7
GMd	6.8789	2.3535	8
GMf	9.2602	1.4737	10
KKM	4.5639	2.1481	3
KM	2.9748	1.4929	2
KMD	6.8881	2.2675	9
MK	4.5821	1.9249	4
WA	5.3835	2.2060	5

5.5. Baseline Methods

Two baseline methods were compared with the proposed approach: (1) standard ranking and (2) majority class ranking. The former is calculated as described in Table 7, returning the average ranking for all training datasets. The later returns the most frequent ranking in the training datasets, according to Table 8. Table 9 shows the mean and standard deviation regarding SRC, ARI and AMI for standard ranking and majority class ranking.

Table 8: Number of times each algorithm assumed each rank position.

Rank	**Algorithm**									
	AA	CA	FCM	GMd	GMf	KKM	KM	KMD	MK	WA
1	99	18	8	2	0	10	44	6	17	15
2	34	31	13	5	4	29	51	11	24	17
3	20	21	14	20	0	43	48	10	21	22
4	19	8	25	20	5	38	38	14	36	16
5	9	16	35	18	1	30	21	14	47	28
6	12	24	25	20	3	24	12	24	36	39
7	15	22	32	25	4	20	3	40	19	39
8	3	46	37	32	18	8	2	37	11	25
9	6	26	25	58	34	7	0	42	6	15
10	2	7	5	19	150	10	0	21	2	3

Table 9: SRC, ARI and AMI mean and standard deviations for the baseline methods.

Baseline Method	**SRC**		**ARI**		**AMI**	
	Mean	**STD**	**Mean**	**STD**	**Mean**	**STD**
Standard Ranking	0.6387	0.0002	0.1956	0.0002	0.1989	0.0001
Majority Class Ranking	0.5748	0.0018	0.1956	0.0002	0.1989	0.0001

5.6. Meta-learning Algorithms

In this study, two learning algorithms were used to induce meta-models able to predict the ranking for a new dataset: k-Nearest Neighbor (k-NN), often used in recommender systems based on meta-learning, and Random Forest (RF), used in more recent studies. Thus, when the meta-learning system is applied to a new dataset, a ranking-based classifier (meta-model) returns a ranking of the most suitable clustering algorithms, based on the meta-feature values extracted from the dataset. Sections 5.7 and 5.8 show the results obtained using the k-NN and RF algorithms, respectively.

5.7. K-Nearest Neighbor Results

This recommendation approach is very simple and widely used in the literature [4, 8]. When a new dataset is presented to the meta-learning system, k-NN finds the most similar datasets in the meta-dataset and aggregates their associated rankings using average ranking. Tables 10, 11 and 12 show, respectively, the SRC, ARI and AMI values (the best results with statistical significance are highlighted in bold) for k-NN recommendations using Euclidean distance. To improve predictive performance, the number of neighbors (k) was varied from 1 to 10.

The change of SRC, ARI and AMI mean values for k varying from 1 to 10 can be seen in Fig. 2, 3 and 4, respectively. The horizontal dotted lines represent the standard ranking and majority class ranking, the line with triangles represents the mean of values obtained by the statistical-based method, the line with circles represents the mean of values when the distance-based method is used, the line with squares represents the mean of values

Table 10: SRC values (mean ± standard deviation) for the ranking recommendation by k-NN.

k	Statistical	Distance	Evaluation	CaD
1	0.5659 ± 0.0045	**0.6201 ± 0.0032**	0.5946 ± 0.0027	0.6114 ± 0.0042
2	0.5840 ± 0.0024	0.6237 ± 0.0036	0.6121 ± 0.0044	**0.6332 ± 0.0032**
3	0.6004 ± 0.0009	0.6444 ± 0.0007	0.6221 ± 0.0013	**0.6501 ± 0.0003**
4	0.6081 ± 0.0022	0.6509 ± 0.0002	0.6287 ± 0.0003	**0.6589 ± 0.0015**
5	0.6172 ± 0.0050	0.6507 ± 0.0033	0.6346 ± 0.0030	**0.6637 ± 0.0022**
6	0.6242 ± 0.0016	0.6544 ± 0.0011	0.6349 ± 0.0040	**0.6728 ± 0.0034**
7	0.6263 ± 0.0019	0.6616 ± 0.0020	0.6427 ± 0.0042	**0.6829 ± 0.0028**
8	0.6330 ± 0.0029	0.6630 ± 0.0013	0.6445 ± 0.0041	**0.6828 ± 0.0025**
9	0.6323 ± 0.0030	0.6675 ± 0.0008	0.6462 ± 0.0015	**0.6812 ± 0.0017**
10	0.6326 ± 0.0038	0.6621 ± 0.0009	0.6484 ± 0.0048	**0.6795 ± 0.0037**

Table 11: ARI values (mean ± standard deviation) for the ranking recommendation by k-NN.

k	Statistical	Distance	Evaluation	CaD
1	**0.4848 ± 0.0032**	0.4732 ± 0.0052	0.4452 ± 0.0003	**0.4850 ± 0.0054**
2	0.4779 ± 0.0032	0.4898 ± 0.0021	0.4291 ± 0.0044	**0.4947 ± 0.0023**
3	0.4718 ± 0.0180	**0.4852 ± 0.0074**	0.4826 ± 0.0104	0.4822 ± 0.0028
4	0.5057 ± 0.0028	**0.5197 ± 0.0078**	0.4599 ± 0.0015	0.5105 ± 0.0083
5	0.5033 ± 0.0026	**0.5371 ± 0.0104**	0.4647 ± 0.0010	**0.5383 ± 0.0150**
6	0.5067 ± 0.0227	0.5182 ± 0.0002	0.5007 ± 0.0044	**0.5561 ± 0.0071**
7	0.5253 ± 0.0024	0.5216 ± 0.0092	0.5110 ± 0.0090	**0.5390 ± 0.0029**
8	0.5234 ± 0.0034	0.5250 ± 0.0043	0.4900 ± 0.0072	**0.5548 ± 0.0028**
9	0.5365 ± 0.0046	0.4954 ± 0.0059	0.4935 ± 0.0003	**0.5618 ± 0.0004**
10	0.5509 ± 0.0192	0.5261 ± 0.0005	0.5257 ± 0.0069	**0.5681 ± 0.0090**

