Knowledge-Based Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A Meta-learning approach for recommending the number of clusters for clustering algorithms☆

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

Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP), São Carlos, Brazil

ARTICLE INFO

Article history:
Received 10 September 2019
Received in revised form 19 February 2020
Accepted 20 February 2020
Available online xxxx

Keywords:
Meta-learning
Recommendation
Number of clusters
Clustering

ABSTRACT

One of the main challenges in Clustering Analysis is choosing the optimal number of clusters. A typical methodology is to evaluate a validity index over the data and to optimize it as a function of the number of clusters. However, this process can have a high computational cost. In this work, we introduce a new approach for recommending the number of clusters for a particular dataset by using Meta-learning. As the predictive performance of the meta-models induced by Meta-learning is affected by how datasets are described by meta-features, we propose a new set of meta-features able to improve the predictive performance of meta-models used for recommending the number of clusters. Experimental results show that the proposed approach provides a good recommendation of the number of clusters. Additionally, the proposed meta-feature obtains better results than meta-features for clustering tasks found in the literature.

1. Introduction

Clustering algorithms can extract novel, useful and relevant information from datasets [1]. They look for data partitions able to represent patterns in a dataset by a set of clusters [1–3]. With the growing interest in understanding, processing and summarizing data automatically, clustering algorithms have been applied to various application domains, such as anomaly detection, gene expression analysis, community detection and object segmentation [4–7].

Most clustering algorithms need a previous definition of the number of clusters in the final data partition [8,9]. Selecting a suitable number of clusters for a given clustering task is fundamental to obtain a good data partition [10,11]. Several alternatives have been proposed for estimating this number, optimizing the number of clusters by minimizing or maximizing the value returned by a validity index applied to a candidate partition [12]. However, depending on the size of the dataset and the different numbers of clusters evaluated, this process may have a high computational cost.

In this work, we introduce a new approach, based on Meta-learning (MtL), to recommend a suitable number of clusters for a clustering task. MtL investigates the use of Machine Learning (ML) algorithms for the induction of predictive models able to recommend the most suitable algorithm (and its hype-parameter values) for a new dataset [13]. Several research groups have investigated new approaches for using MtL for the recommendation of algorithms for tasks such as: classification [14,15]; clustering [16,17]; time series analysis [18,19]; optimization [20,21]; data quality [22,23]; instance selection [24,25]; recommender systems [26,27] and hyperparameter tuning [28,29].

In this paper, we investigate MtL for clustering analysis. In particular, how to use MtL to induce predictive models (meta-models) able to recommend a suitable number of clusters when applying clustering algorithms to a new dataset. These meta-models are trained using characteristics extracted from each training dataset (called meta-features). The target attribute is the best number of clusters obtained when a clustering algorithm was applied to the training dataset. The meta-features and the target attribute obtained for each training dataset creates a meta-dataset.

As MtL uses information extracted from datasets (dataset characterization) for the recommendation, how well characterization extracts relevant information is of the utmost importance for the successful use of MtL. This work also proposes a new set of meta-features able to extract more relevant information to use of MtL in the investigated task.

The remainder of this paper is organized as follows. In Section 2, we introduce the basic aspects of MtL. In Section 4, we present the main contributions from this work. In Section 5, we describe how the experiments were carried out. In Section 6,

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105682.

* Corresponding author.
E-mail addresses: bapimentel@icmc.usp.br (B.A. Pimentel), andre@icmc.usp.br (A.C.P.L.F. de Carvalho).

https://doi.org/10.1016/j.knosys.2020.105682
0950-7051/

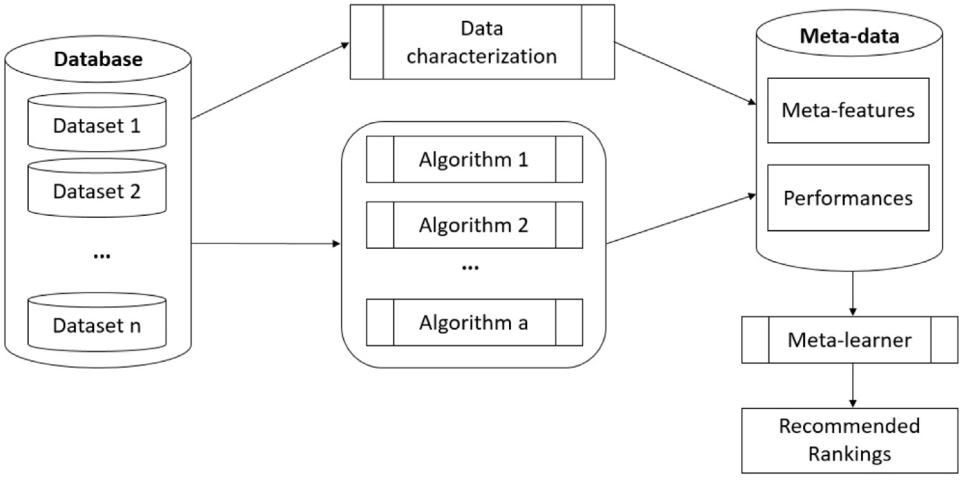

Fig. 1. Ranking recommendation using MtL.
Source: Adapted from [13].

we present and analyze the main experimental results when we compare the proposed approach with related works from the literature. We also compare the proposed meta-features with those often used in the literature on MtL for clustering. In Section 7, after our final remarks, we point out future work directions.

2. Meta-learning

This section briefly describes the main concepts of MtL and how MtL has been used in recommendation systems. According to Brazdil et al. (2008) [13], one of the aims of MtL is to assist ML practitioners and researchers by recommending the most suitable learning algorithm(s) for the problem they want to solve. Algorithm recommendation can save time in ML experiments by reducing the number of alternatives tested. It can also improve the final predictive performance by recommending the most promising algorithm(s). A recommender system based on MtL can predict the relative performance of algorithms when applied to a new dataset. For such, MtL applies an ML algorithm to a meta-dataset.

As a dataset used in ML experiments consists of examples, a meta-dataset comprises of meta-examples, where each meta-example corresponds to a dataset. The predictive attributes of meta-examples, named meta-features, are functions that extract values describing characteristics of a dataset [30]. The target attribute is related to the predictive performance of a set of ML algorithms when they are applied to a dataset. The target is usually either the label of the algorithm that obtained the best performance or the ranking of the labels of the algorithms, defined according to their performance, with the best performer on the top.

In an MtL experiment, an ML algorithm is applied to the meta-dataset to learn a meta-model able to map *meta-feature* values to the correct target attribute [31]. Meta-features are functions that extract values describing characteristics of a dataset [30].

In MtL, an ML algorithm, called *meta-learner*, is applied to the meta-dataset to induce a predictive *meta-model*. The meta-model can be applied to new datasets to predict either the most suitable ML algorithm, or the ranking of the most suitable algorithms. This task can be a simple classification task, where each class is one of the candidate ML algorithms. In this work, we use MtL to predict a ranking of labels, in particular a ranking of the best numbers of clusters. Fig. 1 illustrates how MtL can be used in a recommendation task to recommend the best number of clusters.

Clustering algorithms can be applied to unlabeled datasets. Thus, meta-features describing these datasets do not need to extract information related to labeling of objects. Next, we describe five types of meta-features that have been proposed for unlabeled datasets.

The first type of meta-feature is based on 8 statistical measures, proposed by Souto et al. (2008) [32], collected from each dataset. The second type, proposed by Souza (2010) [33] contains values from 10 clustering validity indices. The third set, proposed by Ferrari and Castro (2016) [34], is based on the distance between pairs of instances in the dataset, and contains 19 measures. The fourth type, proposed by Vukicevic et al. (2016) [35] is based on 19 clustering evaluation measures collected from the dataset. The fifth, proposed by Pimentel and Carvalho (2019) [16] is based on the distance and correlation computed from pairs of instances in the dataset, and contains 19 measures.

Next, we discuss the task addressed in this study, which is the prediction of the number of clusters for a data clustering experiment.

3. Number of clusters in a clustering partition

One of the major challenges in Clustering Analysis is the choice of the optimal number of clusters (k). A typical approach to select the number of clusters for a clustering algorithm in a given dataset is to compare the values of a validity index (extracted from the clustering structure) after applying the algorithm with different values of k. The value of k that resulted in the best validity index value is selected.

Since the choice of the optimal number of clusters depends on indices extracted from a clustering structure, and the structure of a clustering can have different interpretations, different techniques have been proposed for choosing the optimal number of clusters. The most simple and well known technique, the Elbow technique, is based on intra-cluster variation (or total within-cluster sum of square (WSS)). The WSS measures the compactness of clusters, thus the smaller the WSS value, the better. The goal of the Elbow technique is to find the number of clusters to which the addition of a new cluster does not clearly improve the WSS value. Fig. 6 illustrates an example of the variation of the WSS value according to the number of clusters (see Fig. 2).

Another approach is based on internal validity clustering indices. These indices aim to optimize the clustering quality as a function of the number of clusters. This optimization can be by maximization or minimization of the clustering quality values.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

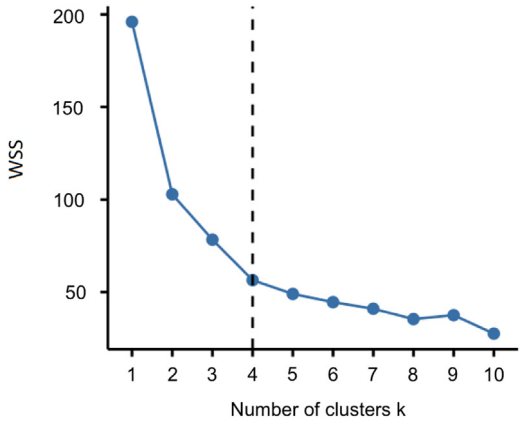

Fig. 2. The choice of the optimal number of clusters using the Elbow technique. *Source:* Adapted from [36].

