Information Sciences 301 (2015) 181–194

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods

Daniel Gomes Ferrari *, Leandro Nunes de Castro

Natural Computing Laboratory (LCoN), Mackenzie University, Brazil

ARTICLE INFO

Article history:
Received 20 March 2014
Received in revised form 13 December 2014
Accepted 27 December 2014
Available online 3 January 2015

Keywords:
Clustering
Problem characterization
Algorithm ranking
Algorithm selection
Meta-knowledge
Meta-learning systems

ABSTRACT

Data clustering aims to segment a database into groups of objects based on the similarity among these objects. Due to its unsupervised nature, the search for a good-quality solution can become a complex process. There is currently a wide range of clustering algorithms, and selecting the best one for a given problem can be a slow and costly process. In 1976, Rice formulated the Algorithm Selection Problem (ASP), which postulates that the algorithm performance can be predicted based on the structural characteristics of the problem. Meta-learning brings the concept of learning about learning; that is, the meta-knowledge obtained from the algorithm learning process allows the improvement of the algorithm performance. Meta-learning has a major intersection with data mining in classification problems, in which it is normally used to recommend algorithms. The present paper proposes new ways to obtain meta-knowledge for clustering tasks. Specifically, two contributions are explored here: (1) a new approach to characterize clustering problems based on the similarity among objects; and (2) new methods to combine internal indices for ranking algorithms based on their performance on the problems. Experiments were conducted to evaluate the recommendation quality. The results show that the new meta-knowledge provides high-quality algorithm selection for clustering tasks.

1. Introduction

It is well known that no single algorithm achieves the best performance over all instances of a problem class [75,76,80]. Rice [70] formulated the *Algorithm Selection Problem* (ASP), which proposes that there is a relation between the characteristics of a problem and the performance of the algorithm that can be used to solve it. The ASP is considered an NP-hard problem and has been tackled in different research fields [20,64].

The *meta-learning* field deals with the ASP by learning about the behavior of the learning algorithms [2,52,57,74]. A meta-learning system aims to learn which problems' characteristics contribute to a better performance of one algorithm over others [32,69] and, from this knowledge (*meta-knowledge*), to select the most suitable algorithm for a new, unseen problem [12].

The meta-knowledge, also known as *meta-data*, can be composed of the *meta-attributes* and the *meta-target* [12]. The meta-attributes are the characteristics or features extracted from the problems. The meta-target is the target variable for the meta-learning system [15,52]. Concerning the ASP, the meta-target may be the performance to be estimated or the rank

* Corresponding author.
E-mail addresses: ferrari.dg@gmail.com (D.G. Ferrari), lnunes@mackenzie.br (L.N. de Castro).

http://dx.doi.org/10.1016/j.ins.2014.12.044
0020-0255/

to be recommended for an algorithm [12,78]. An extensive review of meta-learning and the ASP can be found in Smith-Miles [75,76], in which the author presented how their integration is applied to different research fields, such as time series prediction [5,68], sorting [34], and optimization [75,76], among others.

Clustering is an unsupervised data mining task in which the goal is to find groups of similar objects [1,46]; the objects in the same group are more similar to one another than to objects from other groups. There are currently countless clustering algorithms for data mining tools. However, there is a lack of guidelines in the selection of algorithms for the analysis of a new problem [12].

The connections between data mining and meta-learning have been widely investigated for supervised tasks, such as classification [57,71,75,76]; however, there is no study, for instance, about the best meta-attributes to be extracted from unsupervised learning problems [12]. New data mining fields are applying meta-learning techniques to improve the performance of new methods, such as ensembles [49,51], mining data stream [72], mining big data [54], and among others.

Despite the large number of works related to the algorithm selection problem in the meta-learning literature, there are still grand challenges to be overcome [10]. The costly problem characterization and algorithm evaluation processes, together with the high-dimensional data sets, demand new ways to characterize and evaluate problems [10].

The works that tackle the algorithm selection problem by using meta-learning systems in the clustering context have in common the use of external indices to evaluate the algorithms. In other words, the clustering problems have previous known solutions, and these are used to quantify the proposed solutions. This common ground among these works makes it harder to extend the meta-knowledge because real-world clustering tasks usually do not have a priori known solutions.

In de Souto et al. [22], the authors used 32 microarray data sets about cancer gene expression as problems and 7 clustering algorithms. The problems were characterized by 8 meta-attributes, including one specific to the technology used in the construction of the microarray, and the algorithms were evaluated by an external index because the solution was already known. With a regression technique used to select the algorithm, the work presented good results. In Nascimento et al. [61], the authors applied meta-learning techniques to construct rules to automatically select clustering algorithms for gene expression data. A collection of 35 data sets was characterized by 13 meta-attributes, some of which were related to the size of existing clusters. By using an external index, 7 algorithms were evaluated by varying the similarity measure. The experiments aimed to find the best algorithm for rule extraction.

Soares et al. [77] used 160 artificially generated data sets characterized by 9 meta-attributes; the experiments evaluated two algorithms for ranking prediction: a neural network and a regression technique. The ranking was built with 9 clustering algorithms by using an error rate because the object labels were known in advance. In Ferrari and de Castro [26,27], the authors built a meta-learning system with 30 problems and 10 meta-attributes; 5 clustering algorithms were evaluated by using an external measure. Three estimation techniques were tested to predict the performance of the algorithms, and four classifiers were evaluated in a ranking recommendation task.

By taking into account these few related works, the present paper carries out an investigation into the meta-knowledge for clustering tasks. It proposes a new unsupervised characterization scheme based on the distance of objects; that is, an approach to obtain the meta-attributes that does not take into account the labels of objects but, instead, their distance, thus maintaining the unsupervised nature of the clustering task. Besides, the algorithms are evaluated by using different internal indices, which do not need a known solution, and they are combined by means of two new ranking combination methods.

To validate the proposals, a collection of 84 problems, 7 algorithms, and 10 internal indices are used, and the meta-knowledge is built with three distinct sets of meta-attributes: the traditional approach, the new unsupervised distance-based method, and a combination of both characterizations. Three ranking combination techniques are used for performance assessment: an existing method that uses the average rank position and two new methods based on score and competition. Then, this knowledge is applied to a meta-learning system to learn the relation between the problems' features and the performance of the algorithms, and the meta-knowledge quality is assessed by a selection mechanism.

The paper is organized as follows. Section 2 presents a meta-learning system for algorithm selection with the classic approach to obtain the meta-knowledge. Section 3 presents the main contributions of this work to the meta-knowledge of clustering tasks. Section 4 describes the experiments and the results obtained by the meta-learning system for the algorithm selection problem. Section 5 provides a discussion about the results presented and avenues for future research.

2. Meta-learning system

Building a meta-learning system to deal with the ASP requires the meta-knowledge from the learning process. This involves the following steps: collecting problems, choosing algorithms and evaluation measures, extracting meta-attributes, and determining the performance of the algorithms [12,52]. Then, when a new problem is presented to the meta-learning system, the meta-attributes are obtained, and a selection mechanism that makes use of the meta-knowledge database provides the ranking of the algorithms for the unseen problem.

Obtaining the meta-knowledge is a crucial step for the success of a meta-learning system and has been the subject of study in different research fields in the machine-learning community [12,47,52,75,76]. In the present paper, the concept of meta-knowledge is used as meta-data; i.e., the meta-knowledge is stored as an object composed of meta-attributes, which characterize the problems, and the ranking, which indicates the performance of the algorithms.