Table 12: AMI values (mean ± standard deviation) for the ranking recommendation by k-NN.

k	Statistical	Distance	Evaluation	CaD
1	**0.4866 ± 0.0027**	0.4777 ± 0.0046	0.4514 ± 0.0006	**0.4878 ± 0.0057**
2	0.4811 ± 0.0025	0.4926 ± 0.0024	0.4350 ± 0.0039	**0.4968 ± 0.0021**
3	0.4755 ± 0.0185	**0.4889 ± 0.0067**	0.4837 ± 0.0102	0.4844 ± 0.0033
4	0.5099 ± 0.0034	**0.5239 ± 0.0078**	0.4618 ± 0.0023	0.5121 ± 0.0082
5	0.5069 ± 0.0022	**0.5421 ± 0.0101**	0.4675 ± 0.0002	**0.5399 ± 0.0154**
6	0.5121 ± 0.0221	0.5231 ± 0.0007	0.5028 ± 0.0054	**0.5572 ± 0.0068**
7	0.5285 ± 0.0015	0.5271 ± 0.0096	0.5140 ± 0.0086	**0.5409 ± 0.0035**
8	0.5250 ± 0.0042	0.5298 ± 0.0044	0.4934 ± 0.0061	**0.5564 ± 0.0018**
9	0.5390 ± 0.0051	0.5003 ± 0.0058	0.4969 ± 0.0006	**0.5636 ± 0.0000**
10	0.5531 ± 0.0191	0.5304 ± 0.0005	0.5276 ± 0.0072	**0.5701 ± 0.0096**

obtained by the evaluation-based method and the line with stars represents the mean values obtained by the CaD method.

To determine the normality of the data, the Shapiro-Wilk test with a significance level of 5% was applied to the SRC, ARI and AMI results obtained over 30 executions. The null hypothesis (H_0) claims that the sample comes from a normally distributed population. All the results showed that the values presented came from a normal distribution. Therefore, the difference between the mean values of the SRC, ARI and AMI results was verified using the Student's t-test, where the null hypothesis (H_0) claims that samples come from normal distributions with equal means. Tables 13, 14 and 15 show p-values from Student's t-test for the SRC, ARI and AMI values, respectively.

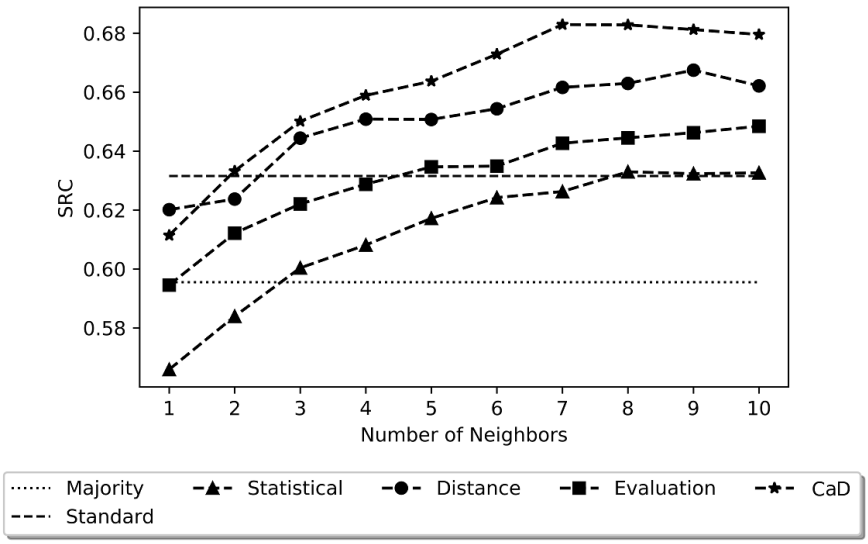

Figure 2: Mean of SRC values according to variation of number of neighbors.

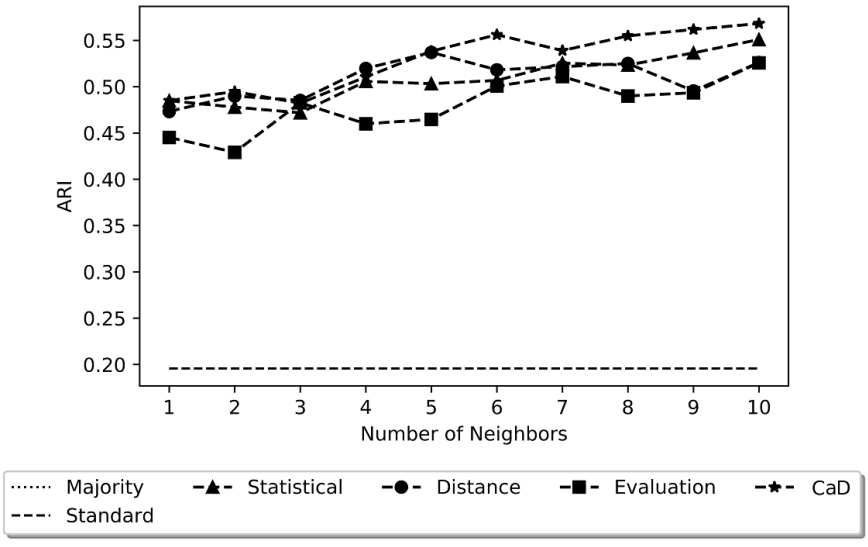

Figure 3: Mean of ARI values according to variation of number of neighbors.

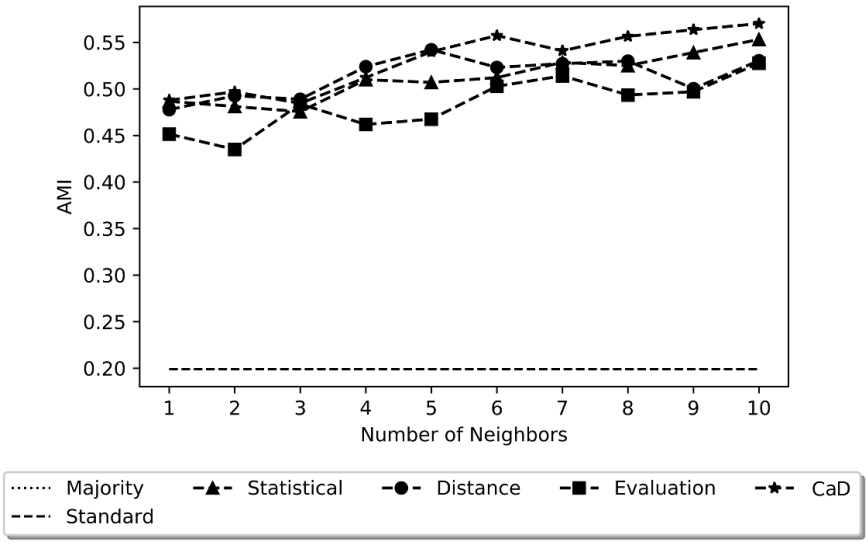

Figure 4: Mean of AMI values according to variation of number of neighbors.