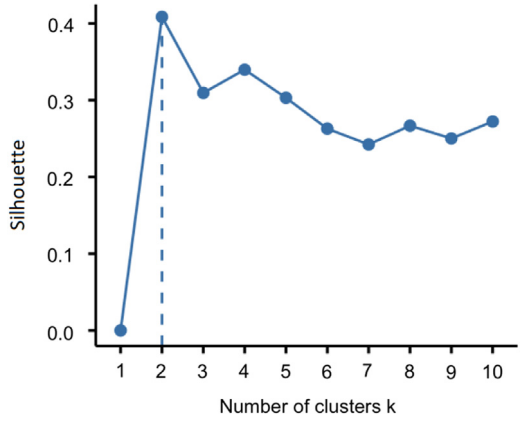

Fig. 3. The choice of the number of clusters using SI. *Source:* Adapted from [36].

Examples of internal validity indices are: Davies–Bouldim [37], Dunn [38], Krzanowski–Lai [39], Scattering-Distance [40] and Silhouette [41]. Fig. 3 illustrates an example of the variation of the Silhouette index value according to the number of clusters.

Another approach is based on information criteria [42]. The two most widely used are Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). AIC estimates the loss of information by a given model and BIC, based on the Bayesian theory framework, and tries to maximize the posterior probability of a given dataset. In the context of the recommendation of the number of clusters, AIC and BIC choose the number where the loss of information is the lowest.

4. Research contributions

This work has two main contributions to MtL research in data clustering. The first contribution is a new set of meta-features, which we believe will improve the predictive performance of a recommender system, based on MtL, to suggest the best number of clusters for a data clustering task. The second contribution is the MtL based approach for this recommender system. These two contributions are detailed in the next subsections.

4.1. A new meta-feature for MtL

Section 2 presented the six types of meta-features specifically designed to describe unlabeled datasets. However, these meta-features have been proposed for the recommendation of clustering algorithms. As in this paper we investigate the use of MtL for the recommendation of the number of clusters, we propose a new meta-feature, called Density, specifically designed to extract information relevant to estimate the best number of clusters when a clustering algorithm is applied to a given dataset. Density is based on the Parzen window method, a kernel density estimation widely used for estimating continuous density function from a dataset. This meta-feature assumes that regions with high density can indicate clusters, and this information can be useful to induce a meta-model able to recommend the number of clusters for a dataset.

To show how this meta-feature is computed, consider the following notation. Let $\Omega = \{1, \ldots, r, \ldots, n\}$ be a dataset of n elements. Let $\mathbf{x}_r = (x_{r1}, \ldots, x_{rj}, \ldots, x_{rp})$ be an element from the dataset Ω described by p variables. The probability density function P regarding the element r is given as:

$$P[r] = \frac{1}{n}\sum_{s=1}^{n}\frac{1}{h^p}\phi\left(\frac{d(\mathbf{x}_r, \mathbf{x}_s)}{h}\right) \tag{1}$$

where $h > 0$ is a smoothing parameter, called bandwidth. The function ϕ, called window function, in many applications, uses the Gaussian function:

$$\phi(u) = \frac{1}{\sqrt{2\pi}}e^{-u^2/2} \tag{2}$$

The value $d(\mathbf{x}_r, \mathbf{x}_s)$ is the Euclidean distance between elements r and s, given as follows:

$$d(\mathbf{x}_r, \mathbf{x}_s) = \sqrt{\sum_{j=1}^{p}(x_{rj} - x_{sj})^2} \tag{3}$$

The values of probability density function are stored in the vector $\mathbf{P} = (P[1], \ldots, P[r], \ldots, P[n])$. After computing all values of this function, vector $\mathbf{P}$ is normalized in the interval [0, 1] using:

$$P'[r] = \frac{P[r] - min(\mathbf{P})}{max(\mathbf{P}) - min(\mathbf{P})} \tag{4}$$

To compute the meta-features, vector $\mathbf{P}'$ is represented by a histogram. The previously described meta-feature can be extracted for each bin in the histogram, resulting in a number of meta-feature values equal to the number of bins in the histogram. In this study, the number of bins (n_{bins}) is calculated as follows:

$$n_{bins} = \left\lceil \sqrt{\overline{n}} \right\rceil \tag{5}$$

where $\lceil . \rceil$ is the ceiling function and $\overline{n}$ is the mean of dataset sizes (n_b) regarding the N datasets used to induce meta-models. For the experiments carried out in this study, Density extracts 70 measures.

$$\overline{n} = \frac{1}{N}\sum_{b=1}^{N} n_b \tag{6}$$

Therefore, the same meta-feature function measures the frequency of values of $\mathbf{P}'$ that fall into a given bin of the histogram. According to Kalousis [43], histograms may provide more information about the data being characterized than a single value. In this work, the data being characterized is the vector of values of probability density function. Algorithm 1 shows how these meta-features are extracted.

4.2. Using MtL to recommend the number of clusters

As we are using MtL to predict the number of clusters for a new dataset, we use regression algorithms to predict these

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

Algorithm 1 Meta-feature extraction

Require: Dataset Ω with n instances; bandwidth h

Let **P** and **P'** be vectors without initial values and Q be a matrix
for $1 \leq r \leq n$ **do**
  for $r \leq s \leq n$ **do**
    Compute $Q[r,s] = \phi\left(\frac{d(\mathbf{x}_r,\mathbf{x}_s)}{h}\right)$ (Eq. (2))
  end for
end for
for $1 \leq r \leq n$ **do**
  Compute $P[r]$ using matrix Q (Eq. (1))
end for
for $1 \leq r \leq n$ **do**
  $$\mathbf{P'}[r] = \frac{\mathbf{P}[r] - min(\mathbf{P})}{max(\mathbf{P}) - min(\mathbf{P})}$$
end for
Compute meta-features using a histogram (Eq. (5))

return Meta-features.

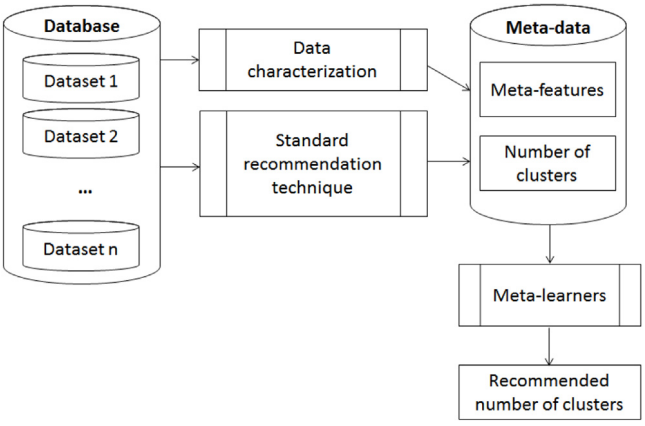

Fig. 4. Number of clusters recommendation using MtL.

numbers. We are using N regression algorithms to predict N numbers of clusters. Each number receives a score, given by how good the clustering partition obtained is regarding a clustering validity index.

We use the scores obtained by each regression algorithm for each dataset to create a ranking to be used as a meta-target. In this ranking, the best number of clusters is placed at the top and the worst at the bottom. Thus, each training meta-example has as a target attribute the ranking of the best options, numbers of clusters, for the associated dataset. When we repeat this process for all D datasets, we have a meta-dataset with D meta-examples. We can then apply an ML algorithm to induce a meta-model able to predict the best ranking of numbers of clusters for a new dataset. Figs. 4 adapts 1 to illustrate this process.

In this paper, as we also investigate the influence of the type of meta-feature and the validity index used in the quality of the meta-model, we create a meta-dataset for each meta-feature set and for each clustering validity index.

When applied to a new dataset, each meta-model predicts, for a particular clustering algorithm, the best number of clusters for the dataset. Thus, if we use N meta-models, a list of N numbers of clusters is produced. As we want to recommend only 1 number of clusters, we need to select one of the N numbers.

If we repeat the MtL process for M validation indices, we end up with NxM numbers of clusters. As we want to recommend 1 number of clusters, we look for the number of clusters most frequently predicted in the list of NxM numbers.

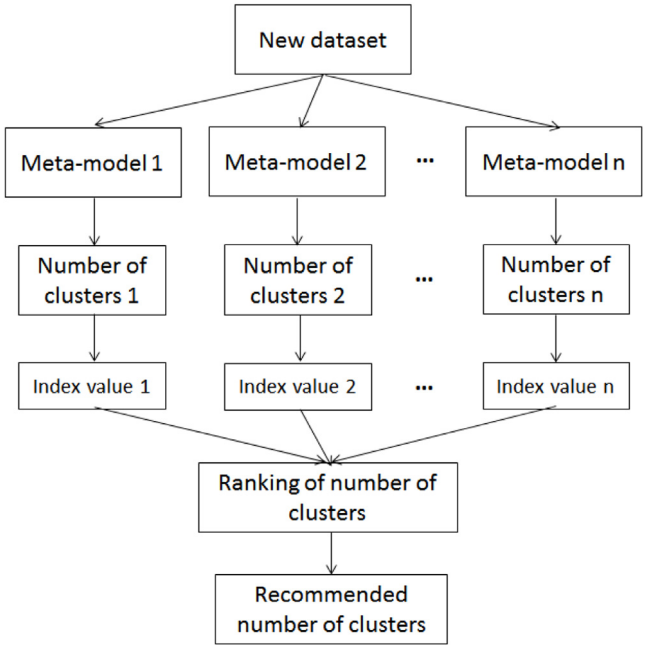

Fig. 5. Predictions of meta-models for a new dataset and the final recommended number of clusters.

After applying different meta-models, a recommendation list of the best numbers of clusters is produced. As the goal of the recommender system using MtL is to recommend only one value of the number of clusters, this list of the best numbers of clusters is summarized. For this, a validity index, using a number of clusters as the parameter, is applied to the new dataset, in such a way that each meta-model is related to a validity index value. Thus, the list is sorted according to the validity index values, resulting in a ranking of number of clusters. Then, the final recommended number of clusters is that value whose rank is the lowest. Fig. 5 details the process of recommending a number of clusters from the list of numbers of clusters.