2.1. Clustering problems

The problem collection consists of 84 data sets from the UCI Machine Learning Repository [6,29], covering different task domains, such as engineering, social sciences, physics, biology, medicine, and games, among others (Appendix A). The dimensionalities of the problems are plotted in Fig. 1. Before being added to the database, all data sets underwent a preprocessing step in the following order:

(1) All information about labels or classes were removed from the data sets.
(2) All nominal values were converted into numbers following the alphabetical order of the unique values.
(3) Attributes that have the same value for all objects were removed.
(4) Attributes that have a different value for each object were removed.
(5) Attributes that have more than 40% missing values were removed.
(6) Objects with missing values in the attributes were removed.
(7) The data were normalized over the interval [0,1].

2.2. Problem characterization

Meta-attributes are common characteristics of several problems, and their aim is to identify structural similarities and differences among problems [12,75,76]. In 1992, the first works on problem characterization for meta-learning were published; these addressed, among others, classification problems; obtaining meta-attributes, such as the number of objects and classes; correlations between objects and classes; separability of classes; and entropy [2,13].

This characterization approach was later extended and is currently known as *direct characterization* [23] because the meta-attributes are extracted directly from the problems' attributes. These meta-attributes can be divided into three categories [13]:

- **Simple:** based on simple information, such as the number of objects, attributes, or classes.
- **Statistical:** based on statistical measures obtained from the problems' attributes, such as means, central moments, and correlations.
- **Informational:** based on information theory, such as entropy, mutual information, and noise.

In 2000, a new characterization method, called *landmarking*, was proposed, focusing on the use of simple classification algorithms to extract the meta-attributes [66]. This characterization is obtained by determining the accuracy of simple algorithms, such as Naïve Bayes and decision trees, applied directly to the problem. The idea is that the performance of classifiers is related to the intrinsic features of the problem; thus, classifiers with similar accuracy may indicate problems with similar characteristics.

Another characterization method is based on information extracted by *models* built out of the problems [8,39,65]. For instance, from a decision tree (model) constructed over a problem, it is possible to extract structural information about the tree itself, such as the number of nodes or leaves or the tree depth.

Characterization with the use of *landmarking* or *models* is known as *indirect characterization* because it is not directly related with the problems' attributes. This characterization makes use of a secondary methodology applied to the problem, in which the meta-attributes consist of information obtained from the results of their application [23]. Usually, when meta-learning is applied to the ASP, the problem characterization is crucial to learning the relationship between problem and performance [12].

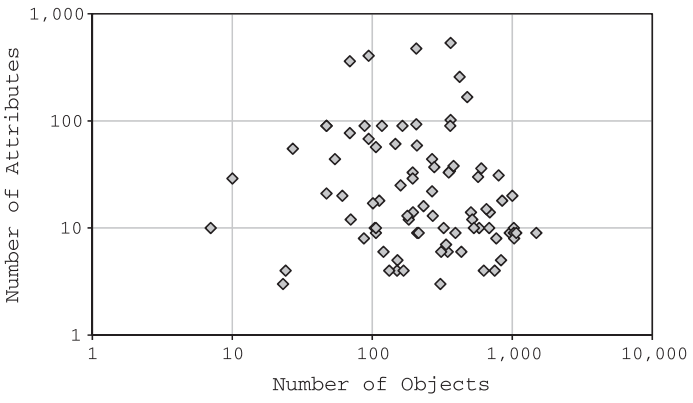

Fig. 1. The dimensionality of the problem collection.

In this study, the meta-attributes chosen to compose the traditional approach came from the *StatLog* project [57] and the work of Kalousis [48]. The meta-attributes described in the *StatLog* are taken as standard by the community on the characterization of classification problems [75,76]. Because there is no agreement on the meta-attribute set for clustering [10,12], the present work considers only those meta-attributes that do not need class information to be extracted from the problem.

Table 1 presents the meta-attributes based on the objects' attributes. An important property of this set is its unsupervised nature, i.e., its independence from class labels; thus, meta-attributes that measure the relationship between attributes and classes or the class separability were not used. The meta-attributes MA_1, MA_2, MA_3, and MA_4 extract global information about the problem; the meta-attributes MA_5 and MA_6 are extracted only for discrete attributes, and the meta-attributes MA_7, MA_8, and MA_9 only for continuous attributes. If a problem has only discrete attributes, then the meta-attributes for the continuous attributes will receive zero values, and vice versa. To automatically distinguish whether an attribute is discrete or not so as to calculate MA_3, the following rules were defined:

(1) If the attribute has real numbers, then the attribute is continuous.
(2) If the number of unique values is less than 30%, then the attribute is discrete.
(3) If otherwise, the attribute is continuous.

To extract MA_4, a method based on the *boxplot* is used [40,50]. The lower and upper limits for each attribute is calculated as follows:

$$\text{Lower Limit} = Q1 - (1,5 * IQR) \tag{1}$$

$$\text{Upper Limit} = Q3 + (1,5 * IQR) \tag{2}$$

where Q1 is the first quartile, Q3 is the third quartile, and IQR is the interquartile range (Q3–Q1). An object is considered an outlier if at least one of its attributes has a value outside its respective upper or lower limits.

2.3. Clustering algorithms

The collection of algorithms used to validate the proposed methods consists of seven clustering algorithms chosen to provide a broad and complementary analysis of the problems under different perspectives. These include two partitional algorithms (*K-means* and *K-harmonic means*), two bio-inspired methods with different search techniques (EAC and PSC), one hierarchical approach (*single-linkage*), one density-based approach (DBSCAN), and one graph-based algorithm (*minimum spanning tree*).

The *K-means* (KM) is the simplest and most popular partitional algorithm for clustering; besides its simplicity, easy encoding and efficiency are the main reasons for its popularity [45]. The number of clusters (k) is an input parameter, and the initial centroids have a major influence on the final solution [16,56].

The *K-harmonic means* (KHM) is a partitional algorithm in which the centroids are defined by using the harmonic mean [85,86]. In contrast to the standard KM, this algorithm is insensitive to the centroid initialization [82], which means that, independently of the centroid initial position, the solution is always the same. The number of clusters (k) is an input parameter.

The *evolutionary algorithm for clustering* (EAC) is a genetic algorithm with an automatic search method for the optimal number of clusters for a problem [4,41–43]. The variation proposed by Ferrari and de Castro [28] is used, with a modification that removes the transformation operator because its operation is costly and very similar to that of a local search mechanism. The EAC calculates the solution quality (*fitness*) by using a simplified version of the *silhouette* [42,43]:

$$S = \frac{1}{N}\sum_{i=1}^{N}\frac{b(i) - a(i)}{\max[a(i), b(i)]} \tag{3}$$

where N is the number of objects, $a(i)$ is the distance between object i and its centroid, and $b(i)$ is the smallest distance between object i and the other centroids. The value of S varies in the interval $[-1,+1]$ and is normalized to $[0,1]$, where 1 indicates the best possible clustering.

Table 1
Meta-attribute set based on the attributes.

Meta-attribute	Description
MA_1	Log2 of the number of objects
MA_2	Log2 of the number of attributes
MA_3	Percentage of discrete attributes
MA_4	Percentage of outliers
MA_5	Mean entropy of discrete attributes
MA_6	Mean concentration between discrete attributes
MA_7	Mean absolute correlation between continuous attributes
MA_8	Mean skewness of continuous attributes
MA_9	Mean kurtosis of continuous attributes

The *particle swarm clustering* (PSC) algorithm uses concepts from the emergent behavior of swarms to solve clustering problems [18]. The particles move about the space of objects and are positioned to become prototypes of the natural groups of data. Each particle represents the prototype of a single group, and the final solution involves all particles in the swarm. The number of clusters (k) is an input parameter.