Table 13: P-values from the Student's t-test for SRC: comparison with CaD method.

k	Majority	Standard	Statistical	Distance	Evaluation
1	4.5447e-28	1.7993e-25	4.4968e-27	6.4180e-10	1.4881e-17
2	1.2156e-36	2.6625e-10	2.0297e-33	1.1124e-11	3.1873e-19
3	1.2544e-48	2.8155e-45	1.2403e-51	3.1519e-27	4.0147e-40
4	7.1392e-47	1.9185e-34	6.3338e-39	5.9468e-23	2.3542e-39
5	3.8619e-45	2.2362e-32	7.9651e-29	2.9966e-17	8.9445e-28
6	1.4712e-42	7.5820e-31	4.7703e-34	1.2454e-22	8.8110e-27
7	1.2966e-45	1.6299e-36	2.8411e-37	6.9545e-25	5.3538e-28
8	1.4118e-46	6.7019e-38	4.0558e-34	1.9030e-26	5.1320e-28
9	3.8653e-49	3.0921e-42	3.3455e-35	6.6304e-27	2.8829e-36
10	1.5187e-42	4.9225e-32	2.6765e-29	3.4802e-21	1.3694e-22

Table 14: P-values from the Student's t-test for ARI: comparison with CaD method.

k	Majority	Standard	Statistical	Distance	Evaluation
1	6.5471e-52	6.5471e-52	0.8626	1.7055e-09	5.1039e-27
2	4.9207e-63	4.9207e-63	2.3751e-20	1.7225e-09	2.5762e-34
3	4.9154e-60	4.9154e-60	0.0039	0.0467	0.8402
4	1.4432e-47	1.4432e-47	0.0054	0.0001	1.6876e-24
5	3.4473e-41	3.4473e-41	2.7852e-13	0.7213	5.1011e-22
6	3.1078e-51	3.1078e-51	3.2770e-12	4.5812e-23	9.8277e-26
7	7.1499e-62	7.1499e-62	1.7954e-18	8.6754e-11	4.4019e-16
8	7.0473e-63	7.0473e-63	1.2613e-26	4.2224e-24	1.2139e-28
9	2.9553e-86	2.9553e-86	2.1742e-23	2.8033e-32	1.0595e-63
10	1.1575e-48	1.1575e-48	0.0001	2.02411e-21	8.6467e-19

Table 15: P-values from the Student's t-test for AMI: comparison with CaD method.

k	Majority	Standard	Statistical	Distance	Evaluation
1	3.24972e-51	3.2497e-51	0.3059	2.5168e-08	3.3650e-25
2	3.6626e-64	3.6626e-64	8.4198e-22	6.0907e-08	5.3554e-35
3	6.0649e-58	6.0649e-58	0.0147	0.0025	0.7232
4	1.1808e-47	1.1808e-47	0.1851	3.5305e-06	2.6239e-24
5	8.5113e-41	8.5113e-41	1.9761e-12	0.5180	1.5824e-21
6	1.0528e-51	1.0528e-51	1.4430e-11	3.0234e-22	4.9516e-25
7	1.7752e-59	1.7752e-59	3.5679e-17	3.7672e-08	7.7968e-16
8	2.1412e-68	2.1412e-68	3.5904e-26	1.2055e-23	1.0308e-30
9	4.5433e-105	4.5433e-105	7.7239e-22	6.3526e-32	4.1664e-61
10	8.2696e-48	8.2696e-48	0.0001	5.6953e-20	3.7499e-18

From the SRC values in Table 10, it is important to highlight that, for $k > 1$, the CaD method obtained better results than the two baseline methods and the statistical, distance and evaluation-based ranking methods. On the other hand, statistical-based ranking obtained the worst results, which were similar to those from standard ranking with $k = 10$. These results are confirmed by Table 13. The best k-NN results regarding SRC values were obtained by the CaD method with $k = 7$. The k-NN algorithm was not robust for small values of hyperparameter k regarding the SRC measure. In fact, k-NN does not perform well when k is small [25].

Regarding the ARI index results from Table 11, the results obtained by the CaD method were worse than those obtained by statistical and distance-based ranking for $1 \leq k \leq 5$. However, for $k > 5$, the CaD method recommended better clustering algorithms than all other methods. These results are confirmed in Table 14. The best k-NN results regarding ARI values were obtained by the CaD method with $k = 10$. A similar conclusion applies to the AMI measure shown in Table 12 and confirmed by Table 15. Concerning the reliability of k-NN, it was stable according to the variation of hyperparameter k regarding ARI and AMI measures.

5.8. Random Forest Results

The RF algorithm has been used in several meta-learning studies [40]. RF is an ensemble learning method for classification and regression tasks whose training induces a set of decision trees. When a new instance is presented in a test phase, the RF output is the most frequent classes (on classification task) or mean prediction (on regression task) of the individual trees. RF was used for multi-target regression [33], where the regression values produced

are related with the rank position of the clustering algorithms. Here, nodes are expanded until all leaves are pure or until all leaves contain less than 2 instances. All meta-features are used to expand the nodes. The number of trees (T) was varied from 1 to 10, to obtain a good ranking, with a low forest size. The outputting class is the mean prediction of the individual trees. Tables 16, 17 and 18 show, respectively, the SRC, ARI and AMI values (the best results with statistical significance are highlighted in bold) for RF recommendations.

