An Analysis of Meta-learning Techniques for Ranking Clustering Algorithms Applied to Artificial Data

Rodrigo G.F. Soares*, Teresa B. Ludermir, and Francisco A.T. De Carvalho

Federal University of Pernambuco,
Center of Informatics
{rgfs,tbl,fatc}@cin.ufpe.br
http://www.cin.ufpe.br

Abstract. Meta-learning techniques can be very useful for supporting non-expert users in the algorithm selection task. In this work, we investigate the use of different components in an unsupervised meta-learning framework. In such scheme, the system aims to predict, for a new learning task, the ranking of the candidate clustering algorithms according to the knowledge previously acquired.

In the context of unsupervised meta-learning techniques, we analyzed two different sets of meta-features, nine different candidate clustering algorithms and two learning methods as meta-learners.

Such analysis showed that the system, using MLP and SVR meta-learners, was able to successfully associate the proposed sets of dataset characteristics to the performance of the new candidate algorithms. In fact, a hypothesis test showed that the correlation between the predicted and ideal rankings were significantly higher than the default ranking method. In this sense, we also could validate the use of the proposed sets of meta-features for describing the artificial learning tasks.

Keywords: Meta-learning, Clustering.

1 Introduction

Selecting suitable algorithms for solving one given problem requires, generally, a great deal of effort. In the context of Machine Learning, we can point out some tasks that one may tackle using more than one technique, such as: classification, regression, clustering. In such domain, there are many alternatives for solving particular problems. This fact raises one of the most difficult tasks in Machine Learning: predicting the performance of candidate algorithms for a given problem. Typically, the choice of which algorithm might be used relies on trial-and-error procedures or on the expensive and rare users' expertise.

Meta-learning approaches have been proposed in order to predict the performance of candidate algorithms for a given problem, so they were able to select

* This work was supported by FACEPE and CNPq - Brazilian funding agencies.

C. Alippi et al. (Eds.): ICANN 2009, Part I, LNCS 5768, pp. 131–140, 2009.

and rank these algorithms indicating to the user the best choices for solving the problem. Such meta-learning techniques offer support to the non-expert user in algorithm selection task, so that there is no need for expertise to deal with this task.

The meta-learning techniques can be used in various domains. One of the main applications of these techniques is in selection and ranking of supervised algorithms. However, just a couple of investigations about the use of these techniques in the unsupervised context were made [20,21]. Those works are the starting points of further researches in that area, although they had a specific case study.

In general, clustering data is a complex task. There are many issues about how clustering can be performed. One single dataset can have more than one cluster structure in different levels of refinement. In fact, there is no even a single definition of what a cluster may looks like. The previous works validated their framework in a particular set of clustering datasets. In this paper, we intend to study and validate the application of meta-learning techniques in the unsupervised context using a wide range of synthetic datasets, covering most of the dataset clustering structures.

In this paper, our aim is to employ the framework proposed in [21,20] with different datasets using a more general set of dataset characteristics, different sets of meta-features in the meta-learning process and a different set of candidate clustering algorithms.

The remainder of this paper is divided into four sections. Section 2 introduces basic concepts about meta-learning and some of its techniques. In Section 3, we present our meta-learning analysis in ranking and selecting clustering algorithms, showing the employed framework, the proposed sets of meta-features and the learning algorithms. Section 4 presents our experiments developed in order to perform the analysis of the unsupervised meta-learning components. Finally, in Section 5, we present some final remarks and further work.

2 Related Works and Basic Concepts

Each meta-example corresponds to a dataset and it is composed of the dataset features (meta-features or meta-attributes) and the information about the performance of one or more algorithms applied to the learning task. The set of meta-examples composes the meta-dataset, which is the input of the meta-learner.

The meta-learner is a system responsible for acquiring knowledge from a set of meta-examples and, then, predicting the performance of the algorithms for new problems. Generally, the meta-features are statistics about the training datasets or some information related to the nature of the data. Examples of these features are: number of training examples, number of attributes, normality tests, number of outliers, among others [13,5,3].

More specifically, each meta-example has, as performance information, a class attribute that indicates the best algorithm for the problem, among a set of candidates [2,14,17,18]. In such a case, the class label for each meta-example is

defined by performing a cross-validation experiment using the available dataset. The meta-learner is simply a classifier which predicts the best algorithm based on the meta-features of the problem.