5. Experimental methodology

This section describes in detail the methodology used in the experiments, in order to evaluate the proposed approach and to compare our proposal with the traditional approach found in the literature. The source code used in the experiments is available at https://github.com/bapimentel/NumberGroups.

5.1. Datasets

The experiments carried out used 219 datasets collected from Open Machine Learning (OpenML), an online ML platform that allows access to a large number of datasets and sharing of ML experiments. These datasets cover different domains, such as engineering, biology, medicine, physics, robotics, and can be downloaded from https://www.openml.org/s/88/data. For each dataset, the predictive attributes were normalized to the interval [0, 1]. Table 1 describes all datasets according to number of classes, attributes and instances using the minimal, maximal, average, mode, standard deviation, skewness and kurtosis of values.

5.2. Meta-features

In the experiments, we used six types of meta-features, the five presented in Section 2 and the Density type meta-feature proposed in Section 4.1. Table 2 summarizes these six types of meta-features used in the experiments.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

Table 1
Description of all datasets according to the number of classes, predictive attributes and instances.

	Classes	Attributes	Instances
Min	2	2	100
Max	250	168	10^6
Mean	5.8499	19.9636	4799.8545
Mode	2	5	100
Std	22.1540	23.9922	67 249.5303
Skewness	6.3278	2.5983	14.7309
Kurtosis	42.0712	8.8355	215.0027

Table 2
Name and length of meta-feature sets used in this study.

Name	Length
Statistical (ST)	8
Clustering (CL)	10
Distance (DI)	19
Evaluation (EV)	19
Correlation (CO)	19
Density (DS)	70

Table 3
Name and abbreviation of the indices used in this study.

Name	Abbreviation
Elbow	EL
Akaike information criterion	AIC
Bayesian information criterion	BIC
Davies–Bouldin	DB
Dunn	DU
Krzanowski–Lai	KL
Scattering-Distance	SD
Silhouette	SI

Table 4
Name and abbreviation of the regression algorithms used in this study.

Name	Abbreviation
K-Nearest Neighbors [44]	K-NN
Linear Support Vector Machine [45]	SVM
Support Vector Machine with Radial Basis Function [45]	SVM-RBF
Decision Tree [46]	DT
Random Forest [47]	RF
AdaBoost [48]	Ada
Perceptron [49]	PC
Lasso [50]	LS
Logistic Regression [51]	LgR
Stochastic Gradient Descent [52]	SGD

5.3. Indices for recommending the number of clusters

In this work, we use a meta-dataset whose predictive variables are meta-features (previously described) extracted from the 219 datasets and the target variable is a validity index extracted from a partition produced by a clustering algorithm.

For such, 8 interval validity indices widely used in the literature to assess partitions with different numbers of clusters, were selected. The name, and the acronym, for each of these 8 indices are listed in Table 3.

5.4. Meta-learners

To reduce the influence of learning bias, in this work we used 10 different regression algorithms to induce meta-models using the meta-datasets. The meta-models were later used to predict the number of clusters in a new dataset. These algorithms are listed, together with their acronyms, in Table 4:

5.5. Error measures

To assess the predictive performance of the induced models, 3 error measures were selected, given their frequent use in the regression literature. To describe these three measures, consider a_i to be the actual value (the number of clusters chosen after using an index in the dataset i), and let p_i be the predicted value (the number of clusters obtained after applying a regression model to the dataset i).

The first measure, Mean Absolute Percentage Error (MAPE), has the positive aspect of interpretability, since values are given in percentage. MAPE can be calculated as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|a_i - p_i|}{a_i} \tag{7}$$

The second measure, Mean Relative Absolute Error (MRAE), compares the model error with the error of a baseline. Thus, this measure can give a more accurate performance value. MRAE can be computed using:

$$MRAE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{a_i - p_i}{a_i - \overline{a}}\right| \tag{8}$$

The third measure, Relative Root Mean Square Error (RRMSE), unlike the previous measures, which compare the error dataset by dataset, compares the global errors (of the model and the baseline) taking into account all datasets. Its value is given by:

$$RRMSE = \sqrt{\frac{\sum_{i=1}^{n}(a_i - p_i)^2}{\sum_{i=1}^{n}(a_i - \overline{a})^2}} \tag{9}$$

For the 3 measures, the smaller the value, the better the model prediction.

5.6. Evaluation of meta-learners

The recommendation process was evaluated using 10-fold cross-validation. For each test fold, the meta-examples (one for each original dataset) in the fold are selected to be the test set and the meta-examples in the remaining 9 folds are used as a training set. Next, we calculated the mean error measures (described next) obtained for the 10 runs. This process was repeated 30 times, with the 10 folds randomly defined at each repetition. Afterwards, the mean and standard deviation of the 30 mean error values are computed.

5.7. Methodology

The steps followed to validate the proposed approach were:

1. Choose a set of datasets and normalize the predictive variables of each dataset in the interval [0, 1] (see Section 5.1);
2. Extract the meta-features (see Section 5.2) from each dataset;
3. Use the number of clusters chosen by each validity index (see Section 5.3) as the target variables for each dataset;
4. Construct the meta-dataset using the meta-features and the target variable;
5. Use regression models (see Section 5.4) to predict the number of clusters based on a training meta-dataset;
6. Compute error measures (see Section 5.5) to compare the number of clusters chosen by an index and the number of clusters predicted by a regression model;
7. Apply the clustering validity approach (see Section 5.6) to obtain the final results.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

6. Experimental results

In this section, we present the main results obtained in experiments to predict the best number of clusters for new datasets. As the best number of clusters can vary for each cluster validity index, we ran experiments to predict the number of classes for eight different validity indices. For all of them, the number of clusters, k, varied from 2 to 10. For each value, we ran the K-Means clustering algorithm to converge to a clustering partition. We used the value of k that produced the best partition, according to the validity index considered, as a target variable in the meta-dataset. As mentioned in Section 5.3, in this study we used eight validity indices, which can be divided into three cluster validity approaches:

- EL index;
- Indices based on Information criteria: AIC and BIC;
- Indices based on Internal validation: DB, DU, KL, SD and SI.

After applying the experimental methodology described in the previous section, we performed five evaluations:

- Prediction of the best number of clusters by MtL for the three validity index approaches (Section 6.1);
- Quality of the partitions obtained with the number of clusters predicted by MtL for the eight validity indices (Section 6.2);
- Ranking of the meta-models with the best predictive performance, for each validity index (Section 6.3);
- Prediction of the number of clusters by MtL using different numbers of meta-models (Section 6.4);
- Computational cost of a recommendation when using MtL and the traditional approach, for each validity index (Section 6.5).

In the experiments reported in Sections 6.1 to 6.3, we applied the 10 ML algorithms mentioned in Section 5.4, to 8x6, 48, meta-datasets. Each meta-dataset has the meta-features extracted from one of the 6 meta-feature sets as predictive attributes and the best number of clusters defined by one of the 8 validity indices as meta-target.

When comparing the validity indices, for each index, we compare the predictive performance obtained by the meta-models using the proposed meta-feature, called Density (DS), with five sets of meta-features from the literature, described in Section 5.2: Statistical (ST); Clustering (CL); Distance (DI); Evaluation (EV) and Correlation (CO). We also included two baselines in the comparison: average and majority. The best results regarding a given error measure are in bold. To assess the statistical significance of the results, we ran hypothesis tests (Appendix B, Appendix C and Appendix D). The following sections describe the predictive results for three error measures (MAPE, MRAE and RRMSE), with the mean and standard deviation (in parenthesis) and for the eight validity indices (divided into three groups).

6.1. Comparison by index validity measures (number of clusters)

Next, Sections 6.1.1–6.1.3 show the predictive performance of the recommendation provided by 8 methods for the Elbow technique, Information Criteria and Internal Validation indices approaches, respectively.

6.1.1. Elbow technique

Table A.10 and Fig. 6 show the results for the MtL and the baselines regarding the EL index.

According to these results, the average and majority baselines presented the best predictive performance for the MAPE, MRAE and RRMSE measures. The best result for MtL was obtained when using the CL meta-feature. The prediction of the number of clusters by MtL with the CL meta-feature was as good as the number suggested by the EL index, but with a lower cost, since it is only necessary to apply meta-models to these meta-features.

6.1.2. Information criteria

Two information criteria indices are used, BIC and AIC. Table A.11 and Fig. 7 show the results for the MtL and the baselines for the BIC index and Table A.12 and Fig. 8 show the results of MtL and baselines for the AIC index.

In the results for the BIC index, the predictive performance using MtL is better than using the baselines, except for the DS meta-feature with the MRAE measure. For the AIC index, the results obtained by using MtL were better than the baselines, except for MtL, using the ST and CL meta-features, for the MRAE and RRMSE measures. For them, the number of clusters recommended was as good as the average of the number of clusters for all datasets.

6.1.3. Internal valuation

Tables A.13 to A.17 and Figs. 9 to 13 show the results of MtL and baselines representing, respectively, DB, DU, KL, SD and SI indices.

For these indices, it is important to highlight that, in all 15 cases, MtL obtained better results than the baselines for all error measures. Moreover, 14 of the 15 best results were obtained when the recommender system used the proposed meta-feature, DS, for all internal valuation indices and error measures analyzed. The exception was the KL index when the average baseline method presented a lower MRAE value. Besides, for all internal valuation indices, the average baseline method performed better than the majority baseline for the MRAE and RRMSE error measures. The opposite occurred for the MAPE error measure. The following section summarizes the clustering quality results obtained by the meta-features evaluated in this work.

6.2. Comparison by index approaches (clustering quality)

This work also analyzed the quality of the clustering after recommending a number of clusters. The clustering quality was calculated using a validity index. Sections 6.2.1–6.2.3 show, respectively, the results obtained for the EL index, the two Information Criteria indices (AIC and BIC) and the five Internal Validation indices (DB, DU, KL, SD and SI).

6.2.1. Results for the EL index

The quality of the partitions obtained when using MtL and the baselines for the EL index and clustering quality can be seen in Table A.18 and Fig. 14.