Table 16: SRC values (mean ± standard deviation) for the ranking recommendations by RF.

k	Statistical	Distance	Evaluation	CaD
1	**0.6032 ± 0.0068**	0.5296 ± 0.0325	0.5475 ± 0.0309	0.5549 ± 0.0367
2	**0.6503 ± 0.0037**	0.6222 ± 0.0001	0.6394 ± 0.0033	0.6424 ± 0.0047
3	**0.6815 ± 0.0054**	0.6585 ± 0.0031	0.6574 ± 0.0087	0.6651 ± 0.0026
4	**0.6795 ± 0.0055**	0.6648 ± 0.0050	0.6723 ± 0.0043	0.6732 ± 0.0025
5	**0.6857 ± 0.0039**	0.6776 ± 0.0019	0.6812 ± 0.0044	0.6830 ± 0.0050
6	**0.6937 ± 0.0021**	0.6862 ± 0.0020	0.6845 ± 0.0052	0.6856 ± 0.0038
7	**0.6927 ± 0.0010**	0.6853 ± 0.0024	0.6819 ± 0.0055	0.6825 ± 0.0026
8	**0.6915 ± 0.0002**	0.6847 ± 0.0011	0.6772 ± 0.0052	0.6815 ± 0.0019
9	**0.6948 ± 0.0007**	0.6885 ± 0.0001	0.6821 ± 0.0030	0.6871 ± 0.0029
10	**0.6912 ± 0.0006**	0.6853 ± 0.0008	0.6760 ± 0.0029	0.6825 ± 0.0027

Based on Tables 16, 17 and 18, Figures 5, 6 and 7 show, respectively, the SRC, ARI and AMI mean values for recommendations by RF with T varying from 1 to 10. The horizontal dotted lines represent the standard ranking and majority class ranking, the line with triangles represents the

Table 17: ARI values (mean ± standard deviation) for the ranking recommendations by RF.

k	Statistical	Distance	Evaluation	CaD
1	0.3661 ± 0.0716	**0.6412 ± 0.0344**	0.5173 ± 0.0467	0.5252 ± 0.0790
2	0.4789 ± 0.0422	0.5286 ± 0.0715	0.5122 ± 0.0375	**0.6300 ± 0.0411**
3	0.3211 ± 0.0412	0.4783 ± 0.0447	**0.5945 ± 0.0448**	0.5459 ± 0.0394
4	0.3211 ± 0.0412	0.5658 ± 0.0448	0.5861 ± 0.0364	**0.6331 ± 0.0362**
5	0.2861 ± 0.0028	**0.5201 ± 0.0019**	**0.5113 ± 0.0378**	**0.5534 ± 0.1150**
6	0.3211 ± 0.0412	0.5201 ± 0.0019	0.5492 ± 0.0792	**0.5911 ± 0.0781**
7	0.3601 ± 0.0803	0.5209 ± 0.0028	0.5515 ± 0.0815	**0.6307 ± 0.1177**
8	0.4391 ± 0.0852	0.5209 ± 0.0028	0.5492 ± 0.0792	**0.6307 ± 0.1177**
9	0.3229 ± 0.0362	0.4455 ± 0.0726	0.5492 ± 0.0792	**0.5885 ± 0.0755**
10	0.4428 ± 0.0097	0.4455 ± 0.0726	0.5492 ± 0.0792	**0.5885 ± 0.0755**

Table 18: AMI values (mean ± standard deviation) for the ranking recommendations by RF.

k	Statistical	Distance	Evaluation	CaD
1	0.3688 ± 0.0716	**0.6430 ± 0.0354**	0.5187 ± 0.0471	0.5267 ± 0.0804
2	0.4806 ± 0.0411	0.5303 ± 0.0727	0.5136 ± 0.0386	**0.6320 ± 0.0395**
3	0.3236 ± 0.0413	0.4811 ± 0.0427	**0.5954 ± 0.0432**	0.5483 ± 0.0406
4	0.3236 ± 0.0413	0.5671 ± 0.0436	0.5879 ± 0.0357	**0.6347 ± 0.0378**
5	0.2882 ± 0.0025	0.5220 ± 0.0010	0.5121 ± 0.0373	**0.5563 ± 0.1164**
6	0.3236 ± 0.0413	0.5220 ± 0.0010	0.5500 ± 0.0786	**0.5931 ± 0.0794**
7	0.3626 ± 0.0802	0.5231 ± 0.0021	0.5520 ± 0.0806	**0.6327 ± 0.1190**
8	0.4412 ± 0.0847	0.5231 ± 0.0021	0.5500 ± 0.0786	**0.6327 ± 0.1190**
9	0.3260 ± 0.0369	0.4475 ± 0.0734	0.5500 ± 0.0786	**0.5909 ± 0.0773**
10	0.4455 ± 0.0085	0.4475 ± 0.0734	0.5500 ± 0.0786	**0.5909 ± 0.0773**

mean SRC values obtained by the statistical-based method, the line with circles represents the mean SRC values when the distance is used, the line with squares represents the mean of values obtained by the evaluation-based method and the line with stars represents the mean SRC values obtained by the CaD method.

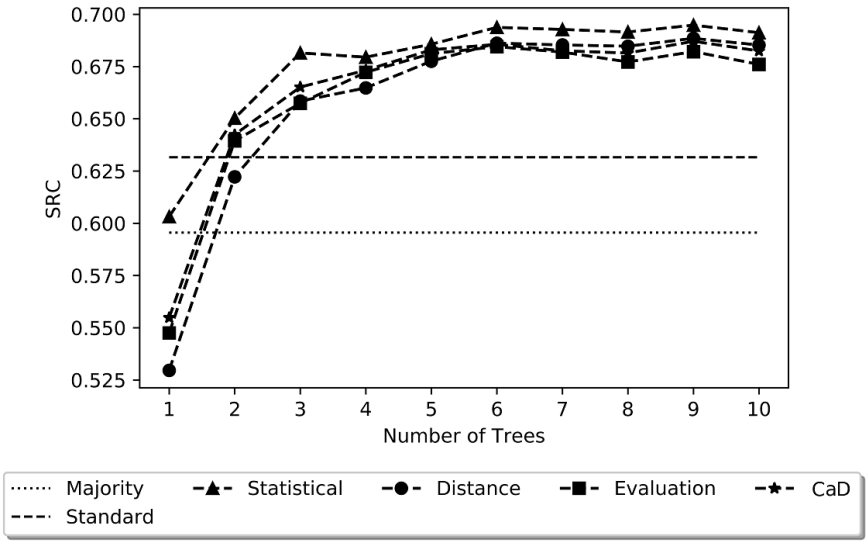

Figure 5: Mean of SRC values according to variation of the number of trees.