There is a variety of approaches using meta-learning techniques in literature. For instance, in [12] and [11], different meta-learners are employed to predict a class label associated to the performance of the algorithms, and to recommend a ranking of the algorithms.

Moreover, in the context of unsupervised learning, a novel method was developed to use meta-learning techniques in clustering problems. In [21], the authors presented a novel framework that applies a meta-learning approach to clustering algorithms. Given a dataset, the proposed framework provides a ranking for the candidate algorithms that could be used with that dataset.

Particularly, this paper employs the previous framework in the ranking task of candidate clustering algorithms in a comprehensive range of artificial clustering problems. Additionally, we use two different sets of meta-features in this analysis.

3 Proposed Analysis

In this section, we present the meta-learning process of acquiring knowledge from various datasets and ranking the candidate clustering algorithms.

3.1 General Framework

Figure 1 presents the general architecture of systems used for selecting and ranking clustering algorithms. In order to acquire knowledge and perform the ranking process, the system has two phases: training and use.

In the training phase, the meta-learner (ML) acquires knowledge from the set of examples stored in the database (DB). This knowledge associates dataset features to the performance of the candidate clustering algorithms. The acquired knowledge may be refined as more examples are available in the DB.

In the phase of use, given a new dataset to be clustered, the feature extractor (FE) extracts the values of the dataset features. According to these values, the ML module suggests a ranking of the available candidate algorithms. For that, it uses the knowledge previously provided as a result of the training phase.

The FE module is responsible for extracting the features values of the input datasets. We present these features in the next section.

The DB stores examples of clustering datasets used in the training phase. Each example associates a dataset (represented by the chosen set of features) to the performance of the candidate algorithms. This set of examples is semi-automatically built: (1) the selection of datasets and algorithms to be considered is a manual task; (2) the extraction of the series features is automatically performed by the FE module; and (3) the performance of the candidate algorithms in the clustering of a dataset is empirically obtained by directly applying each algorithm to that dataset and evaluating the resulting clustering structures.

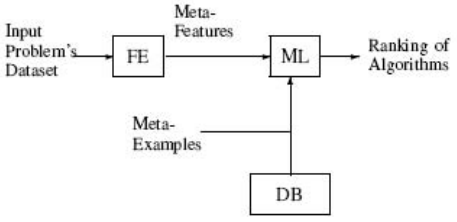

Fig. 1. General architecture

The ML module implements the chosen meta-learning approach to acquiring knowledge (training phase) to be used in the ranking of the candidate algorithms (use phase). The meta-learning approaches implement one or more machine learning algorithms to perform the mentioned task.

3.2 Meta-features

An important issue about implementing the framework is the set of meta-features used by the FE module to describe each dataset. These meta-features depend on the type of dataset under analysis. Some of them are directly related to the nature of the data. In this paper, we use artificial data to evaluate the meta-learning approach, because in this sense we know the clustering data structure. Such fact facilitates the testing and validation of the system (in further work we intend to use real data on the experiments). Then, there is no specific meta-feature describing the data, that is, the set of meta-feature used in this work may be applied in any dataset, since these meta-features are based only on statistics.

Generally, a subjective feature extraction is time consuming, requires expertise, and has a low degree of reliability, such as visual inspection of plots [1]. The presented meta-features are reliably identified, avoiding subjective analysis.

In order to avoid a time consuming selection process, we present a reasonable number of meta-features: nine relevant dataset statistics were used. Some of them were first proposed in [15] for supervised learning.

In this paper, we used two different set of meta-features. Both of them employs the Hotelling's T^2 vector statistics [10]. Given a dataset $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, the T^2 vector can be calculated as following:

$$t_i^2 = (\mathbf{x}_i - \bar{\mathbf{x}})\Sigma^{-1}(\mathbf{x}_i - \bar{\mathbf{x}}), \tag{1}$$

where $\bar{\mathbf{x}}$ is the mean vector of the dataset examples and Σ stands for the covariance matrix. With this equation, one can transform a multivariate dataset into a unidimensional array, condensing the multivariate nature in a single vector [10].

Both sets of meta-features include the following statistics:

1. $\log_{10}$ of the number examples, indicating the amount of available training data.
2. $\log_{10}$ of the number of attributes, indicating the data dimensionality.