According to these results, except for the MAPE measure, the quality of the clustering partitions obtained using the number of clusters recommended by MtL was better than those obtained using the number of clusters recommended by the baselines. The MRAE results suggest that the clustering quality obtained by MtL is more similar to the target clustering quality (using the EL index) than the clustering quality obtained by the baselines. This can be explained by the wide diversity of the application domain of the datasets.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

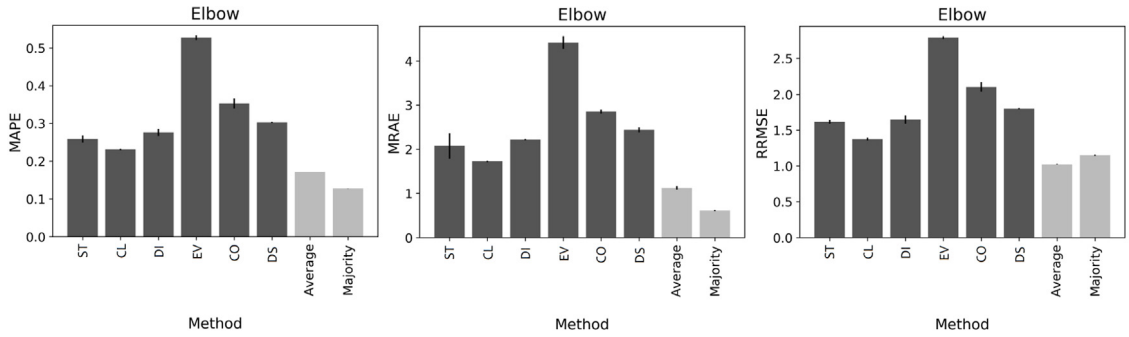

Fig. 6. Predictive performance of the methods regarding MAPE, MRAE and RRMSE measures for the EL index.

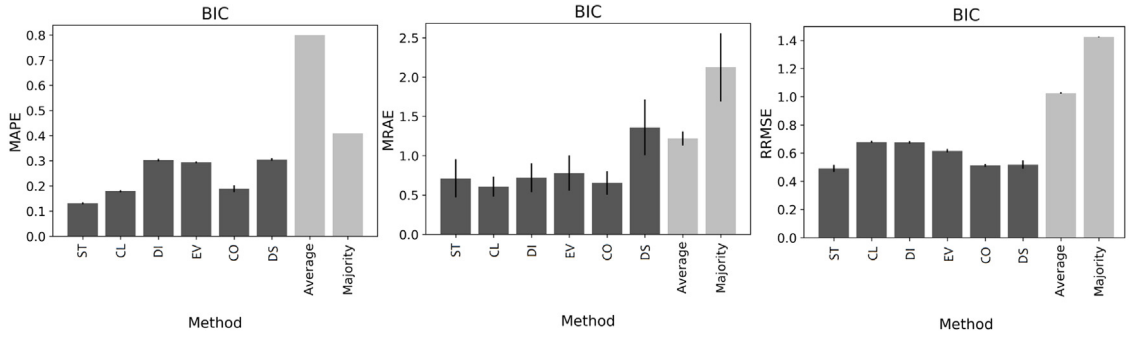

Fig. 7. Predictive performance of the methods regarding MAPE, MRAE and RRMSE measures for the BIC index.

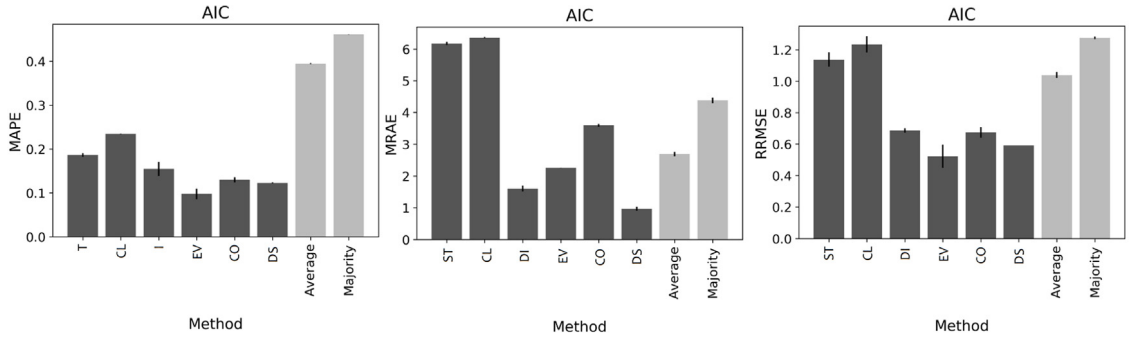

Fig. 8. MAPE, MRAE and RRMSE performance of the methods for the AIC index.

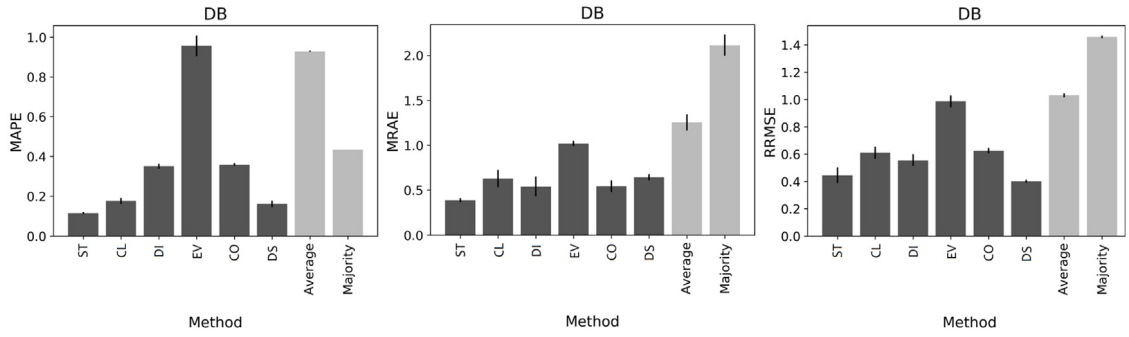

Fig. 9. MAPE, MRAE and RRMSE performance of the methods for the DB index.

6.2.2. Results for the information criteria indices

Table A.19 and Fig. 15 show the results for the MtL and the baselines regarding the BIC index and Table A.20 and Fig. 16 show the results for the MtL and the baselines regarding the AIC index.

Concerning the BIC index, it is possible to highlight that the clustering quality after applying MtL approach is better than the clustering quality after applying baselines, for MRAE and RRMSE measures. Whereas, for the AIC index, this conclusion can be observed for all error measures analyzed in this work. Besides, except for MAPE when the BIC index is used, recommending the number of clusters using the average baseline produced a cluster with a higher quality than using the majority baseline.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

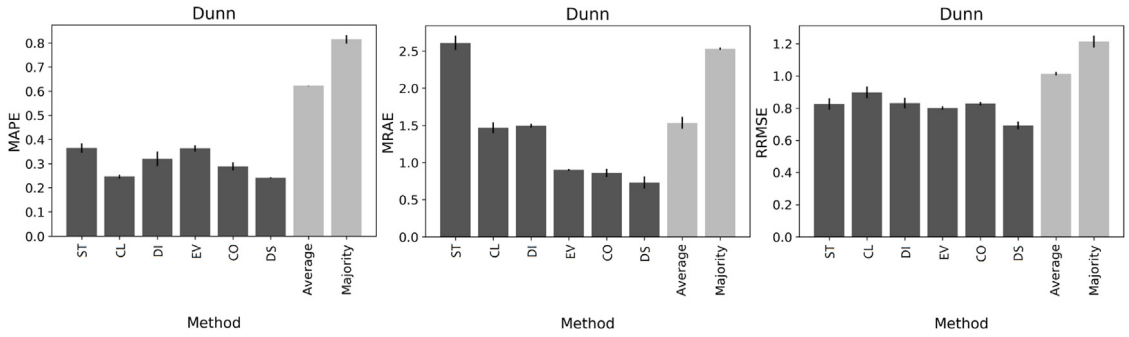

Fig. 10. MAPE, MRAE and RRMSE performance of the methods for the DU index.

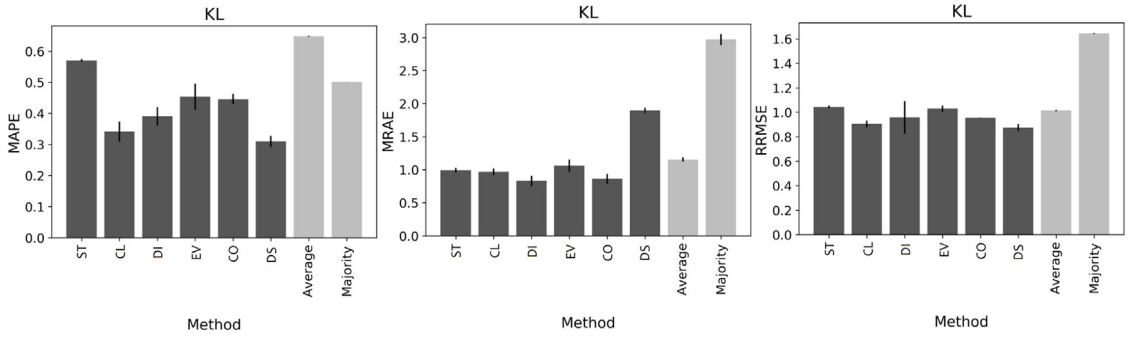

Fig. 11. MAPE, MRAE and RRMSE performance of the methods for the KL index.