The Shapiro-Wilk test was used with a significance level of 5% to determine the normality of the data. This statistical test was applied to the SRC, ARI and AMI results over 30 executions. In this test, the null hypothesis (H_0) states that the sample comes from a normally distributed population. All the results for RF showed that the values presented came from a normal distribution. Therefore, the Student's t-test was used to analyze the difference between the mean values of the SRC, ARI and AMI results, where the null hypothesis (H_0) claims that samples come from normal distributions

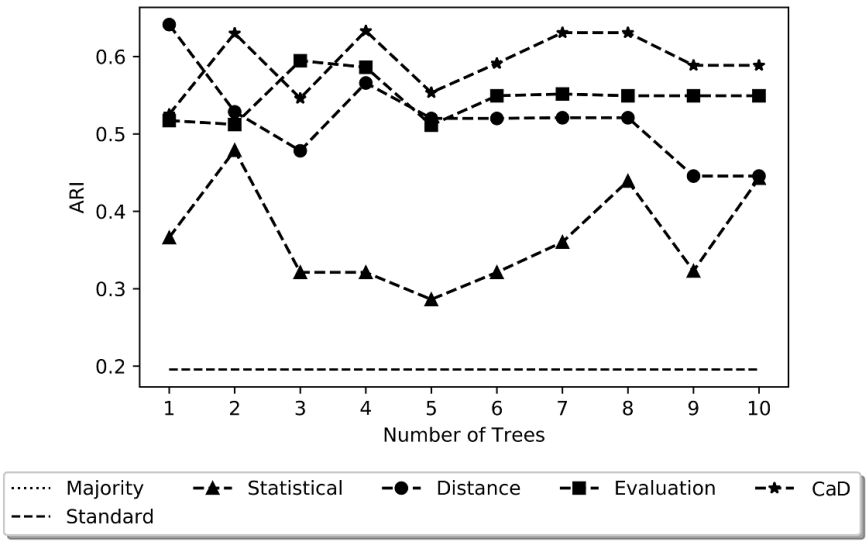

Figure 6: Mean of ARI values according to variation of the number of trees.

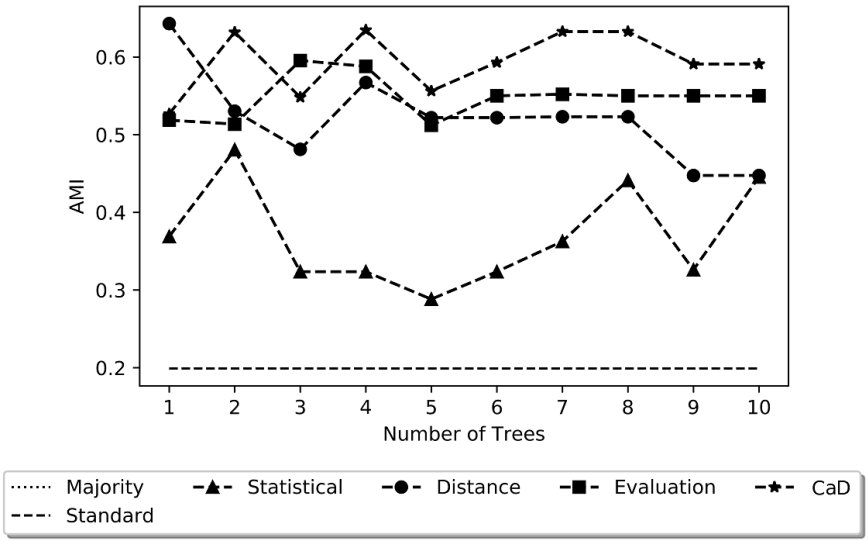

Figure 7: Mean of AMI values according to variation of the number of trees.

with equal means. Tables 19, 20 and 21 show p-values from the Student's t-test for SRC, ARI and AMI values, respectively.

Table 19: P-values from the Student's t-test for SRC: comparison with the CaD method.

k	**Majority**	**Standard**	**Statistical**	**Distance**	**Evaluation**
1	0.0059	3.2977e-13	8.4820e-08	0.0084	0.4051
2	1.6039e-34	0.0001	5.7751e-08	1.9098e-20	0.0077
3	5.3869e-44	5.5102e-31	3.4306e-15	7.9672e-10	6.8032e-05
4	2.095e-45	8.0563e-35	3.5237e-06	4.4895e-09	0.3298
5	9.6441e-40	2.5899e-29	0.0268	5.8284e-06	0.1495
6	5.5191e-43	1.9141e-33	4.0319e-11	0.450	0.3572
7	3.2672e-46	2.5193e-37	1.5233e-18	0.0001	0.5931
8	1.7811e-48	6.0696e-41	7.8602e-23	8.3567e-09	0.0001
9	8.8875e-46	3.2449e-37	1.5331e-14	0.0131	3.4383e-07
10	6.9043e-46	7.4525e-37	8.9620e-17	7.3474e-06	7.0574e-10

According to the experimental results from Tables 16 and 19, regarding the SRC index, the performance of the CaD method was better than the performance of the distance-based method for all values of T analysed, except for $T = 5$ when there is no statistically significant difference. Moreover, the CaD method showed better results than the evaluation-based method for $k = \{2, 3, 8, 9, 10\}$. Besides, the CaD method was better than both the standard and majority class ranking for $T > 1$. Moreover, it is important to highlight that the higher the values of the number of trees ($T \geq 7$), the more the performance of the proposed, statistical, distance and evaluation-based recommendation methods tend to stabilize. Similarly to k-NN, the RF algorithm is non-robust for small values of hyperparameter T. However,

Table 20: P-values from the Student's t-test for ARI: comparison with the CaD method.