Single-linkage (SL) clustering is a hierarchical algorithm that builds the solution starting with clusters with a single object, called *singletons*, which are then clustered together until there is only a single cluster containing all objects [60,81]. A heuristic is implemented with the SL clustering to automatically define the number of clusters based on the S value, as expressed in Eq. (3). From the SL hierarchical solution, the S value for the solution with two clusters is calculated. By moving through the solution, the S value is calculated for three clusters, then four clusters, and so on. Instead of making an exhaustive search over all possible solutions, a stopping criterion was designed: If the S value decreases for L consecutive iterations, the search is stopped, and the maximum S value found is chosen as the best solution.

Density-based spatial clustering of applications with noise (DBSCAN; herein called DBS) was developed for finding clusters with unusual formats and noise in the data sets [25]. The density notion is related to the number of objects within a neighborhood radius. The algorithm automatically defines the number of clusters. If the noise found by the DBS is greater than 10%, the algorithm is executed again with a 10% increase in the neighborhood radius. At the end of the process, the remaining noise is assigned to the nearest centroid.

The *minimum spanning tree* (MST) of an undirected graph is the tree that connects all nodes with the minimum weight possible [33]. A clustering algorithm based on the MST is capable of detecting clusters with irregular shapes and of automatically defining the number of clusters [33]. Different MST-based clustering algorithms can be found in Zhou et al. [87]. The data set objects are the graph nodes, and, initially, the graph is complete. In the implementation used here, Prim's algorithm [67] is applied to find the MST. The clustering algorithm used is based on the idea of finding and pruning inconsistent edges from the MST [83]. Each subtree found after pruning the inconsistent edges corresponds to a cluster.

The KM, KHM, and PSC algorithms need the number of clusters (k) as input parameter. To determine the value of k, the solutions of the EAC, SL, DBS, and MST were analyzed by using all the internal indices presented in the next section. Then the number of clusters of the best solution was used as the k value for the other algorithms. In case of a tie, the fastest algorithm was selected as the winner. All algorithms used the *Euclidean* distance as dissimilarity measure, and the stochastic algorithms were executed 30 times. Table 2 shows the parameters used by each algorithm.

2.4. Performance measures

Because clustering is an unsupervised task, most of the problems that it addresses do not have an a priori known solution. Thus, to assess the performance of the algorithms, it is necessary to use internal, instead of external, indices. Internal indices evaluate a clustering based only on the similarity matrix of the objects; several such indices have been proposed in the literature [35,36,55].

To evaluate the solution under different perspectives, this paper makes use of several internal indices, some of them based on inter-/intra-cluster measures. Intra-cluster measures are calculated for every cluster and give an idea of the cluster compactness. Inter-cluster measures are calculated between pairs of clusters and provide the degree of separation between clusters. Bezdek and Pal [9] made a study about the influence of different inter-/intra-cluster measures on several internal indices. Based on their study, the intra-cluster measure was chosen as the cluster diameter in this work, and the inter-cluster measure as the mean distance between the cluster objects:

$$Intra(\mathbf{g}_i) = \max_{\mathbf{x},\mathbf{y}\in\mathbf{g}_i}\{d(\mathbf{x},\mathbf{y})\} \tag{4}$$

Table 2
Parameterization of clustering algorithms.

KM	*PSC*
Maximum iterations: 50	Number of particles: k
Number of clusters: k	Maximum velocity: 1
KHM	
Maximum iterations: 50	Initial inertia value: 0.1
Weighting factor: 2,5	Inertial decay rate: 0.98
Number of clusters: k	Maximum iterations: 400
EAC	*DBS*
Population size: 4 individuals	Minimum number of points: 4
Initial number of clusters: 5	Neighborhood radius: 0.01
Local search maximum iterations: 2	Radius increase rate: 10%
Maximum iterations: 1000	Maximum noise level: 10%
Stability threshold: 0.001	
SL	*MST*
Consecutive iterations: 10	Path length: 2
	Standard deviation: 2.5

$$Inter(\mathbf{g}_i, \mathbf{g}_j) = \frac{1}{|\mathbf{g}_i||\mathbf{g}_j|} \sum_{\mathbf{x}\in\mathbf{g}_i, \mathbf{y}\in\mathbf{g}_j} d(\mathbf{x}, \mathbf{y}), \tag{5}$$

where $|\mathbf{g}_i|$ and $|\mathbf{g}_j|$ refer to the number of objects in clusters i and j, respectively; $\mathbf{x}$ and $\mathbf{y}$ are objects; and $d(\mathbf{x},\mathbf{y})$ is the distance between the objects. These measures were used to calculate the internal indices.

Based on the work of de Souza [23], which characterizes gene expression data sets, ten internal indices were chosen to evaluate the performance of the algorithms. These indices have different approaches to assessing the clustering quality, such as inter-/intra-cluster measures, distance between objects, and parity of objects, among others. All the internal indices use the *Euclidean* distance between the objects. Table 3 presents the internal indices and their corresponding domains and search objectives.

The algorithm performance step makes use of indices intended for quantitatively measuring the solution of a clustering problem and obtaining the algorithm performance. The algorithm performance is assessed only by using those internal indices that evaluate the solution based on, primarily, the similarity of objects. Thus, there is no need for previous knowledge of the label of each object. The performance is used as a target variable by the meta-learning system and can be represented as a numerical performance value, an identification of the best algorithm, or an algorithm ranking [12].

2.5. Ranking combination

In the representation based on ranking, the performance of the algorithms is ordered, and each algorithm is assigned to a rank position (ordinal value), with the first being the best and the last being the worst algorithm. This creates a numeric vector as target variable [15]. The rankings are classified according to two properties: linearity and completeness; linearity determines if the ranking can accept ties, and completeness refers to the number of algorithms in the ranking [12]. The use of algorithm rankings for the ASP has received some attention from the machine learning community due to its flexibility because the ranking provides more options compared with selecting only the best algorithm [12].

Meta-learning systems are usually applied to classification problems, with the performance evaluated by a single measure (classification error); in some cases, the running time is combined [12,15,75,76]. Having the objective of minimizing both (classification error and execution time), the combination is made by using a weighted mean that can be set to recommend better or faster algorithms. To the best of the authors' knowledge, thus far, meta-learning systems applied to clustering problems have been assessed only by a single external index.

The combination of various internal indices aims to qualify the solution with different perspectives provided by different evaluation measures. However, most of these indices do not share the same domain interval or optimal values. To solve the combination problem, the values for the internal indices are ranked; i.e., for each internal index, the performance of the algorithms is transformed into a ranking representation. This ranking corresponds to a performance order for different algorithms over the same problem [14,22]. A complete nonlinear ranking representation is applied, in which all the algorithms are used and the presence of ties is allowed [12].

The outcome for each clustering problem in the collection consists of 10 different rankings, one for each internal index, with 7 positions, one for each algorithm. Using all these data to select an algorithm for a new, unseen problem could make the ASP even more difficult. Thus, the ranking combination methods are used to merge all 10 rankings, and the resultant combined rankings are applied by the selection mechanism.

Average ranking is a combination method usually applied to meta-learning as a reference or ideal ranking in the algorithm selection process, in which the search for the relationship between meta-attributes and performance requires methods or algorithms that provide better solutions than the average ranking between problems [12,14,22]. The average ranking can be obtained as follows:

$$\bar{\boldsymbol{r}}_i = \frac{1}{k}\sum_{j=1}^{k} \boldsymbol{r}_{ij} \tag{6}$$

Table 3
The internal indices and their respective domains and search objectives.