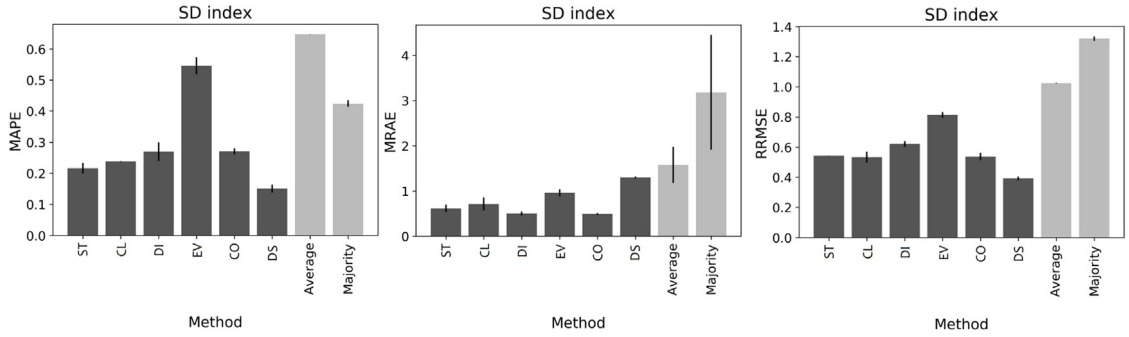

Fig. 12. MAPE, MRAE and RRMSE performance of the methods for the SD index.

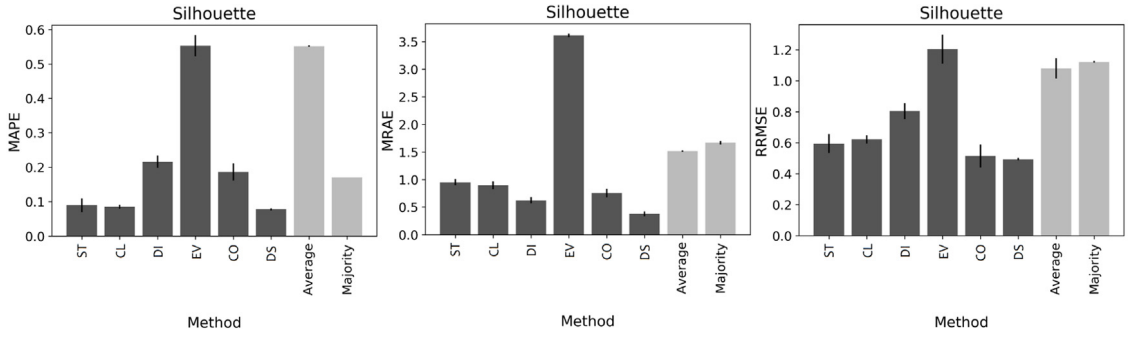

Fig. 13. MAPE, MRAE and RRMSE performance of the methods for the SI index.

6.2.3. Results for the internal validation indices

Tables A.21 to A.25 and Figs. 17 to 21 show the results for the MtL and the baselines, representing, respectively, the DB, DU, KL, SD and SI indices, regarding the clustering quality after using the number of clusters recommended.

The results pointed out that, for all indices and meta-features, MtL was able to obtain a better clustering quality when compared with the baselines. Moreover, the proposed meta-feature, DS, was the best option among the investigated meta-features for all indices, except the KL index, regarding MRAE, and the SI index, concerning MAPE and MRAE.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

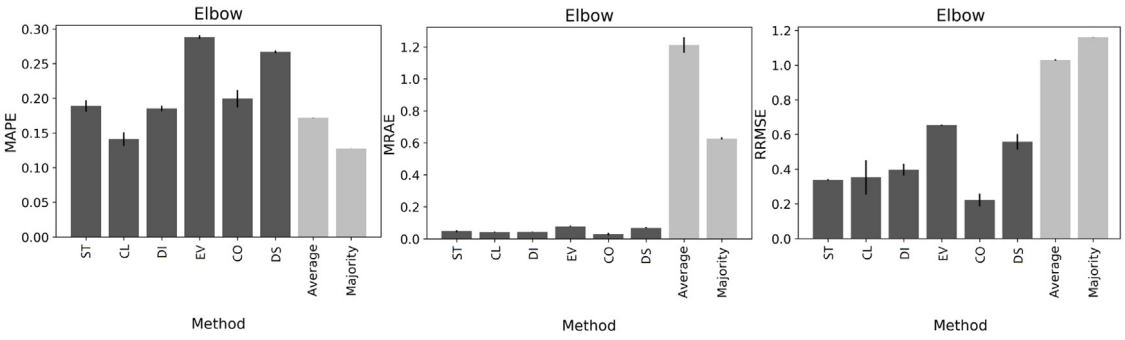

Fig. 14. MAPE, MRAE and RRMSE performance of the methods for the EL index concerning clustering quality.

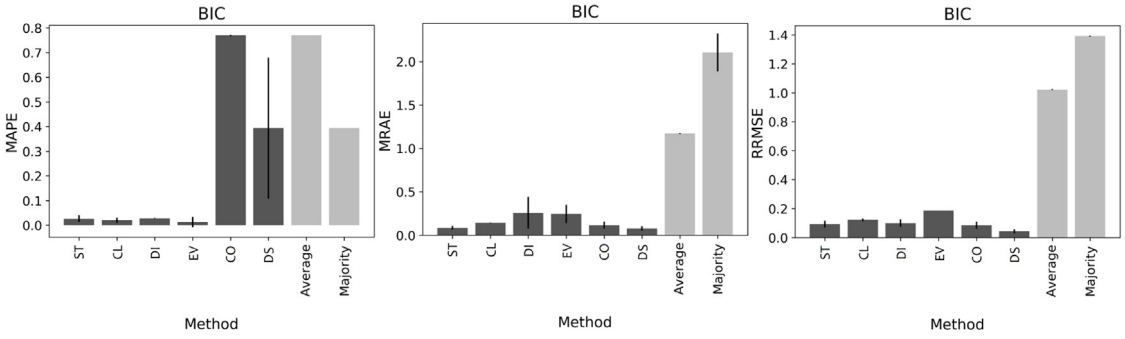

Fig. 15. MAPE, MRAE and RRMSE performance of the methods for the BIC index concerning clustering quality.

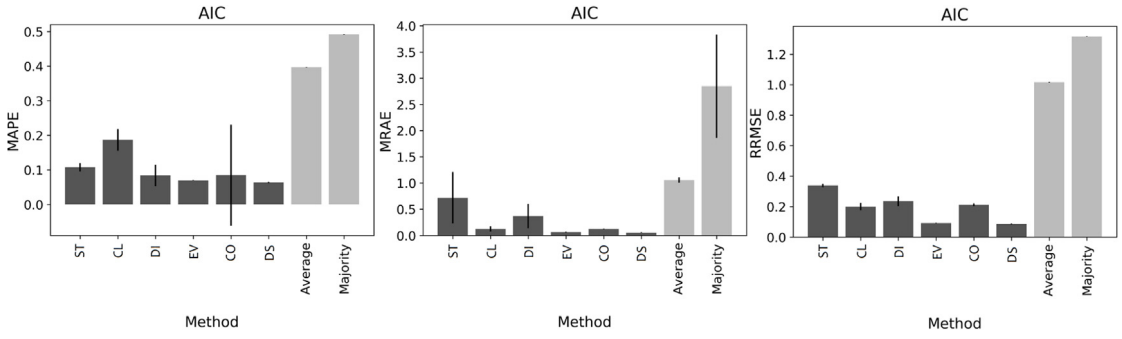

Fig. 16. MAPE, MRAE and RRMSE performance of the methods for the AIC index concerning clustering quality.

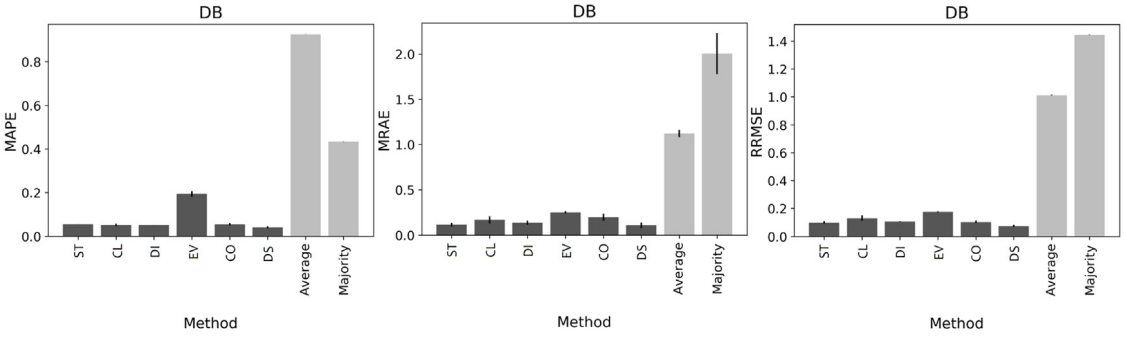

Fig. 17. MAPE, MRAE and RRMSE performance of the methods for the DB index concerning clustering quality.

6.3. Comparison of the meta-features using rankings

Based on Section 6.1, we assessed, for each index, the ranking of the set of meta-features regarding each error measure. For this analysis, we created a ranking for the mean predictive performance obtained using the different sets of meta-features. In this ranking, the lower the mean ranking position of a meta-feature set, the better the predictive performance obtained is using this set. Tables 5–7 show the rankings for all indices and error measures, respectively.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

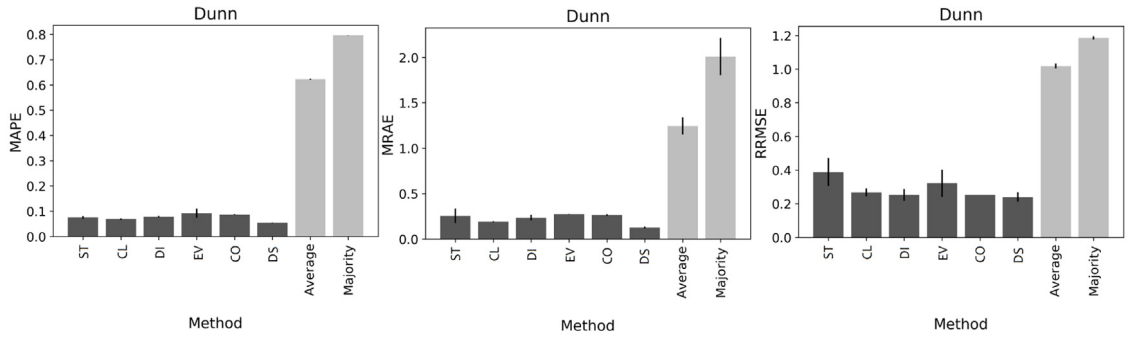

Fig. 18. MAPE, MRAE and RRMSE performance of the methods for the DU index concerning clustering quality.