k	Majority	Standard	Statistical	Distance	Evaluation
1	4.3025e-20	4.3025e-20	5.1782e-09	4.0000e-08	0.6408
2	1.5986e-31	1.5986e-31	1.7955e-14	2.1718e-07	2.0683e-12
3	2.2920e-29	2.2920e-29	2.0220e-19	8.8718e-07	0.0001
4	3.3704e-33	3.3704e-33	7.5585e-24	5.3505e-07	2.4358e-05
5	1.1969e-16	1.1969e-16	2.1412e-13	0.12363	0.0667
6	1.9846e-22	1.9846e-22	1.8936e-16	2.6979e-05	0.0481
7	1.1760e-18	1.1760e-18	2.6775e-11	1.8779e-05	0.0051
8	1.1760e-18	1.1760e-18	5.9491e-08	1.8779e-05	0.0038
9	9.2489e-23	9.2489e-23	7.1674e-17	3.0552e-08	0.0587
10	9.2489e-23	9.2489e-23	2.2354e-11	3.0552e-08	0.0587

Table 21: P-values from the Student's t-test for AMI: comparison with the CaD method.

k	Majority	Standard	Statistical	Distance	Evaluation
1	8.1058e-20	8.1058e-20	7.3678e-09	5.5184e-08	0.6416
2	5.5590e-32	5.5590e-32	7.3855e-15	2.1826e-07	1.5329e-12
3	5.8265e-29	5.8265e-29	3.1511e-19	8.1100e-07	0.0001
4	1.3101e-32	1.3101e-32	1.4656e-23	5.1147e-07	3.0804e-05
5	1.6972e-16	1.6972e-16	2.6775e-13	0.1173	0.0571
6	3.4521e-22	3.4521e-22	2.8354e-16	3.3097e-05	0.0433
7	1.7180e-18	1.7180e-18	3.2926e-11	2.2459e-05	0.0045
8	1.7180e-18	1.7180e-18	6.6356e-08	2.2459e-05	0.0035
9	1.9078e-22	1.9078e-22	1.4036e-16	4.0624e-08	0.0513
10	1.9078e-22	1.9078e-22	3.8357e-11	4.0624e-08	0.0513

RF was more robust than k-NN when the hyperparameter was increased, as expected, since RF is relatively robust to outliers and noise [29].

In Tables 17 and 20, related with the ARI index, it can be observerd that the best method recommended by RF using the CaD method obtained the better clustering quality than the using the statistical-based and distance-based rankings and the baselines, except for $T = 5$, when the CaD, distance and evaluation methods obtained similar results. The same is true regarding the AMI measure shown in Table 18 and confirmed by Table 21. Concerning the ARI and AMI measures, RF showed to be less reliable than k-NN, since ARI and AMI values have a larger variation when RF algorithm is applied.

6. Importance of Meta-features

Different meta-features can have different effects in the induction of the meta-models. This information can be used to select a subset of the most relevant meta-features, reducing feature extraction and model induction costs and inducing less complex meta-models, which can increase its generalization. Despite these motivations, few works have proposed strategies to analyze the importance of each meta-feature for the recommendation process. To understand the importance of each meta-feature for the four sets of meta-features, the proposed strategy looks at the meta-features selected by the RF algorithm. This analysis was based on the relative frequency of the meta-features in the trees created by the RF algorithm. The meta-features selected to split the tree nodes indicate the importance of the meta-feature for the decision tree model created. As the RF algorithm creates an ensemble of decision trees, the relative frequency of a given meta-feature is computed as

the average of occurrences of this meta-feature for all trees. The following equations show how the relative frequency is computed.

Let s_{tj} be an indicator function, which is equal to 1 if the meta-feature j is selected to split a node in tree t of the RF algorithm.

$$s_{tj} = \begin{cases} 1, & \text{if meta-feature } j \text{ is selected to split a node in tree } t, \\ 0, & \text{otherwise.} \end{cases}$$

Based on s_{tj}, the relative frequency f_j of meta-feature j is computed according to the average of frequencies regarding all trees (T) of the RF algorithm.

$$f_j = \frac{\sum_{t=1}^{T} s_{tj}}{T} \tag{12}$$

The higher the f_j value, the more important the meta-feature j for the clustering algorithm recommendation. For the statistical-based method, if all meta-features have the same importance, the importance value for these meta-features will be 12.5% (100%/8). For the distance-based meta-features, evaluation-based meta-features and the CaD method, the importance value for these meta-features will be 5.2631% (100%/19). Thus, importance values higher than 12.5% or 5.2631% indicate an important meta-feature.

The overall values were calculated with the same 10-fold cross-validation partitions used in the previous meta-learning experiments. For each fold partition, a dataset is selected and the importance value of a given meta-feature is computed. After selecting all the datasets in a fold partition, the mean of the importance values is calculated. When all the fold partitions

are used, the mean of 10 importance values is computed. This process is repeated 30 times for all meta-features. Afterwards, the mean and standard deviation of the 30 importance values of each meta-feature are computed. Table 22 shows the importance values of meta-features for the RF-based ranking recommendation when $T = 10$ (mean $\pm$ standard deviation and rank between parenthesis).

It can be observed that, on average, the meta-features 1 (LgE), 2 (LgREA), 5 (SK) and 8 (PO) were the most important for the statistical-based method. In particular, LgREA (the logarithm of the ratio between the number of instances and the number of attributes in a dataset) was the most relevant to recommend a clustering algorithm. On the other hand, the meta-feature 4 (MN - Multivariate Normality) had a low effect. Regarding the distance-based method, the meta-feature MF_6 (% of values in the interval $(0.0, 0.1]$) was the most important. For evaluation-based meta-features, measure 18 (XB index) was the most relevant for the recommendation, as observed in [46]. Moreover, meta-features 15 (silhouette index) and 2 (LgREA) were also important to recommend a clustering algorithm. For the CaD method, the meta-feature MF_{10} (% of values in the interval $(0.4, 0.5]$) was the most important. It is also worth noticing that the meta-features MF_2 (Variance) and MF_3 (Standard deviation) had a lower importance for both distance-based and CaD methods. On the other hand, the meta-features MF_{10} (% of values in the interval $(0.4, 0.5]$) and MF_{11} (% of values in the interval $(0.5, 0.6]$) had a high importance for both methods.

Table 22: Importance values (mean ± standard deviation) for the ranking recommendation by RF.