Internal index	Interval	Objective
BP [9]	$[0;+\infty]$	Max
DU [24]	$[0;+\infty]$	Max
CH [17]	$[0;+\infty]$	Max
SIL [73]	$[-1;+1]$	Max
GK [7]	$[-1;+1]$	Max
MC [58]	$[-1;+1]$	Max
DB [21]	$[0;+\infty]$	Min
HKK [38]	$[0;+\infty]$	Min
FR [30]	$[0;+1]$	Min
HL [44]	$[0;+1]$	Min

where $\bar{\boldsymbol{r}}_i$ is the mean rank position value for the i-th algorithm, k is the number of rankings, and $\boldsymbol{r}_{ij}$ is the rank position for the i-th algorithm and the j-th ranking. After calculating the mean rank value, it is necessary to build the average ranking, in which the lowest value occupies the first position and the highest value holds the last position.

Because the average ranking is commonly used in meta-learning systems, it is applied as a combination method to merge the individual rankings (internal indices rankings) to build a single ranking for each problem. Furthermore, the average ranking is used as a reference for the new ranking combination methods proposed in this paper.

3. Novelty in the meta-knowledge for clustering

The present paper aims to approach the ASP for clustering tasks through the application of meta-learning systems, maintaining the unsupervised nature of the task. To achieve this, a new problem characterization is proposed based on the distances among the objects in the database, and two new ranking combination methods are introduced.

3.1. Distance-based characterization

In contrast to classification problems, in which the learning process is guided by the classes, in clustering problems, the learning is usually based on the similarity among objects. In the search for the best groups, the information is encoded according to similarity measures, which have a direct influence on the clustering process [1,81].

The new meta-attributes proposed in this paper are based on the distance between the objects of the clustering problem instead of on the object attributes themselves. The distance indicates how dissimilar the objects are based on their attribute values [37,63,79]. There is no need for any label information to extract these meta-attributes; thus, the method proposed here can be generically applied to any clustering problem. The proposed characterization begins with the construction of a vector $\mathbf{d}$ containing the distance among all objects:

$$\mathbf{d} = [d_{1,2}, d_{1,3}, \ldots, d_{i,j}, \ldots, d_{n-2,n-1}, d_{n-1,n}] \tag{7}$$

where n is the number of objects, and $d_{i,j}$ is the *Euclidean* distance between the i-th and j-th objects. The vector is normalized in the interval [0,1], and 19 meta-attributes are extracted (Table 4).

This approach is categorized as an indirect characterization with the use of statistical meta-attributes because the distance vector can be treated as a continuous random variable. The statistical measures used in the characterization based on the attributes aim to capture information about the structure of the problem. These same statistical measures can be used to characterize the distance vector.

The first meta-attributes proposed are extracted by using some basic statistical measures. The mean, variance, standard deviation, skewness, and kurtosis of the distance vector, which form MD_1 to MD_5, respectively, are calculated. The mean value provides information about the center of the distribution function of $\mathbf{d}$. The variance and standard deviation complement the mean by indicating the dispersion degree of the objects in the space, and the skewness and kurtosis indicate the shape of the probability distribution function based on the distance vector.

Kalousis [48] made an extensive evaluation of 98 meta-attributes for classification problems. The author supported the use of histograms to capture the most information possible about the problem being characterized. Following this idea, the next 14 meta-attributes proposed here are based on two different histograms extracted from the distance vector.

Table 4
Meta-attribute set based on the similarity among objects.

Meta-attribute	Description
MD_1	Mean of $\mathbf{d}$
MD_2	Variance of $\mathbf{d}$
MD_3	Standard deviation of $\mathbf{d}$
MD_4	Skewness of $\mathbf{d}$
MD_5	Kurtosis of $\mathbf{d}$
MD_6	% of values in the interval [0,0.1]
MD_7	% of values in the interval (0.1,0.2]
MD_8	% of values in the interval (0.2,0.3]
MD_9	% of values in the interval (0.3,0.4]
MD_{10}	% of values in the interval (0.4,0.5]
MD_{11}	% of values in the interval (0.5,0.6]
MD_{12}	% of values in the interval (0.6,0.7]
MD_{13}	% of values in the interval (0.7,0.8]
MD_{14}	% of values in the interval (0.8,0.9]
MD_{15}	% of values in the interval (0.9,1.0]
MD_{16}	% of values with absolute Z-score in the interval [0,1)
MD_{17}	% of values with absolute Z-score in the interval [1,2)
MD_{18}	% of values with absolute Z-score in the interval [2,3)
MD_{19}	% of values with absolute Z-score in the interval $[3,\infty)$

To illustrate the distance distribution, a histogram is built with 10 bins of equal size for the distance vector, with the boundaries [0.0, 0.1], (0.1, 0.2], (0.2, 0.3], (0.3, 0.4], (0.4, 0.5], (0.5, 0.6], (0.6, 0.7], (0.7, 0.8], (0.8, 0.9], (0.9, 1.0]. By using the histogram, the relative frequency for each bin is calculated and associated with ten meta-attributes, from MD_6 to MD_{15}, respectively.

The last four meta-attributes (MD_{16}–MD_{19}) are extracted from the histogram of the absolute *Z*-score obtained from the distance vector to find the normality of the vector. The *Z*-score represents how many standard deviations away an element is from the distribution mean value and is calculated as:

$$z = \frac{x - \mu}{\sigma} \tag{8}$$

where x is the element value, μ is its mean value, and σ is its standard deviation. The absolute *Z*-score value is discretized into four bins, with the boundaries [0, 1), [1, 2), [2, 3), and [3, ∞). Thus, each meta-attribute, from MD_{16} to MD_{19}, respectively, indicates the relative frequency for each bin.

3.2. Ranking combination methods

The individual rankings of the internal indices need to be merged into one single ranking. For this purpose, the present paper proposes two new methods: *score ranking* and *winner ranking*.

The *score ranking* method is based on race tournaments, in which the pilots receive points at the end of each race. At the end of the tournament, the pilot (algorithm) with more points is declared the winner. In this technique, each algorithm receives a number of points according to its position in each index ranking. These points are totaled, and the final ranking is built in descending order. Table 5 shows the number of points assigned to each rank position.

The *winner ranking* method is based on the number of victories in a pairwise competition between the algorithms, taking into account all internal indices. The algorithm with the largest number of victories occupies the first rank position, and the algorithm with the smallest number of victories holds the last position. The number of victories can be obtained as follows:

$$\boldsymbol{v}_i = \sum_{j=1}^{m} |r| - \boldsymbol{p}_{ij} \tag{9}$$

where $\boldsymbol{v}_i$ is the number of victories of the i-th algorithm, m is the number of internal indices, $|r|$ is the number of algorithms in the ranking, and $\boldsymbol{p}_{ij}$ is the rank position for the i-th algorithm in the j-th internal index.

To illustrate how the combination ranking methods work, consider the Servo problem from the collection. The first lines in Table 6 present the individual rankings for the Servo problem. The rankings show the positions of the algorithms in the internal index used; for example, in the DB index, the algorithms SL and DBS share the first and second positions, followed by the EAC, PSC, KHM, KM, and EMST, respectively.

The average ranking is obtained by calculating the mean position value for each algorithm on all indices and then reassigning the rank positions. For example, the PSC algorithm has a mean position value of 2.7 in the individual rankings. After calculating the other mean positions, it becomes necessary to rebuild the ranking such that the PSC, which has the lowest value, is assigned as the best algorithm in the average ranking.