3. Multivariate normality test. That is the proportion of T^2 (examples transformed via T^2) that are within 50% of a Chi-squared distribution (degree of freedom equals to the number of attributes describing the example). A rough indicator on the approximation of the data distribution to a normal distribution.
4. Percentage of outliers. This feature is the ratio between the number of t_i^2 farther more than two standard deviations from the mean in the T^2 vector and the total number of dataset examples.

Besides the statistics shown above, the first set of meta-features has five other values.

5. Coefficient of variance (CV) of the first quartile of each attribute. First, for each dataset attribute, we calculate its first quartile. Then, with this vector of quartiles, we compute the standard deviation of such vector and divide it by its mean.
6. CV of the second quartile. It is calculated similarly as the previous feature.
7. CV of the third quartile, computed likewise the previous values.
8. CV of the skewness of the dataset attributes. This value is computed as the previous ones, but considering the skewness of each attribute.
9. CV of the kurtosis of the dataset attributes. This feature takes into account the kurtosis of each attribute, summarizing these measures in a coefficient of variance likewise the previous meta-features.

As shown before, the first set of meta-features, denoted as M_1, has some meta-features (more precisely, meta-features 5 to 9) that are calculated using univariate statistics: quartiles, skewness and kurtosis. Then, when these measures are calculated for each attribute of a dataset, a vector of the analyzed statistic is generated. Since we do not use modal meta-features to compose the meta-dataset, we must use a single value to describe one given characteristic of a learning task dataset. Thus, in this work, in order to summarize the multivariate nature of the datasets, such meta-features rely on the simple coefficient of variance.

We also analyzed a second set of meta-features, denoted as M_2. This set have the same first four meta-features as presented before. The last five values are based on the T^2 vector. They are calculated as the three quartiles, skewness and kurtosis of the T^2 vector. Instead of using the coefficient of variance to summarize these statistics, it is expected that such vector is able to retrieve more properly the multivariate information of the data.

3.3 Algorithms

In the framework analyzed here, we must define the candidate clustering algorithms that will be applied on each learning task. An appropriate set of clustering techniques might have algorithms with different types of internal mechanisms, so that this set can deliver a variety of performances in the clustering task.

The selected candidate algorithms are: single linkage (SL), complete linkage (CL), average linkage (AL), k-means (KM), Shared Nearest Neighbors (SNN),

mixture model clustering (M), farthest first (FF), DB-scan (DB) and x-means algorithm (XM) [9,23,6,7,16].

The ranking of algorithms is a more informative way of selecting algorithm [21]. Then, using the meta-learner, we intend to predict the rank of each clustering algorithm according to the quality of the partitions generated by the algorithm under analysis. Each meta-example is labeled with nine values according to the rank of each candidate algorithm. Given a dataset to be labeled, a label is set to 1 if the corresponding algorithm had the best performance, 2 if it had the second best performance and so on, until the value 9 is given to the worst algorithm. The system uses a average ranking to deal with ties, for instance, if the two best algorithms are tied, their rank is set to 1.5.

The global error rate was used as the performance criterion since it allows a fair comparison between the clustering algorithms runs [22]. Such measure is simply the proportion of examples that fall outside the cluster that corresponds to its actual class. We considers that two algorithm are tied if the difference between them, in terms of global error rate, is less than 0.01.

Once we have composed the meta-dataset with the meta-features and the rankings, we now can define the meta-learner: the learning system that will associate the dataset characteristics (meta-features) to the ranking of the algorithms for predicting the rankings for new datasets. We analyzed two different learning methods as the meta-learner.

The first meta-learner is the Multilayer Perceptron network (MLP) used as a regressor of the rankings. The predictions are taken directly from the output nodes of the network. Each output node is responsible for delivering the rank of the corresponding algorithm, yielding the predicted ranking vector.

The Support Vector Regression (SVR) [19] were also employed as meta-learner. In this case, one SVR is trained for predicting the rank of each clustering algorithm, thus the system have nine independent regressors. The outcome of the SVR-based meta-learner is a vector with the predictions of each candidate algorithm.