MF	Statistical	Distance	Evaluation	CaD
1	0.2390 ± 0.0014 (3)	0.0356 ± 0.0021 (16)	0.0641 ± 0.0021 (7)	0.0520 ± 0.0028 (9)
2	0.2450 ± 0.0032 (1)	0.0342 ± 0.0016 (18)	0.0849 ± 0.0034 (3)	0.0337 ± 0.0019 (19)
3	0.0000 ± 0.0000 (8)	0.0315 ± 0.0011 (19)	0.0000 ± 0.0000 (19)	0.0343 ± 0.0033 (18)
4	0.0739 ± 0.0015 (5)	0.0711 ± 0.0030 (4)	0.0325 ± 0.0042 (11)	0.0536 ± 0.0018 (7)
5	0.2408 ± 0.0033 (2)	0.0416 ± 0.0034 (14)	0.0746 ± 0.0052 (5)	0.0724 ± 0.0059 (4)
6	0.0000 ± 0.0000 (8)	0.1149 ± 0.0026 (1)	0.0000 ± 0.0000 (19)	0.0454 ± 0.0018 (12)
7	0.0000 ± 0.0000 (8)	0.0480 ± 0.0012 (9)	0.0000 ± 0.0000 (19)	0.0462 ± 0.0006 (11)
8	0.2010 ± 0.0021 (4)	0.0484 ± 0.0007 (7)	0.0657 ± 0.0073 (6)	0.0439 ± 0.0008 (13)
9	–	0.0396 ± 0.0008 (15)	0.0587 ± 0.0022 (9)	0.0448 ± 0.0021 (14)
10	–	0.0730 ± 0.0030 (3)	0.0062 ± 0.0016 (15)	0.0859 ± 0.0019 (1)
11	–	0.0815 ± 0.0035 (2)	0.0051 ± 0.0073 (16)	0.0739 ± 0.0020 (3)
12	–	0.0483 ± 0.0016 (8)	0.0148 ± 0.0072 (14)	0.0643 ± 0.0006 (5)
13	–	0.0353 ± 0.0033 (17)	0.0291 ± 0.0034 (13)	0.0465 ± 0.0017 (10)
14	–	0.0488 ± 0.0016 (6)	0.0780 ± 0.0082 (4)	0.0429 ± 0.0036 (15)
15	–	0.0471 ± 0.0022 (10)	0.1065 ± 0.0025 (2)	0.0742 ± 0.0016 (2)
16	–	0.0445 ± 0.0028 (13)	0.0487 ± 0.0029 (10)	0.0566 ± 0.0032 (6)
17	–	0.0450 ± 0.0034 (11)	0.0624 ± 0.0062 (8)	0.0529 ± 0.0050 (8)
18	–	0.0663 ± 0.0047 (5)	0.1498 ± 0.0034 (1)	0.0368 ± 0.0022 (17)
19	–	0.0445 ± 0.0022 (12)	0.0309 ± 0.0022 (12)	0.0388 ± 0.0007 (16)

7. Conclusion

This paper investigated the automatic recommendation of clustering algorithms. To improve the recommendation, it proposed and experimentally assessed a new set of meta-features for data characterization. Data characterization is one important part of recommender systems based on meta-learning. The proposed set of meta-features combines correlation and dissimilarity measures. Experiments were carried out to evaluate the predictive performance of a clustering algorithm recommender system using this new set of meta-features. The system recommend a ranking of the most suitable clustering algorithms for a new dataset.

The evaluation occurred in two levels. The first, namely the meta-level, compared the predictive performance of the recommender with the new set of meta-features against the stare-of-the-art of recommender systems based on meta-learning for clustering algorithm recommendations. The second, namely the base level, evaluated whether the clustering algorithms recommended using the proposed approach performed better the than clustering algorithms recommended by the existing approaches, used as baselines.

At the meta-level, with statistical significance, the rankings recommended by the proposed approach were better than those recommended by meta-features from the related literature. They were also closer to the true rankings (asserted by Spearman's coefficient), when using K-Nearest Neighbor as predictor method.

At the base level, for meta-models produced by both K-Nearest Neighbor and Random Forest, the clustering algorithm recommended as the most suitable by the proposed method obtained better clustering quality (asserted

by Adjusted Rand's Index). A similar conclusion was obtained for Adjusted Mutual Information.

In order to analyze the importance of the meta-features for the three data characterization approaches, the meta-features selected by Random Forest for each set of meta-features were analyzed. According to the analysis performed, some meta-features showed to be more suitable than other ones for the recommendation of clustering algorithms.

As future work, new meta-features, extracting other aspects from the datasets, could be proposed and experimentally investigated. The use of more datasets could increase the reliability of the meta-learners. Besides, other classification algorithms could also be used in the recommender system. Finally, an attribute selection technique could be applied for the meta-features extracted by the three methods.

Acknowledgement

The authors would like to thank FAPESP (processes 2013/07375-0, 2016/18615-0 and 2017/20265-0), Intel, CAPES and CNPq for their support.

References

[1] Adam, A., Blockeel, H., 2015. Dealing with overlapping clustering: a constraint-based approach to algorithm selection. In: Meta-learning and Algorithm Selection Workshop. Vol. 1. CEUR Workshop proceedings, pp. 43–54.

[2] Baker, F. B., Hubert, L. J., 1975. Measuring the power of hierarchi-

cal cluster analysis. Journal of the American Statistical Association 70 (349), 31–38.

[3] Boddy, R., Smith, G., 2009. Statistical methods in practice: for scientists and technologists. John Wiley & Sons.

[4] Brazdil, P., Carrier, C. G., Soares, C., Vilalta, R., 2008. Metalearning: Applications to data mining. Springer Science & Business Media.

[5] Caliński, T., Harabasz, J., 1974. A dendrite method for cluster analysis. Communications in Statistics-theory and Methods 3 (1), 1–27.

[6] Cunha, T., Soares, C., Carvalho, A. C. P. L. F., 2018. Metalearning and recommender systems: A literature review and empirical study on the algorithm selection problem for collaborative filtering. Information Sciences 423, 128–144.

[7] Davies, D. L., Bouldin, D. W., 1979. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (2), 224–227.

[8] De Souto, M. C. P., Prudencio, R. B., Soares, R. G., De Araujo, D. S., Costa, I. G., Ludermir, T. B., Schliep, A., 2008. Ranking and selecting clustering algorithms using a meta-learning approach. In: IEEE International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 3729–3735.

[9] de Winter, J. C., Gosling, S. D., Potter, J., 2016. Comparing the pearson and spearman correlation coefficients across distributions and sample

sizes: A tutorial using simulations and empirical data. Psychological Methods 21 (3), 273.