In the score ranking, each position is associated with a number of points. Thus, for each algorithm, the points obtained in the individual rankings are totaled, and then the algorithm with the highest total is considered the best algorithm. The PSC, which has a total of 69 points, is ranked first, followed by the SL and DBS (which share the 2nd and 3rd positions, respectively).

In the winner ranking, the number of victories of each algorithm in each individual ranking is calculated, and the victories are totaled. The PSC achieved 43 victories, followed by the EAC with 41.

4. Experimental evaluation

To assess the proposed distance-based characterization and ranking schemes, two different analyses were done with the meta-knowledge database. The first one investigates the complexity in obtaining the meta-attributes for both methods. The second analysis assesses the viability of applying the meta-knowledge to the ASP.

Table 5
Number of points for each rank position.

Rank	1	2	3	4	5	6	7
Points	10	8	6	4	3	2	1

Table 6
Individual and combined rankings for the Servo problem. The combined ranking are highlighted.

Internal indices	Clustering algorithms						
	KM	KHM	EAC	SL	PSC	DBS	EMST
Individual ranking							
DB	6	5	3	1.5	4	1.5	7
HKK	6	5	4	1.5	3	1.5	7
FR	7	5	6	1.5	4	1.5	3
HL	4	3	1	5.5	2	5.5	7
DU	4	2	1	5.5	3	5.5	7
BP	6	5	4	1.5	3	1.5	7
CH	4	2	3	5.5	1	5.5	7
SIL	6	5	3	1.5	4	1.5	7
GK	6	3	1	4.5	2	4.5	7
MC	4	2	3	5.5	1	5.5	7
Combined ranking							
Mean value	5.3	3.7	2.9	3.4	2.7	3.4	6.6
Average ranking	**6**	**5**	**2**	**3.5**	**1**	**3.5**	**7**
Sum of points	31	51	66	67	69	67	15
Score ranking	**6**	**5**	**4**	**2**	**1**	**2**	**7**
Number of victories	17	33	41	36	43	36	4
Winner ranking	**6**	**5**	**2**	**3.5**	**1**	**3.5**	**7**

4.1. Assessing the complexity of the problem characterization methods

Fig. 2 shows the characterization processing time for the attribute-based and distance-based methods, considering the number of objects and attributes (dimensions) of the problems. The attribute-based is highly influenced by the number of attributes, while the distance-based is influenced by the number of objects. It can be observed that with a lower number of objects and attributes, both characterization methods have similar processing times. As the number of attributes increases, the impact on the attribute-based characterization is much higher compared with the increase of objects on the distance-based approach. It is worth mentioning that the distance-based methods have almost twice the number of meta-attributes.

4.2. Assessing the meta-knowledge for the algorithm selection problem

A meta-learning approach to the ASP can consist of a recommendation algorithm that selects the most suitable algorithm ranking based on the meta-knowledge database [12]. To recommend the algorithm ranking, a well-known classification algorithm, the *k-Nearest Neighbors* (kNN), is used due to its simplicity, speed, and recommendation quality [10,15]. The kNN is an instance-based algorithm that classifies an object based on a search for its nearest neighbors [3,59].

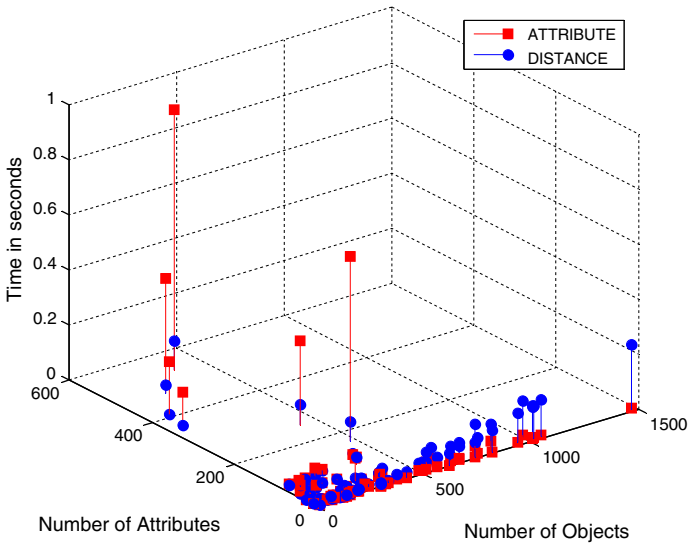

Fig. 2. The characterization processing time for both methods.

When a new problem is presented to the meta-learning system, the meta-attributes are extracted. Then the kNN algorithm is applied to find the nearest neighbors (most similar problems) in the meta-knowledge database based only on the meta-attributes. When the problems are found, a ranking aggregation technique merges their rankings and gives the ranking for the new problem. In the present paper, the recommendation mechanism uses the kNN algorithm with the average ranking [12].

In the experiments, the *Euclidean* distance [31] is used as a measure to find the nearest neighbors of an object, and the number of neighbors is varied from 1 to 11 to find the most suitable value. The recommendation process is evaluated by a 10-fold cross-validation method with approximately the same number of problems. One by one, each fold is used as a test set to assess the recommendation quality, whereas the remaining nine folds are the instances for the kNN. The process is executed 30 times, with the 10 folds randomly built each time.

To evaluate the similarity between the recommended and the ideal rankings, *Spearman's rank correlation* (SRC) is used, which assumes values over the interval $[-1,+1]$, where +1 indicates that the rankings are equal [62]:

$$\mathrm{SRC}(\mathbf{rr},\mathbf{ir}) = 1 - \frac{6\sum_{i=1}^{p}(\mathbf{rr}_i - \mathbf{ri}_i)^2}{p^3 - p} \tag{10}$$

where **rr** and **ri** are the recommended and ideal rankings, respectively; and p is the number of rank positions. The SRC has been used for ranking comparison in several meta-learning works [12,15,22,77].

The experiments are done with three sets of meta-attributes: (1) attribute-based, (2) distance-based, and (3) hybrid, with the latter formed by the concatenation of the first two sets. The three ranking combination methods are compared with each other and with the *standard* ranking of each approach. The *standard ranking*, or *ideal ranking*, is obtained by the same method as the average ranking but uses the ranking of the problems to build a single ranking to represent the meta-knowledge database. Table 7 shows the standard ranking for each combination method. For each standard ranking, the SRC value for all rankings in the meta-knowledge database is calculated; the mean value is used as reference in assessing the kNN recommendation.

Fig. 3(a)–(c) shows the SRC results for the kNN recommendation, with the number of neighbors varied for the average, score and winner ranking schemes, respectively. The horizontal dotted line represents the mean SRC value for the standard ranking for all rankings in the meta-knowledge and serves as a reference for the kNN quality recommendation; values greater than the standard recommendation indicate better quality, and vice versa.

As shown in the figure, when only one neighbor is used, the combinations of meta-attribute and ranking were worse than the SRC value for the standard ranking. When two neighbors were used, almost all combinations reached SRC values equal to the standard ranking. When more than three neighbors were used, all SRC values were better than the standard ranking. The results for the kNN with five neighbors (5NN) were chosen for a closer examination because the SRC presented a much smaller variation in the increase in k for $k \geqslant 5$ than for smaller values.

Table 8 shows the SRC values for a recommendation of 5NN with the *Euclidean* distance. The SRC has a significance test tabulated with the null hypothesis (H_0) that there is no correlation between the rankings. The alternative hypothesis (H_1) is that there is a positive correlation, or agreement, between the rankings [84]. For a sample of size 30 and at 1% significance level, two rankings are considered to have a positive correlation if they reach an SRC value above 0.425. All the results obtained showed a positive correlation.