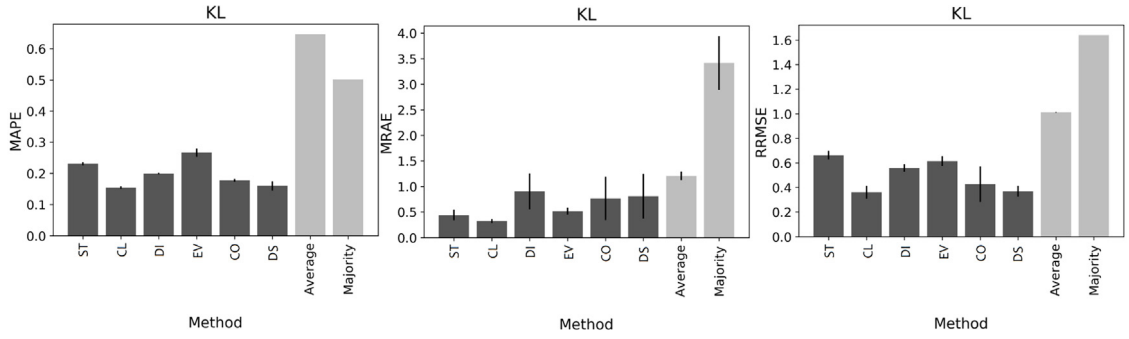

Fig. 19. MAPE, MRAE and RRMSE performance of the methods for the KL index concerning clustering quality.

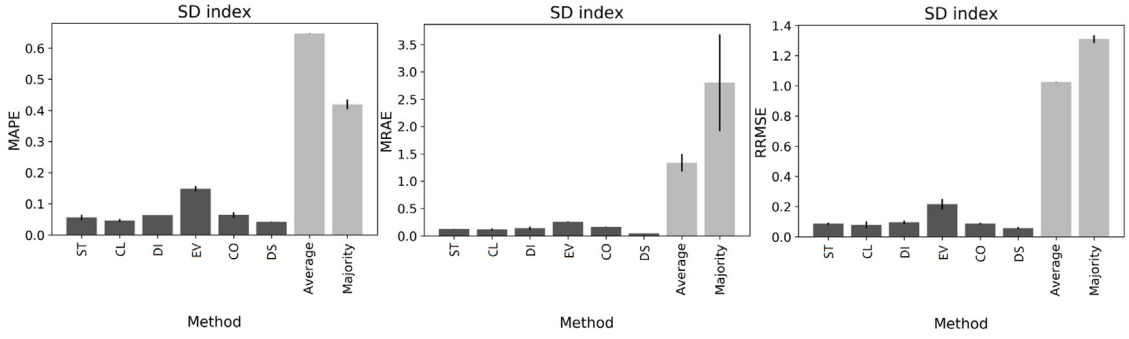

Fig. 20. MAPE, MRAE and RRMSE performance of the methods for the SD index concerning clustering quality.

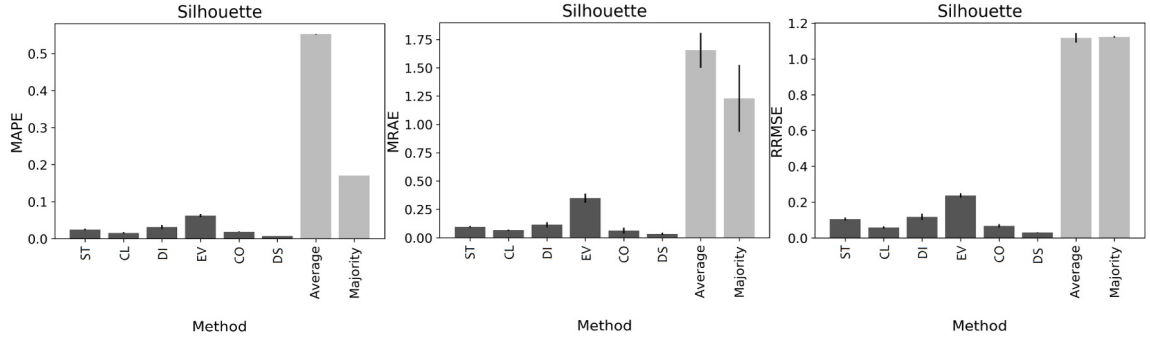

Fig. 21. MAPE, MRAE and RRMSE performance of the methods for the SI index concerning clustering quality.

According to Table 5, on average, the best performance was obtained by the recommender system using the proposed meta-feature DS, with MAPE as an error measure. For the MRAE error measure (Table 6), the recommender system using the DI meta-feature was the best option. For the RRMSE error measure (Table 7), the best choice is the recommender system using the proposed meta-feature, DS. For all tables, the results confirmed that the baselines obtained the worst predictive performances.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.

Table 5
Ranking according to MAPE.

Index	MtL						Baseline	
	ST	CL	DI	EV	CO	DS	Average	Majority
EL	4	3	5	8	7	6	2	1
BIC	1	3	6	5	4	2	8	7
AIC	5	6	4	1	3	2	8	7
BD	1	3	4	6	5	2	7	8
DU	6	2	4	5	3	1	7	8
KL	6	2	3	5	4	1	8	7
SD	2	3	4	7	5	1	8	6
SI	3	2	6	8	5	1	7	4
Mean	3.5000	3.0000	4.5000	5.6250	4.5000	**2.0000**	6.8750	6.0000
STD	2.0702	1.3093	1.0690	2.2638	1.3093	1.6903	2.0310	2.3905

Table 6
Ranking according to MRAE.

Index	MtL						Baseline	
	ST	CL	DI	EV	CO	DS	Average	Majority
EL	4	3	5	8	7	6	2	1
BIC	3	1	4	5	2	7	6	8
AIC	7	8	2	3	5	1	4	6
BD	1	4	2	6	3	5	7	8
DU	8	4	5	3	2	1	6	7
KL	4	3	1	5	2	7	6	8
SD	3	4	2	5	1	7	6	8
SI	5	4	2	8	4	1	6	7
Mean	4.3750	3.8750	**2.8750**	5.3750	3.2500	4.3750	5.3750	6.6250
STD	2.2638	1.9594	1.5526	1.9226	1.9821	2.8754	1.5980	2.3867

Table 7
Ranking according to RRMSE.

Index	MtL						Baseline	
	ST	CL	DI	EV	CO	DS	Average	Majority
EL	4	3	5	8	7	6	1	2
BIC	1	5	4	3	2	6	7	8
AIC	5	6	4	1	3	2	7	8
BD	2	4	3	6	5	1	7	8
DU	3	6	5	2	4	1	7	8
KL	7	2	4	6	3	1	5	8
SD	4	2	5	6	3	1	7	8
SI	3	4	5	6	2	1	7	8
Mean	3.6250	4.0000	4.3750	4.7500	3.6250	**2.3750**	6.0000	7.2500
STD	1.8468	1.6036	0.7440	2.4349	1.6850	2.2638	2.1381	2.1213

It is also important to highlight that the best rank positions regarding the proposed meta-feature DS were obtained for Internal Validation indices.

6.4. Increasing the number of ML algorithms in the recommender system

In the MtL experiments presented so far, we summarized the results obtained by the combination of the 10 ML algorithms. We believe that if we combine a subset of these algorithms in an ensemble we can improve the predictive performance obtained by the recommender system. On the other hand, this section aims to analyze the influence of the size of this subset of algorithms for the recommender system performance.

For such, we first compare the predictive performance obtained by using each ML algorithm individually. For such, for each set of meta-features, we assign a ranking position to each ML algorithm, in a similar way to what we did for the sets of meta-features and baselines in Section 6.3, when we compared sets of meta-features and the baselines.

The ML algorithms used in the experiments were ranked according to the predictive performance they obtained for each validity index in the prediction of the best number of clusters

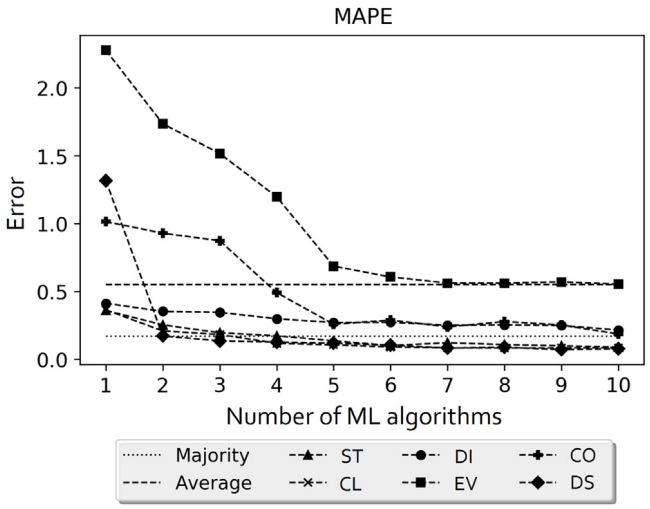

Fig. 22. Comparison of recommender system performance for each meta-feature according to the subset size variation: MAPE error measure.

in a clustering partition. Table 8 shows the mean and standard deviation (between parenthesis) of the ranking position of each algorithm for each set of meta-features. In the last row, we show, for each algorithm, the mean and the standard deviation of their mean rankings for the 8 sets of meta-features.

The mean and standard deviation were calculated as follows. For each one of the 8 sets of meta-features, the 10 ML algorithms are run 300 times. For each run, we rank the 10 algorithms according to their predictive performance in that run. Afterwards, we calculated, for each algorithm, the mean and standard deviation of its 300 ranking positions.

Appendix D details the rankings according to each index and meta-feature. Table 8 shows the mean and standard deviation (between parenthesis) of the rankings concerning all meta-features. It can be seen in Table 8 that K-NN was the best algorithm to predict the number of clusters, and SGD presented the lowest predictive performance.