4 Experiments

4.1 Description of the Datasets

In order to evaluate the proposed methodology, we generated 160 artificial datasets with the data generators available in [8]. We aim to obtain datasets in a wide representative range of cluster structures, so that the system can properly learn the performance of the algorithms.

The first generator is based on a standard cluster model using multivariate normal distributions. The Table 1 shows the parameter setup of the gaussian cluster generator. For each of the 8 combinations of cluster number and dimension, 10 different instances were generated, giving 80 data sets in all.

Due to the lack of generality of spherical clusters, we employed a second alternative cluster generator that delivers more elongated cluster shapes in arbitrarily

Table 1. Parameter setup for the gaussian cluster generator

Parameter	**Range**
Number of clusters	2,4,8,16
Dimension	2,20
Size of each cluster	uniformly in $[10, 100]$ for 2 and 4 cluster instances, and $[5, 50]$ for 8 and 16 cluster instances.

high dimensions. This second generator creates ellipsoidal clusters with the major axis at an arbitrary orientation.

The ellipsoid cluster generator delivers sets of high dimension. The Table 2 presents the parameters of the ellipsoid cluster generator. For each of the 8 combinations of cluster number and dimension, 10 different instances were generated, giving 80 data sets in all.

Table 2. Parameter setup for the ellipsoid cluster generator

Parameter	**Range**
Number of clusters	2,4,8,16
Dimension	50,100
Size of each cluster	uniformly in $[10, 100]$ for 2 and 4 cluster instances, and $[5, 50]$ for 8 and 16 cluster instances.

4.2 Evaluating the System

We executed 30 runs of each non-deterministic candidate algorithms. The number k of clusters was set to the actual class number of each dataset. We evaluate the performance of the meta-learners using the leave-one-out procedure.

The quality of a suggested ranking for a given dataset is evaluated by measuring the similarity to the ideal ranking, which represents the correct ordering of the models according to the global error rate. We employed the Spearman's rank correlation coefficient [3] to measure the similarity between a suggested and the ideal rankings.

In order to calculate this coefficient, we compute, given a meta-example i, the sum of squared differences between the predicted and ideal rankings for each clustering algorithm j as shown in the Equation 2.

$$D_i^2 = \sum_j D_{ij}^2 \tag{2}$$

And then, the average of Spearman's coefficient for the 160 meta-examples is calculated using the Equation 3:

$$SRC = \frac{1}{160} * \sum_{i=1}^{160} \{1 - \frac{6 * D_i^2}{P^3 - P}\} \tag{3}$$

where P is the number of candidate algorithms. The value of this coefficient ranges from $[-1, 1]$. The larger is the value of SRC_i, the greater is the similarity between the suggested and the ideal rankings for the dataset i.

In our implementation, we used the WEKA software to execute the MLP regressors [22] and the regression Support Vector Machine (SVR) algorithm, implemented in LIBSVM: a library for support vector machines [4].

4.3 Results

As highlighted before, we used two sets of meta-features, giving two different meta-datasets to analyze. These datasets were applied to both MLP and SVR regressors used as meta-learners. The results of both meta-learners were compared to the default ranking method. In such method, the average rank of each algorithm is suggested for every test example.

For the first meta-dataset, M_1, the Table 3 shows the means and standard deviations of SRC for each meta-learner.

Table 3. Results of the meta-learners for the M_1 meta-dataset

Meta-learner	Mean	Standard deviation
MLP	0.886	0.138
SVR	0.850	0.153
Default ranking	0.846	0.142

And for the second meta-dataset, formed by statistics of the T^2 vector, the Table 4 presents the results obtained by the tested meta-learners.

Table 4. Results of the meta-learners for the M_2 meta-dataset

Meta-learner	Mean	Standard deviation
MLP	0.891	0.137
SVR	0.883	0.152
Default ranking	0.846	0.142

For both datasets, M_1 and M_2, the rankings predicted by the MLP and SVR methods were more correlated to the ideal rankings than the default ranking method. A hypothesis test at a significance level of 5% showed that the mean of the correlation values of both MLP and SVR meta-learners were statistically higher than that obtained with the default ranking. However, there was no relevant difference between the correlation values of MLP and SVR methods for both meta-datasets. Both meta-features sets were validated using the tested meta-learners. However, further investigations about the choice of such features can lead to an improvement of the overall performance of the system.