A statistical test was done on the SRC results over 30 executions to determine the normality of the data. The *Lilliefors* test was applied to the null hypothesis (H_0) that the sample comes from a normally distributed population, without specifying the expected value and variance of the distribution [53]. At 1% significance level, all the results showed that the values presented came from a normal distribution. To verify the difference between the mean values of the SRC results, a *t*-test [11,19] was done, which verified the null hypothesis (H_0) that the samples come from normal distributions with equal means. The following analyses were done at 1% significance level.

Concerning the ranking combination method, no significant difference was found between the attribute-based and the hybrid set. When the distance-based set was used, the average and winner rankings showed no difference, but the score ranking achieved better results than the others. Comparing the different sets of meta-attributes, only the average and winner rankings with the attribute-based set showed no difference; all other combinations were found to have statistically different results. Based on these findings and the statistical analyses, the distance-based meta-attribute set presented a better recommendation than did all the other schemes, especially when the proposed score ranking combination method was used.

Table 7
The standard ranking of all problems for each ranking combination method.

Ranking	KM	KHM	EAC	SL	PSC	DBS	EMST
Average	4	5	2	1	3	6	7
Score	4	5	2	1	3	6	7
Winner	4	5	2	1	3	6	7

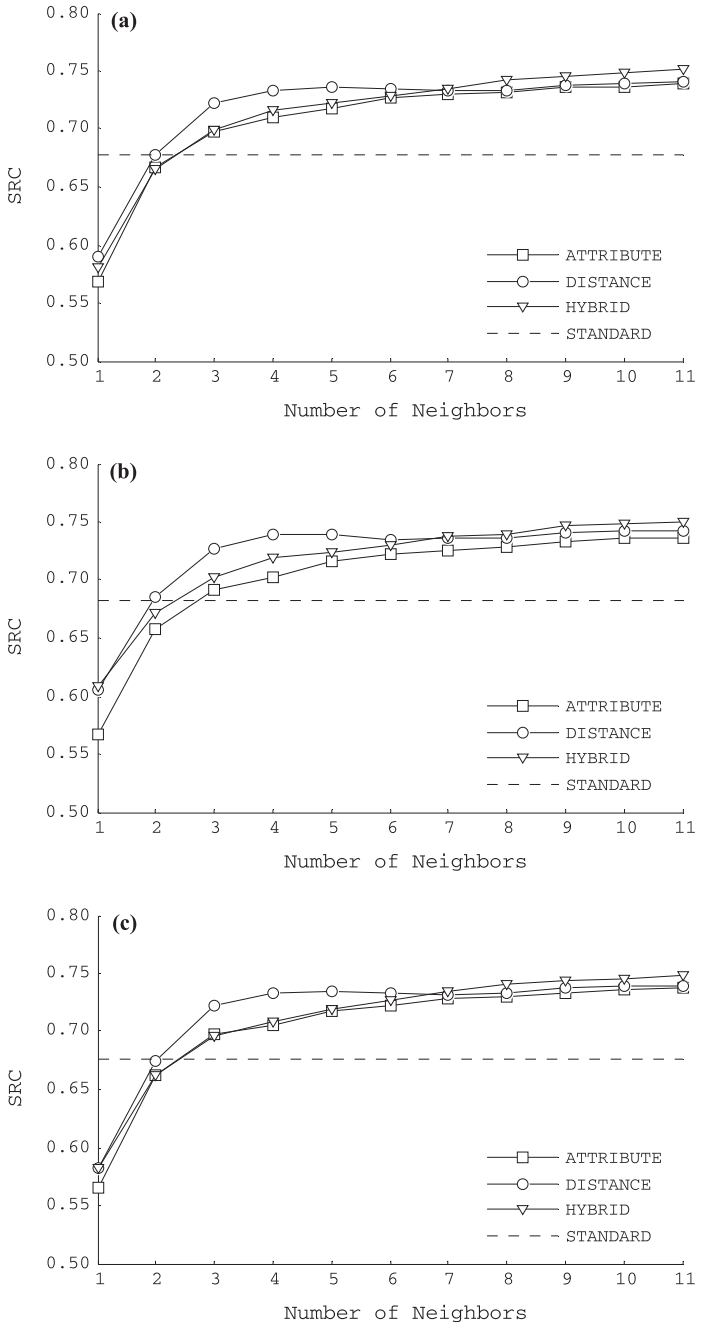

Fig. 3. SRC values in relation to the number of neighbors: (a) average ranking, (b) score ranking, and (c) winner ranking.

Table 8
SRC values (mean ± standard deviation) for the ranking recommendation of 5NN with different meta-attributes sets, with the standard ranking as reference. Best values are highlighted.

Ranking	Standard raking	Attribute-based	Distance-based	Hybrid
Average	0.678 ± 0.310	0.718 ± 0.007	**0.736 ± 0.005**	0.722 ± 0.005
Score	0.683 ± 0.297	0.716 ± 0.006	**0.739 ± 0.006**	0.723 ± 0.005
Winner	0.676 ± 0.301	0.718 ± 0.007	**0.735 ± 0.005**	0.719 ± 0.006

5. Conclusions and future works

Clustering is an unsupervised task that searches for natural clusters within data sets, which is a complex combinatorial problem. The literature about clustering includes a large number of algorithms based on many different techniques that involve adjusting countless parameters, thus making the selection of the best algorithm an extremely hard process. Following the formulation of the ASP and the development of meta-learning systems, various works have been done in search of a methodology to support the algorithm selection process.

The present work proposed the analysis of the ASP for clustering by using meta-learning systems, specifically through the introduction of a new problem characterization mechanism and two ranking combination schemes. From the clustering problems, three sets of meta-attributes were extracted; the first uses a traditional direct characterization method based on the problems' attributes, the second applies a proposed indirect method based on the distance among objects, and the third combines the first two methods. The algorithm solutions were evaluated using 10 internal measures combined by three different techniques: average, score, and winner ranking.

At the end of the process, a meta-knowledge database was built, linking the meta-attributes of a problem with the rankings of the algorithms for the same problem. To verify the existence of a relation between the meta-attributes and the rankings, the database was evaluated with a recommendation process by using kNN. The results showed that, independently of the meta-attribute set and ranking combination method used, it is possible to recommend a high-quality ranking for clustering algorithms.

The proposed meta-attribute set (distance-based) presented better results than the classic approach (attribute-based), along with a higher recommendation quality, a shorter processing time per meta-attribute, and a lower impact with an increase in the problems' dimensionality. The experiment results indicated the viability of meta-learning systems for an unlabeled approach to the clustering algorithm selection problem. The characterization based on meta-attributes that do not depend on the object labels and the algorithm evaluation by using internal indices that avoid known solutions allow the proposed methodology to be applied to any clustering problem.

Further studies should be done regarding new meta-attributes that can be extracted from the distance vector, as well as the use of other similarity measures to build the vector. Variations in the recommendation technique may also improve the results.

Acknowledgements

The authors thank MackPesquisa, CNPq, CAPES (Proc. n. 9315/13-6), and FAPESP (#2011/21816-3) for the financial support, and the CENAPAD-SP for the computational facilities.