In the second step, we use the mean ranking of each algorithm in the last row of Table 8 to progressively increase the number of meta-learners used in an ensemble. We start with only one ML, the algorithm with the best ranking position, which is K-NN. We keeping adding the next best performer ML algorithm, until we have in our recommender system one ensemble with all 10 ML algorithms.

For the sake of simplicity, only one index was chosen in this analysis: SI. We chose SI because it is one of most popular indices for recommending the number of clusters [41,53].

Tables A.26, A.27 A.28 and Figs. 22–24 show how the predictive performance of the recommender system is affected as we increase the number of ML algorithms, for the MAPE, MRAE and RRMSE error measures, respectively.

The results for all three error measures show that as we add more algorithms, we usually improve the predictive performance, until a limit were there is no further improvement. This occurs for all sets of meta-features. The best predictive performances for MAPE, MRAE and RRMSE were obtained using the DS meta-feature. In these cases, the prediction errors were very low, 0.0706, 0.3015 and 0.4592. For the average and majority baselines, the errors were clearly higher, 0.5527 and 0.1705 (MAPE), 1.6546 and 1.2290 (MRAE) and 1.1184 and 1.1234 (RRMSE).

According to Table A.26 and Fig. 22, the error of recommender system using the proposed meta-feature DS and 1 model is higher than the error after using baselines (average and majority). On

Table 8
Mean and standard deviation of rankings for each index.

Index	Meta-models									
	K-NN	SVM	SVM-RBF	DT	RF	Ada	PC	LS	LgR	SGD
EL	3.1667 (3.2301)	5.0000 (2.7080)	3.6111 (1.2896)	4.7222 (0.9756)	4.5556 (0.5019)	5.3333 (2.6750)	5.0556 (1.8308)	6.5556 (1.4089)	8.3334 (1.3166)	8.6667 (1.8738)
BIC	1.0000 (0.0000)	3.1111 (1.3608)	4.1667 (1.0274)	5.3334 (1.2293)	4.2222 (0.2722)	5.3333 (0.8944)	6.8889 (0.5018)	6.7778 (0.9813)	8.6667 (1.1926)	9.5000 (0.6236)
AIC	2.5000 (3.6742)	4.6667 (3.3665)	3.1111 (1.4707)	4.7778 (1.6009)	5.5000 (1.2427)	5.7222 (1.4969)	6.2222 (0.6885)	6.9445 (1.6920)	7.3333 (1.8257)	8.2222 (2, 1671)
DB	1.0000 (0.0000)	3.2222 (1.8459)	4.4444 (1.8579)	4.8889 (1.0681)	5.5556 (0.6206)	6.0556 (1.2003)	5.8889 (1.8217)	7.1111 (1.8935)	7.5556 (1.2049)	9.2778 (0.9289)
DU	1.0000 (0.0000)	3.4445 (1.8698)	4.0556 (0.9290)	4.7222 (1.3067)	4.7222 (0.7429)	6.0000 (0.8165)	6.9445 (0.9526)	7.0000 (0.9888)	8.3333 (1.3166)	8.7778 (1.7596)
KL	1.0000 (0.0000)	2.8889 (1.3931)	4.6111 (0.8798)	4.9445 (0.7123)	4.3334 (0.9189)	5.6667 (0.5578)	6.4445 (1.3608)	7.0000 (0.9661)	8.5000 (1.0904)	9.6111 (0.3277)
SD	1.0000 (0.0000)	3.5000 (3.2094)	4.1667 (2.2681)	5.5556 (2.2278)	5.4445 (1.3278)	6.1111 (0.8861)	6.1667 (1.4870)	7.0555 (1.4820)	7.6667 (1.6330)	8.3333 (2.2706)
SI	1.0000 (0.0000)	3.5000 (3.2094)	4.1667 (2.2681)	5.5556 (2.2278)	5.4445 (1.3278)	6.1111 (0.8861)	6.1667 (1.4870)	7.0555 (1.4820)	7.6667 (1.6330)	8.3333 (2.2706)
Mean	1.4583 (0.8672) 1	3.6667 (0.7554) 2	4.0417 (0.4758) 3	5.0625 (0.3618) 5	4.9723 (0.5697) 4	5.7917 (0.3290) 6	6.2223 (0.5969) 7	6.9375 (0.1841) 8	8.0070 (0.5043) 9	8.8403 (0.5543) 10

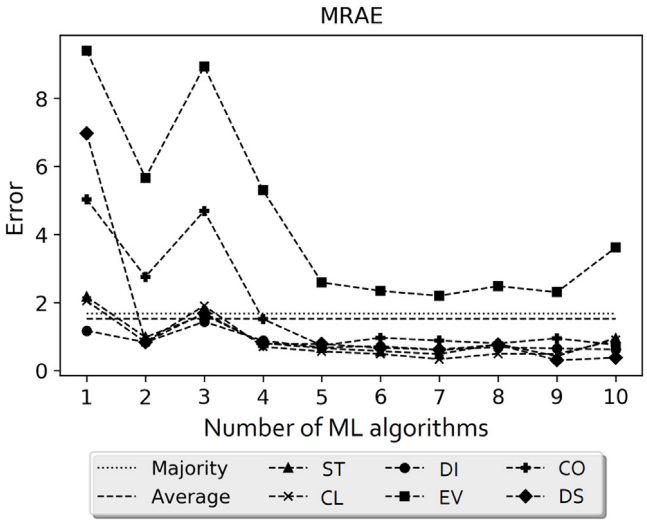

Fig. 23. Comparison of recommender system performance for each meta-feature according to the subset size variation: MRAE error measure.

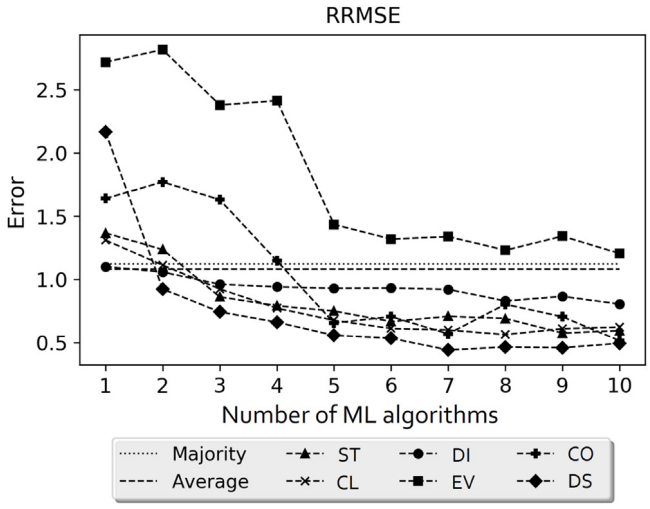

Fig. 24. Comparison of recommender system performance for each meta-feature according to the subset size variation: RRMSE error measure.

the other hand, the error decreases and the recommender system obtains better results when the size of subsets is larger than 2.

The results presented in Table A.27 and Fig. 23 which deal with the MRAE error measure show that the recommendation obtained using most meta-features is better than the recommendation of baseline methods, except for Distance meta-feature. It is also important to highlight that the error values stabilize from subset sizes equal to 5.

The results presented in Table A.28 and Fig. 24, for the RRMSE error measure, show that all sets of meta-features induce to an inferior recommendation performance when compared with the baselines for a small number of models. On the other hand, the performance is better when the number of models is larger than 4, except for the EV meta-feature, whose performance was worse than baseline methods for all sizes of subsets.

6.5. Comparison of computational cost

In this section, we compare the processing time of the proposed approach against the traditional approach used in the literature for the recommendation of the number of clusters (Elbow technique or minimum/maximum of indices values — Section 3). Concerning the recommender system using meta-learning, for each meta-feature, and for each index, the mean and the standard deviation of the processing time of the meta-models induced using this meta-feature were computed. Table 9 shows these values. The last column of this table shows these figures for the baseline.

From these results, it is important to highlight that, for all meta-features and indices analyzed in this work, the recommender system using meta-learning obtained better results, concerning the processing time, than the traditional approach used in the literature. Moreover, as excepted, the recommender system using ST meta-feature is the fastest alternative, since its length is 8, as presented in Table 2. On the other hand, although the

Table 9
Mean and standard deviation of processing time (ms) for each meta-feature and index.

Index	MtL						Traditional
	ST	CL	DI	EV	CO	DS	
EL	0.0126 (0.0002)	0.0179 (0.0013)	0.0231 (0.0002)	0.0190 (0.0012)	0.0222 (0.0008)	0.0199 (0.0002)	0.3942 (0.0000)
BIC	0.0109 (0.0005)	0.0118 (0.0001)	0.0193 (0.0006)	0.0183 (0.0005)	0.0189 (0.0001)	0.0315 (0.0002)	0.2667 (0.0000)
AIC	0.0109 (0.0009)	0.0126 (0.0004)	0.0212 (0.0013)	0.0180 (0.0002)	0.0238 (0.0005)	0.0357 (0.0004)	0.3299 (0.0000)
DB	0.0106 (0.0002)	0.0140 (0.0005)	0.0187 (0.0005)	0.0154 (0.0008)	0.0200 (0.0006)	0.0301 (0.0002)	0.0871 (0.0000)
DU	0.0102 (0.0002)	0.0136 (0.0002)	0.0186 (0.0004)	0.0181 (0.0003)	0.0209 (0.0002)	0.0160 (0.0002)	0.0903 (0.0000)
KL	0.0089 (0.0004)	0.0105 (0.0004)	0.0202 (0.0008)	0.0155 (0.0006)	0.0178 (0.0023)	0.0197 (0.0007)	0.1505 (0.0000)
SD	0.0107 (0.0008)	0.0201 (0.0008)	0.0247 (0.0009)	0.0231 (0.0010)	0.0251 (0.0005)	0.0198 (0.0001)	0.5387 (0.0000)
SI	0.0121 (0.0004)	0.0129 (0.0002)	0.0257 (0.0013)	0.0225 (0.0004)	0.0211 (0.0008)	0.0477 (0.0004)	0.3705 (0.0000)

proposed meta-feature DS has more values (length is 70), the recommender system using this meta-feature obtained similar results when compared with other meta-feature for most of the indices analyzed.