5 Final Remarks

In this work, we employed different regression methods as meta-learners in a meta-learning approach. We also proposed two sets of meta-features. One has meta-features based on statistics extracted directly from the data and the other has meta-features calculated from the T^2 vector.

In order to evaluate the meta-learning techniques in a more comprehensive way, we used synthetic data with a wide range of cluster structures. Moreover, we applied nine well-known clustering algorithms with different internal mechanisms.

We were able to validate the use of both sets of meta-features in describing the unsupervised artificial datasets, allowing the meta-learners to successfully associate these characteristics to the performance of the clustering algorithms.

Since we could successfully instantiate the meta-learning framework described before and validate its use in artificial data, an even more comprehensive work can be done by applying other clustering algorithms, or other learning methods as meta-learners. Additionally, one can apply this meta-learning approach to datasets from other contexts.

An important issue that can have further analysis is the set of meta-features. Such issue is a open investigation point, causing a great impact in the final result of the system and it is not trivially obtained: these measure cannot rely on the class attribute.

References

1. Adya, J.A.M., Collopy, F., Kennedy, M.: Automatic identification of time series features for rule-based forecasting. International Journal of Forecasting 17(2), 143–157 (2001)
2. Aha, D.W.: Generalizing from case studies: A case study. In: Proceedings of the Ninth International Workshop on Machine Learning, pp. 1–10. Morgan Kaufmann, San Francisco (1992)
3. Brazdil, P.B., Soares, C., Da Costa, J.P.: Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. Machine Learning 50(3), 251–277 (2003)
4. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
5. Engels, R., Theusinger, C.: Using a data metric for preprocessing advice for data mining applications. In: European Conference on Artificial Intelligence, pp. 430–434 (1998)
6. Ertoz, L., Steinbach, M., Kumar, V.: A new shared nearest neighbor clustering algorithm and its applications. In: Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining, pp. 105–115 (2002)
7. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., Han, J., Fayyad, U.M. (eds.) Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press, Menlo Park (1996)

8. Handl, J., Knowles, J.: Cluster generators for large high-dimensional data sets with large numbers of clusters (2008), http://dbkgroup.org/handl/generators
9. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
10. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis, 5th edn. Prentice Hall, Englewood Cliffs (2002)
11. Kalousis, A., Hilario, M.: Feature selection for meta-learning. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 222–233. Springer, Heidelberg (2001)
12. Kalousis, A., Theoraris, T.: Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. Intelligent Data Analysis 3(5), 319–337 (1999)
13. Kalousis, A., Gama, J., Hilario, M.: On data and algorithms: Understanding inductive performance. Machine Learning 54(3), 275–312 (2004)
14. Kalousis, A., Hilario, M.: Representational issues in meta-learning. In: ICML, pp. 313–320 (2003)
15. Michie, D., Spiegelhalter, D.J., Taylor, C.C., Campbell, J.: Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River (1994)
16. Pelleg, D., Moore, A.W.: X-means: Extending k-means with efficient estimation of the number of clusters. In: Seventeenth International Conference on Machine Learning, pp. 727–734. Morgan Kaufmann, San Francisco (2000)
17. Prudêncio, R.B.C., Ludermir, T.B., de A.T. de Carvalho, F.: A modal symbolic classifier for selecting time series models. Pattern Recognition Letters 25(8), 911–921 (2004)
18. Prudêncio, R.B.C., Ludermir, T.B.: Meta-learning approaches to selecting time series models. Neurocomputing 61, 121–137 (2004)
19. Chen, P.H., Fan, R.E., Lin, C.J.: Working set selection using the second order information for training svm. Journal of Machine Learning Research 6, 1889–1918 (2005)
20. Soares, R.G.F.: The use of meta-learning techniques for selecting and ranking clustering algorithms applied to gene expression data (in portuguese). Master's thesis, Federal University of Pernambuco - Center of Informatics (2008)
21. Souto, M.C.P., Prudêncio, R.B., Soares, R.G.F., Araújo, D.A.S., Filho, I.G.C., Ludermir, T.B., Schliep, A.: Ranking and selecting clustering algorithms using a meta-learning approach. In: IEEE (ed.) Proceedings of International Joint Conference on Neural Networks, pp. 3729–3735 (2008)
22. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
23. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)