Appendix A

A.1. List of UCI data sets used

Acute inflammations, PubChem Bioassay AID1284Morered, PubChem Bioassay AID1284red, PubChem Bioassay AID439Morered, PubChem Bioassay AID439red, PubChem Bioassay AID644Morered, PubChem Bioassay AID644red, PubChem Bioassay AID721morered, PubChem Bioassay AID721red, Annealing, Arrhythmia, Audiology, Australian credit approval, Automobile, Auto mpg, Balance scale, Cylinder bands, Breast cancer Wisconsin (original), Breast tissue, Pittsburgh bridges, Liver disorders, Musk v1, Could 1, Could 2, Vertebral column, Concrete compressive strength, Credit approval, Dermatology, Echocardiogram, E. coli, Flags, Solar flare 1, Solar flare 2, Forest fires, German credit, Glass, Haberman's survival, Hayes-Roth, Heart, Hepatitis, Horse colic, Congressional voting records, Housing, Ionosphere, Iris, Labor relations, Lenses, Indian liver patient, Robot failures 1, Robot failures 2, Robot failures 3, Robot failures 4, Robot failures 5, Low resolution spectrometer, Lung cancer, Computer hardware, Mammographic mass, Monk's problems, Libras movement, Challenger space shuttle, Parkinson's disease, Pima Indians diabetes, Planning relax, Post-operative patient, Promoter gene sequences, Servo, Concrete slump test, Sonar: mines vs rocks, Soybean small, SPECT heart, SPECTF heart, Sponge, Synthetic control chart, Teaching assistant eval, Tic-tac-toe, Trains, Blood transfusion, Vehicle silhouettes, Water treatment plant, Breast cancer Wisconsin (diagnostic), Wine, Breast cancer Wisconsin (prognostic), Yeast, Zoo.

References

[1] C.C. Aggarwal, C.K. Reddy, Data Clustering: Algorithms and Applications, Chapman and Hall/CRC, 2013.
[2] D.W. Aha, Generalizing from case studies: a case study, in: Proceedings of the Ninth International Workshop on Machine Learning, Morgan Kaufmann Publishers Inc., 1992, pp. 1–10.
[3] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1991) 37–66.
[4] V.S. Alves, R.J.G.B. Campello, E.R. Hruschka, Towards a fast evolutionary algorithm for clustering, in: IEEE Congress on Evolutionary Computation, 2006, CEC 2006, 2006, pp. 1776–1783.
[5] B. Arinze, S.-L. Kim, M. Anandarajan, Combining and selecting forecasting models using rule based induction, Comput. Oper. Res. 24 (1997) 423–433.
[6] K. Bache, M. Lichman, UCI Machine Learning Repository, University of California, School of Information and Computer Science, 2013.
[7] F.B. Baker, L.J. Hubert, Measuring the power of hierarchical cluster analysis, J. Am. Statist. Assoc. 70 (1975) 31–38.
[8] H. Bensusan, C. Giraud-Carrier, C. Kennedy, A higher-order approach to meta-learning (2000).

[9] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Trans. Syst. Man. Cybern. B Cybern. 28 (1998) 301–315.
[10] N. Bhatt, A. Thakkar, A. Ganatra, A survey & current research challenges in meta learning approaches based on dataset characteristics, Int. J. Soft Comput. Eng. 2 (2012) 239–247.
[11] G.E.P. Box, W.G. Hunter, J.S. Hunter, Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building (Wiley Series in Probability and Statistics), John Wiley & Sons, 1978.
[12] P. Brazdil, C.G. Carrier, C. Soares, R. Vilalta, Metalearning: Applications to Data Mining, Springer, 2008.
[13] P. Brazdil, R. Henery, Analysis of results, in: D. Michie, D.J. Spiegelhalter, C.C. Taylor (Eds.), Machine Learning, Neural and Statistical Classification, Ellis Horwook, New York, 1994.
[14] P. Brazdil, C. Soares, A comparison of ranking methods for classification algorithm selection, in: Proceedings of the 11th European Conference on Machine Learning, Springer-Verlag, 2000, pp. 63–74.
[15] P.B. Brazdil, C. Soares, J.P. da Costa, Ranking learning algorithms: using IBL and meta-learning on accuracy and time results, Mach. Learn. 50 (2003) 251–277.
[16] S. Bubeck, M. Meila, U. von Luxburg, How the initialization affects the stability of the k-means algorithm (2009).
[17] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. 3 (1974) 1–27.
[18] S.C.M. Cohen, L.N. de Castro, Data clustering with particle swarms, in: IEEE Congress on Evolutionary Computation, 2006, CEC 2006, 2006, pp. 1792–1798.
[19] J. Crawshaw, J. Chambers, A Concise Course in Advanced Level Statistics: With Worked Examples, Nelson Thornes, 2001.
[20] L. Cruz-Reyes, C. Gómez-Santillán, J. Pérez-Ortega, V. Landero, M. Quiroz, A. Ocho, Algorithm Selection: From Meta-Learning to Hyper-Heuristics, in: V.M. Koleshko (Ed.), Intelligent Systems, InTech, 2012.
[21] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (1979) 224–227.
[22] M.C.P. de Souto, R.B.C. Prudencio, R.G.F. Soares, D.S.A. de Araujo, I.G. Costa, T.B. Ludermir, A. Schliep, Ranking and selecting clustering algorithms using a meta-learning approach, in: IEEE International Joint Conference on Neural Networks, 2008, IJCNN 2008 (IEEE World Congress on Computational Intelligence), 2008, pp. 3729–3735.
[23] B.F. De Souza, Meta-aprendizagem aplicada à classificação de dados de expressão gênica (Ph.D. thesis), in: ICMC, University of São Paulo, 2010.
[24] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J. Cybernet. 3 (1973) 32–57.
[25] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 226–231.
[26] D.G. Ferrari, L.N. De Castro, Clustering algorithm recommendation: a meta-learning approach, in: Proceedings of the Third International Conference on Swarm, Evolutionary, and Memetic Computing, Springer-Verlag, 2012, pp. 143–150.
[27] D.G. Ferrari, L.N. de Castro, Performance estimation for clustering algorithms with meta-learning techniques, in: 19th Brazilian Conference on Automation, 2012, pp. 2380–2386.
[28] D.G. Ferrari, L.N. de Castro, New genetic operators for the evolutionary algorithm for clustering, in: 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013, pp. 55–59.
[29] A. Frank, A. Asuncion, UCI Machine Learning Repository, University of California, School of Information and Computer Science, 2010.
[30] J.H. Friedman, L.C. Rafsky, Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests, Ann. Statist. 7 (1979) 697–717.
[31] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability), SIAM, Society for Industrial and Applied Mathematics, 2007.
[32] C. Giraud-Carrier, R. Vilalta, P. Brazdil, Introduction to the special issue on meta-learning, Mach. Learn. 54 (2004) 187–193.
[33] O. Grygorash, Y. Zhou, Z. Jorgensen, Minimum spanning tree based clustering algorithms, in: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society, 2006, pp. 73–81.
[34] H. Guo, Algorithm Selection for Sorting and Probabilistic Inference: A Machine Learning-Based Approach, Kansas State University, 2003 (256).
[35] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inf. Syst. 17 (2001) 107–145.
[36] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Cluster validity methods: part I, ACM SIGMOD Rec. 31 (2002) 40–45.
[37] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011.
[38] J. Handl, J. Knowles, D.B. Kell, Computational cluster validation in post-genomic data analysis, Bioinformatics 21 (2005) 3201–3212.
[39] M. Hilario, A. Kalousis, Building algorithm profiles for prior model selection in knowledge discovery systems, in: 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999, IEEE SMC '99 Conference Proceedings, vol. 953, 1999, pp. 956–961.
[40] V. Hodge, J. Austin, A survey of outlier detection methodologies, Artif. Intell. Rev. 22 (2004) 85–126.
[41] E.R. Hruschka, R.G.B. Campello, L.N. De Castro, Improving the efficiency of a clustering genetic algorithm, in: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004, Springer, Berlin Heidelberg, 2004, pp. 861–870.
[42] E.R. Hruschka, R.J.G.B. Campello, L.N. De Castro, Evolving clusters in gene-expression data, Inf. Sci.: Int. J. 176 (2006) 1898–1927.
[43] E.R. Hruschka, L.N.D. Castro, R.J.G.B. Campello, Evolutionary algorithms for clustering gene-expression data, in: Proceedings of the Fourth IEEE International Conference on Data Mining, IEEE Computer Society, 2004, pp. 403–406.
[44] L.J. Hubert, J.R. Levin, A general statistical framework for assessing categorical clustering in free recall, Psychol. Bull. 83 (1976) 1072.
[45] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recogn. Lett. 31 (2010) 651–666.
[46] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. (CSUR) 31 (1999) 264–323.
[47] N. Jankowski, Complexity measures for meta-learning and their optimality, in: D.L. Dowe (Ed.), Algorithmic Probability and Friends – Bayesian Prediction and Artificial Intelligence, Springer, Berlin Heidelberg, 2013, pp. 198–210.
[48] A. Kalousis, Algorithm Selection via Meta-Learning (Ph.D. thesis), University of Geneve, Department of Computer Science, 2002.
[49] Y. Kim, C. Hong, A meta-learning approach for combining multiple classifiers, Adv. Sci. Technol. Lett. 29 (2013) 50–54.
[50] J. Laurikkala, M. Juhola, E. Kentala, Informal identification of outliers in medical data, in: Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology, 1998, pp. 20–24.
[51] J.-S. Lee, S. Olafsson, A meta-learning approach for determining the number of clusters with consideration of nearest neighbors, Inf. Sci. 232 (2013) 208–224.
[52] C. Lemke, M. Budka, B. Gabrys, Metalearning: a survey of trends and technologies, Artif. Intell. Rev. (2013) 1–14
[53] H.W. Lilliefors, On the Kolmogorov–Smirnov test for normality with mean and variance unknown, J. Am. Stat. Assoc. 62 (1967) 399–402.
[54] X. Liu, X. Wang, S. Matwin, N. Japkowicz, Meta-learning for large scale machine learning with MapReduce, in: IEEE International Conference on Big Data, IEEE, Silicon Valley, CA, 2013, pp. 105–110.
[55] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, Understanding of internal clustering validation measures, in: Proceedings of the 2010 IEEE International Conference on Data Mining, IEEE Computer Society, 2010, pp. 911–916.
[56] J. Macqueen, Some methods for classification and analysis of multivariate observations, in: 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.
[57] D. Michie, D.J. Spiegelhalter, C.C. Taylor, J. Campbell, Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994.
[58] G.W. Milligan, M.C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika 50 (1985) 159–179.
[59] K.P. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, 2012.
[60] F. Murtagh, A survey of recent advances in hierarchical clustering algorithms, Comp. J. 26 (1983) 354–359.
[61] A.C. Nascimento, R.B. Prudêncio, M.C. Souto, I.G. Costa, Mining rules for the automatic selection process of clustering methods applied to cancer gene expression data, in: Proceedings of the 19th International Conference on Artificial Neural Networks: Part II, Springer-Verlag, 2009, pp. 20–29.
[62] H.R. Neave, P.L. Worthington, Distribution-free Tests, London, 1992.