7. Conclusion

This paper investigated the automatic recommendation of the number of clusters in a data clustering task. For such, we proposed a new methodology using MtL. Moreover, to improve the predictive performance using the recommendation, we proposed a new meta-feature based on kernel density estimation. Experiments were carried out to evaluate the predictive performance of the number of clusters recommender system using meta-feature sets from the literature and the new proposed meta-feature set. In the experiments, we used 10 ML algorithms to induce meta-models. We also analyzed the influence of number of meta-models for the performance recommendation system.

The results pointed out that the recommender system using MtL obtained a better performance than the standard recommender, used as a baseline, for most indices and error measures analyzed. Moreover, the proposed meta-feature set was able to improve the predictive performance of the recommender system for most of the assessed situations. These results were asserted by statistical hypothesis tests. Regarding the number of meta-models used in the recommender system, when at least five meta-models are used, the predictive performance becomes clearly better than the performance obtained by the baselines.

As future work, the authors would like to investigate a technique for index aggregation to improve the recommendation of the number of clusters for a new dataset. We would also like to increase the number of ML algorithms investigated for the induction of meta-models. Finally, to further improve the performance of the recommender system, other types of meta-features, able to better capture important aspects of dataset, could be investigated.

CRediT authorship contribution statement

Bruno Almeida Pimentel: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **André C.P.L.F. de Carvalho:** Writing - review & editing, Supervision.

Acknowledgments

The authors would like to thank FAPESP, Brazil (processes 2012/22608-8, 2016/18615-0 and 2017/20265-0), CAPES and CNPq, Brazil for their support.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.knosys.2020.105682.

References

[1] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[2] A.K. Jain, R.C. Dubes, et al., Algorithms for Clustering Data, Vol. 6, Prentice hall Englewood Cliffs, 1988.
[3] S. Ding, L. Cong, Q. Hu, H. Jia, Z. Shi, A multiway p-spectral clustering algorithm, Knowl.-Based Syst. 164 (2019) 371–377.
[4] P. Berkhin, A survey of clustering data mining techniques, in: Grouping Multidimensional Data, Springer, 2006, pp. 25–71.
[5] M. Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, John Wiley & Sons, 2011.
[6] S. Ding, H. Jia, M. Du, Y. Xue, A semi-supervised approximate spectral clustering algorithm based on hmrf model, Inform. Sci. 429 (2018) 215–228.
[7] T. Deng, D. Ye, R. Ma, H. Fujita, L. Xiong, Low-rank local tangent space embedding for subspace clustering, Inform. Sci. 508 (2020) 1–21.
[8] X. Xu, S. Ding, Z. Shi, An improved density peaks clustering algorithm with fast finding cluster centers, Knowl.-Based Syst. 158 (2018) 65–74.
[9] L. Wang, S. Ding, H. Jia, An improvement of spectral clustering via message passing and density sensitive similarity, IEEE Access 7 (2019) 101054–101062.
[10] H. Wang, Y. Yang, B. Liu, H. Fujita, A study of graph-based system for multi-view clustering, Knowl.-Based Syst. 163 (2019) 1009–1019.
[11] Y. Zhang, Y. Yang, T. Li, H. Fujita, A multitask multiview clustering algorithm in heterogeneous situations based on lle and le, Knowl.-Based Syst. 163 (2019) 776–786.
[12] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, J. R. Stat. Soc. Ser. B Stat. Methodol. 63 (2) (2001) 411–423.
[13] P. Brazdil, C.G. Carrier, C. Soares, R. Vilalta, Metalearning: Applications to Data Mining, Springer Science & Business Media, 2008.
[14] G. Wang, Q. Song, X. Zhang, K. Zhang, A generic multilabel learning-based classification algorithm recommendation method, ACM Trans. Knowl. Discov. Data 9 (1) (2014) 7.
[15] M. Tripathy, A. Panda, A study of algorithm selection in data mining using meta-learning., J. Eng. Sci. Technol. Rev. 10 (2) (2017).
[16] B.A. Pimentel, A.C. de Carvalho, A new data characterization for selecting clustering algorithms using meta-learning, Inform. Sci. 477 (2019) 203–219.
[17] B.A. Pimentel, A.C. de Carvalho, Statistical versus distance-based meta-features for clustering algorithm recommendation using meta-learning, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.
[18] R.B. Prudêncio, T.B. Ludermir, Meta-learning approaches to selecting time series models, Neurocomputing 61 (2004) 121–137.
[19] C. Lemke, B. Gabrys, Meta-learning for time series forecasting and forecast combination, Neurocomputing 73 (10–12) (2010) 2006–2016.
[20] J. Kanda, A.C.P.L.F. Carvalho, E.R. Hruschka, C. Soares, P. Brazdil, Meta-learning to select the best meta-heuristic for the traveling salesman problem: A comparison of meta-features, Neurocomputing 205 (2016) 393–406.
[21] A.E.-S. Ezugwu, A.O. Adewumi, M.E. Frîncu, Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem, Expert Syst. Appl. 77 (2017) 189–210.
[22] L.P.F. Garcia, A.C.P.L.F. Carvalho, A.C. Lorena, Noise detection in the meta-learning level, Neurocomputing 176 (2016) 14–25.
[23] L.P.F. Garcia, A.C. Lorena, S. Matwin, A.C.P.L.F. Carvalho, Ensembles of label noise filters: a ranking approach, Data Min. Knowl. Discov. 30 (5) (2016) 1192–1216.
[24] E. Leyva, Y. Caises, A. González, R. Pérez, On the use of meta-learning for instance selection: An architecture and an experimental study, Inform. Sci. 266 (2014) 16–30.
[25] E. Leyva, A. González, R. Pérez, Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective, Pattern Recognit. 48 (4) (2015) 1523–1537.

[26] M. Ekstrand, J. Riedl, When recommenders fail: predicting recommender failure for algorithm selection and combination, in: Proceedings of the Sixth ACM Conference on Recommender Systems, ACM, 2012, pp. 233–236.
[27] T. Cunha, C. Soares, A.C.P.L.F. Carvalho, Metalearning and recommender systems: A literature review and empirical study on the algorithm selection problem for collaborative filtering, Inform. Sci. 423 (2018) 128–144.
[28] R.G. Mantovani, A.L.D. Rossi, J. Vanschoren, B. Bischl, A.C.P.L.F. Carvalho, 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, 2015, pp. 1–8.
[29] T. Horváth, R.G. Mantovani, A.C.P.L.F. Carvalho, Effects of random sampling on SVM hyper-parameter tuning, in: International Conference on Intelligent Systems Design and Applications, Springer, 2016, pp. 268–278.
[30] F. Pinto, C. Soares, J. Mendes-Moreira, Towards automatic generation of metafeatures, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2016, pp. 215–226.
[31] P. Brazdil, R. Henery, Analysis of results, Mach. Learn. Neural Statist. Classif. (1994) 175–212.
[32] M.C.P. De Souto, R.B. Prudencio, R.G. Soares, D.S. 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 (IJCNN), IEEE, 2008, pp. 3729–3735.
[33] B.F. de Souza, Meta-apredizagem Aplicada à Classificação de Dados de Expressão Gênica (Ph.D. thesis), Instituto de Ciências Matemáticas e de Computação (ICMC), 2010.
[34] D.G. Ferrari, L.N. De Castro, Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods, Inform. Sci. 301 (2015) 181–194.
[35] M. Vukicevic, S. Radovanovic, B. Delibasic, M. Suknovic, Extending meta-learning framework for clustering gene expression data with component-based algorithm design and internal evaluation measures, Int. J. Data Min. Bioinform. 14 (2) (2016) 101–119.
[36] Determining the optimal number of clusters: 3 must know methods, 2019, https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/ (Accessed: 2019-05).
[37] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (2) (1979) 224–227.
[38] J.C. Dunn, Well-separated clusters and optimal fuzzy partitions, J. Cybern. 4 (1) (1974) 95–104.
[39] W.J. Krzanowski, Y. Lai, A criterion for determining the number of groups in a data set using sum-of-squares clustering, Biometrics (1988) 23–34.
[40] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inf. Syst. 17 (2) (2001) 107–145.
[41] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[42] A. Murari, E. Peluso, F. Cianfrani, P. Gaudio, M. Lungaroni, On the use of entropy to improve model selection criteria, Entropy 21 (4) (2019) 394.
[43] A. Kalousis, Algorithm Selection Via Meta-Learning (Ph.D. dissertation), University of Geneva, 2002.
[44] E. Fix, J.L. Hodges Jr, Discriminatory Analysis-Nonparametric Discrimination: Consistency Properties, Tech. Rep., California Univ Berkeley, 1951.
[45] N. Ancona, Classification Properties of Support Vector Machines for Regression, Technical Report, 1999.
[46] L. Breiman, R. Ihaka, Nonlinear Discriminant Analysis Via Scaling and ACE, Department of Statistics, University of California, 1984.
[47] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[48] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[49] F. Rosenblatt, The Perceptron - a Perceiving and Recognizing Automaton, Tech. Rep., Cornell Aeronautical Laboratory, 1957.
[50] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1) (1996) 267–288.
[51] D.E. Duffy, T.J. Santner, On the small sample properties of norm-restricted maximum likelihood estimators for logistic regression models, Comm. Statist. Theory Methods 18 (3) (1989) 959–980.
[52] T. Zhang, Solving large scale linear prediction problems using stochastic gradient descent algorithms, in: ICML 2004: Proceedings of the Twenty-First International Conference on Machine Learning. Omnipress, 2004, pp. 919–926.
[53] R.C. de Amorim, C. Hennig, Recovering the number of clusters in data sets with noise features using feature rescaling factors, Inform. Sci. 324 (2015) 126–145.

Please cite this article as: B.A. Pimentel and A.C.P.L.F. de Carvalho, A Meta-learning approach for recommending the number of clusters for clustering algorithms, Knowledge-Based Systems (2020) 105682, https://doi.org/10.1016/j.knosys.2020.105682.