[10] Dunn, J. C., 1974. Well-separated clusters and optimal fuzzy partitions. Journal of Cybernetics 4 (1), 95–104.

[11] Ekstrand, M., Riedl, J., 2012. When recommenders fail: predicting recommender failure for algorithm selection and combination. In: Proceedings of the sixth ACM conference on Recommender systems. ACM, pp. 233–236.

[12] Ezugwu, A. E.-S., Adewumi, A. O., Frîncu, M. E., 2017. Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. Expert Systems with Applications 77, 189–210.

[13] Ferrari, D. G., De Castro, L. N., 2015. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. Information Sciences 301, 181–194.

[14] Fix, E., Hodges Jr, J. L., 1951. Discriminatory analysis-nonparametric discrimination: consistency properties. Tech. rep., California Univ Berkeley.

[15] Gan, G., Ma, C., Wu, J., 2007. Data clustering: theory, algorithms, and applications. SIAM.

[16] Garcia, L. P. F., Carvalho, A. C. P. L. F., Lorena, A. C., 2016. Noise detection in the meta-learning level. Neurocomputing 176, 14–25.

[17] Garcia, L. P. F., Lorena, A. C., Matwin, S., Carvalho, A. C. P. L. F., 2016. Ensembles of label noise filters: a ranking approach. Data Mining and Knowledge Discovery 30 (5), 1192–1216.

[18] Giraud-Carrier, C., 2005. The data mining advisor: meta-learning at the service of practitioners. In: Proceedings Fourth International Conference on Machine Learning and Applications. IEEE, pp. 7–pp.

[19] Goodman, L., Kruskal, W., 1954. Measures of association for cross classifications. Journal of the American Statistical Association 49 (268), 732–764.

[20] Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2001. On clustering validation techniques. Journal of Intelligent Information Systems 17 (2), 107–145.

[21] Han, J., Pei, J., Kamber, M., 2011. Data mining: concepts and techniques. Elsevier.

[22] Horváth, T., Mantovani, R. G., Carvalho, A. C. P. L. F., 2016. Effects of random sampling on svm hyper-parameter tuning. In: International Conference on Intelligent Systems Design and Applications. Springer, pp. 268–278.

[23] Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification 2 (1), 193–218.

[24] Jain, A. K., Murty, M. N., Flynn, P. J., 1999. Data clustering: a review. ACM Computing Surveys 31 (3), 264–323.

[25] Jiang, L., Cai, Z., Wang, D., Jiang, S., 2007. Survey of improving k-nearest-neighbor for classification. In: Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on. Vol. 1. IEEE, pp. 679–683.

[26] Johnson, R. A., Wichern, D. W., et al., 2014. Applied multivariate statistical analysis. Vol. 4. Prentice-Hall New Jersey.

[27] Kalousis, A., 2002. Algorithm selection via meta-learning. University of Geneva, Genebra.

[28] Kanda, J., Carvalho, A. C. P. L. F., Hruschka, E. R., Soares, C., Brazdil, P., 2016. Meta-learning to select the best meta-heuristic for the traveling salesman problem: A comparison of meta-features. Neurocomputing 205, 393–406.

[29] Khoshgoftaar, T. M., Golawala, M., Van Hulse, J., 2007. An empirical study of learning from imbalanced data using random forest. In: Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE international conference on. Vol. 2. IEEE, pp. 310–317.

[30] Lemke, C., Gabrys, B., 2010. Meta-learning for time series forecasting and forecast combination. Neurocomputing 73 (10-12), 2006–2016.

[31] Leyva, E., Caises, Y., Gonzlez, A., Prez, R., 2014. On the use of meta-learning for instance selection: An architecture and an experimental study. Information Sciences 266, 16 – 30.

[32] Mantovani, R. G., Rossi, A. L. D., Vanschoren, J., Bischl, B., Carvalho, A. C. P. L. F., 2015. To tune or not to tune: Recommending when to

adjust SVM hyper-parameters via meta-learning. In: International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–8.

[33] Melki, G., Cano, A., Kecman, V., Ventura, S., 2017. Multi-target support vector regression via correlation regressor chains. Information Sciences 415-416, 53 – 69.

[34] Prudêncio, R. B., De Souto, M. C., Ludermir, T. B., 2011. Selecting machine learning algorithms using the ranking meta-learning approach. In: Meta-learning in computational intelligence. Springer, pp. 225–243.

[35] Prudêncio, R. B. C., Ludermir, T. B., 2004. Meta-learning approaches to selecting time series models. Neurocomputing 61, 121–137.

[36] Rand, W. M., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66 (336), 846–850.

[37] Ray, S., Turi, R. H., 1999. Determination of number of clusters in k-means clustering and application in colour image segmentation. In: Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques. Calcutta, India, pp. 137–143.

[38] Rousseeuw, P. J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65.

[39] Stephenson, W. R., 1990. Distribution free tests.

[40] Sun, Q., Pfahringer, B., 2013. Pairwise meta-rules for better meta-learning-based algorithm ranking. Machine Learning 93 (1), 141–161.

[41] Thrun, S., Pratt, L., 2012. Learning to learn. Springer Science & Business Media.

[42] Tripathy, M., Panda, A., 2017. A study of algorithm selection in data mining using meta-learning. Journal of Engineering Science & Technology Review 10 (2).

[43] Vilalta, R., Drissi, Y., 2002. A perspective view and survey of meta-learning. Artificial Intelligence Review 18 (2), 77–95.

[44] Vilalta, R., Giraud-Carrier, C. G., Brazdil, P., Soares, C., 2004. Using meta-learning to support data mining. International Journal on Computational Science & Applications 1 (1), 31–45.

[45] Vinh, N., Epps, J., Bailey, J., 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research 11, 2837–2854.

[46] Vukicevic, M., Radovanovic, S., Delibasic, B., Suknovic, M., 2016. Extending meta-learning framework for clustering gene expression data with component-based algorithm design and internal evaluation measures. International Journal of Data Mining and Bioinformatics 14 (2), 101–119.

[47] Wang, G., Song, Q., Zhang, X., Zhang, K., 2014. A generic multilabel learning-based classification algorithm recommendation method. ACM Transactions on Knowledge Discovery from Data 9 (1), 7.

[48] Xie, X., Beni, G., 1991. A validity measure for fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (8), 841–847.