[63] R. Nisbet, J. Eldder, G. Miner, Handbook of Statistical Analysis and Data Mining Applications, Academic Press, 2009.
[64] G. Pappa, G. Ochoa, M.R. Hyde, A.A. Freitas, J. Woodward, J. Swan, Contrasting meta-learning and hyper-heuristic research: the role of evolutionary algorithms, Genet. Program Evolvable Mach. 15 (2014) 3–35.
[65] Y. Peng, P.A. Flach, P. Brazdil, C. Soares, Decision tree-based data characterization for meta-learning, in: European Conference on Machine Learning, 2002.
[66] B. Pfahringer, H. Bensusan, C. Giraud-Carrier, Meta-learning by landmarking various learning algorithms. in: Proceedings of the 17th International Conference on Machine Learning, ICML'2000, Morgan Kaufmann, 2000, pp. 743–750.
[67] R.C. Prim, Shortest connection networks and some generalizations, Bell Syst. Tech. J. 36 (1957) 1389–1401.
[68] R.B.C. Prudêncio, T.B. Ludermir, Meta-learning approaches to selecting time series models, Neurocomputing 61 (2004) 121–137.
[69] M. Reif, F. Shafait, M. Goldstein, T. Breuel, A. Dengel, Automatic classifier selection for non-experts, Pattern Anal. Appl. 17 (2014) 83–96.
[70] J.R. Rice, The algorithm selection problem, Adv. Comp. 15 (1976) 65–118.
[71] P. Ridd, C. Giraud-Carrier, Using metalearning to predict when parameter optimization is likely to improve classification accuracy, in: 21st European Conference on Artificial Intelligence, Czech Republic, 2014, pp. 18–23.
[72] A.L.D. Rossi, A.C.P.L.F. De Carvalho, C. Soares, B.F. De Souza, MetaStream: a meta-learning based method for periodic algorithm selection in time-changing data, Adv. Intell. Syst. 127 (2014) 52–64.
[73] P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[74] M.R. Smith, L. Mitchell, C. Giraud-Carrier, T. Martinez, Recommending learning algorithms and their associated hyperparameters, in: 21st European Conference on Artificial Intelligence, Czech Republic, 2014, pp. 39–40.
[75] K.A. Smith-Miles, Cross-disciplinary perspectives on meta-learning for algorithm selection, ACM Comput. Surv. 41 (1) (2009) 1–25, http://dx.doi.org/10.1145/1456650.1456656.
[76] K.A. Smith-Miles, Towards insightful algorithm selection for optimisation using meta-learning concepts, in: IEEE International Joint Conference on Neural Networks, 2008, IJCNN 2008 (IEEE World Congress on Computational Intelligence), 2008, pp. 4118–4124.
[77] R.G. Soares, T.B. Ludermir, F.A. Carvalho, An analysis of meta-learning techniques for ranking clustering algorithms applied to artificial data, in: Proceedings of the 19th International Conference on Artificial Neural Networks: Part I, Springer-Verlag, 2009, pp. 131–140.
[78] Q. Sun, B. Pfahringer, Pairwise meta-rules for better meta-learning-based algorithm ranking, Mach. Learn. 93 (2014) 141–161.
[79] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison-Wesley, 2005.
[80] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[81] R. Xu, D. Wunsch II, Survey of clustering algorithms, IEEE Trans. Neural Networks 16 (2005) 645–678.
[82] S.M.R. Zadegan, M. Mirzaie, F. Sadoughi, Ranked k-medoids: a fast and accurate rank-based partitioning algorithm for clustering large datasets, Knowl.-Based Syst. 39 (2013) 133–143.
[83] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. 20 (1971) 68–86.
[84] J.H. Zar, Significance testing of the spearman rank correlation coefficient, J. Am. Statist. Assoc. 67 (1972) 578–580.
[85] B. Zhang, Generalized K-Harmonic Means – Boosting in Unsupervised Learning, Hewlett-Packard Laboratories, 2000. p. 13.
[86] B. Zhang, M. Hsu, U. Dayal, K-harmonic means – a spatial clustering algorithm with boosting, in: Proceedings of the First International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining-Revised Papers, Springer-Verlag, 2000, pp. 31–45.
[87] Y. Zhou, O. Grygorash, T.F. Hain, Clustering with minimum spanning trees, Int. J. Artif. Intell. Tools 20 (2011) 139–177.