Towards a Constrained Clustering Algorithm Selection

Guilherme Alves, Miguel Couceiro, Amedeo Napoli

Towards a Constrained Clustering Algorithm Selection

Guilherme Alves, Miguel Couceiro, Amedeo Napoli

Université de Lorraine, CNRS, Inria Nancy G.E., LORIA
{guilherme.alves-da-silva, miguel.couceiro}@inria.fr, amedeo.napoli@loria.fr

Abstract. The success of machine learning approaches to solving real-world problems motivated the plethora of new algorithms. However, it raises the issue of algorithm selection, as there is no algorithm that performs better than all others. Approaches for predicting which algorithms provide the best results for a given problem become useful, especially in the context of building workflows with several algorithms. Domain knowledge (in the form of constraints, preferences) should also be considered and used to guide the process and improve results. In this work, we propose a meta-learning approach that characterizes sets of constraints to decide which constrained clustering algorithm should be employed. We present an empirical study over real datasets using three clustering algorithms (one unsupervised and two semi-supervised), which shows improvements in cluster quality when compared to existing semi-supervised methodologies.

1 Introduction

Novel machine learning algorithms are constantly being proposed. As we do not have a single algorithm that performs better than all other algorithms in all of the cases, it raises the issue of algorithm selection need to design learning workflows. Several approaches for automating algorithm selection become useful to e with this issue (Brazdil et al., 2008). For supervised machine learning tasks, e.g. classification algorithms, plenty of research works are available (Cachada et al., 2017)(Wang et al., 2014). For clustering tasks, which is traditionally an unsupervised task, few methods have been proposed (Pimentel and de Carvalho, 2019).

In some cases, we have domain knowledge available, for instance, a set of constraints. A well-known approach to specify constraints is in the form of instance-level constraints, which are composed of two types: *must-links* and *cannot-links*. A must-link means that two instances should be assigned to the same cluster. A cannot-link implies that the instances cannot belong to the same cluster. Constraints are used to guide the clustering process to improve the quality of the obtained clusters. Research works have extended classical (unsupervised) clustering algorithms to be able to deal with constraints, for instance, COP-KMEANS is the first extension of K-MEANS that can process a set of instance-level constraints (Wagstaff et al., 2001). Some years later, Bilenko et al. (2004) has integrated metric learning proposing the algorithm MPCK-MEANS. Apart from partitional methods, the widely used density-based clustering algorithm DBSCAN has also been extended in the semi-supervised clustering method C-DBSCAN (Ruiz et al., 2007).

Despite the many constrained clustering algorithms proposed in the literature, we are not aware of any contribution towards the automated selection of *constrained* clustering algorithms. To our knowledge, only one research work has proposed a constraint-based metric, Constraint Based Overlap (CBO), to decide which clustering algorithm should be employed (Adam and Blockeel, 2017). CBO is based on how the set of constraints are overlap. The authors argue that CBO captures how difficult is the data to be separated based on a set of constraints. Nevertheless, they do not get improvements when CBO is combined with the metrics based on the unsupervised setting.

Moreover, getting constraints is costly without guarantees of improvements in terms of quality of obtained clusters. Additionally, selecting constraints improperly may deteriorate the constrained clustering algorithm performance (Davidson et al., 2006). In order to cope with this issue, in the active clustering literature, different strategies have been proposed to select informative constraints based on uncertainty (Mallapragada et al., 2008), (Xiong et al., 2014) and k-nearest neighbor graph (Vu et al., 2010).

In this research work, we combine CBO with features based on heuristics for selecting constraints and our proposed feature based on constraints' neighbourhood to predict which constrained clustering algorithm should be used. The main hypothesis of this paper is *that combining CBO with other semi-supervised features along with our proposed feature can help on providing accurate predictions in a constrained clustering algorithm selection*.

This paper is organized as follows. Section 2 introduces the main concepts underlying this work. Section 3 explains our approach. Section 4 presents the experimental setup and discusses the results. The conclusions and future work are discussed in Section 5.

2 Background

A meta-learning system exploits knowledge obtained from previous experiences (Brazdil et al., 2008). In order to represent the previous experiences, we build a dataset named meta-dataset. Each instance of meta-dataset is a meta-instance, which is composed of features extracted from the original dataset and from the associated set of constraints. The extracted features in a meta-dataset are called meta-features. We assign to each meta-instance one class that represents the sequence of recommended algorithms based on criteria of clustering quality. The main problem is to extract meta-features, particularly extract them from a set of constraints. The first proposed meta-feature to characterize the set of constraints is CBO. It summarizes how the clusters overlap based on a given set of constraints by aggregating two components. The first component measures the overlap among short cannot-links and the second measures the overlap among pairs of must-link and cannot-link close to each other.

Let k be a positive integer, let $d(\cdot,\cdot)$ be a distance function and $\mathcal{D} = \{x_i\}_{i=1}^{n}$ a dataset. Given the sets of constraints $ML = \{c_t\}_{t=1}^{m}$ and $CL = \{c_t\}_{t=1}^{m'}$, where $c_t = (x_i, x_j), i \neq j$, let ϵ_i be the distance between instance x_i and the k-th nearest neighbour of x_i. The CBO over $\mathcal{D}$ w.r.t. ML and CL is defined as follows

$$CBO(\mathcal{D}, ML, CL) = \frac{\sum_{c \in CL} score(c) + \sum_{c_i \in CL, c_j \in ML} score(c_i, c_j)}{\sum_{c \in CL \cup ML} score(c) + \sum_{c_i \in CL, c_j \in CL \cup ML} score(c_i, c_j)} \quad (1)$$

Name	Heuristic	Reference
Min-Max	Uncertainty	(Mallapragada et al., 2008)
Ability to Separate between Clusters (ASC)	k-nearest neighbor graph	(Vu et al., 2010)
Normalized Point-based Uncertainty (NPU)	Uncertainty	(Xiong et al., 2014)

TAB. 1: Strategies for selecting constraints employed in this research work.

where $score(c) = s(x_i, x_j)$ and $score(c_i, c_j) = s(x_{i1}, x_{j1}) \times s(x_{i2}, x_{j2})$ for

$$s(x_i, x_j) = \begin{cases} 1 - \frac{d(x_i, x_j)}{\max(\epsilon_i, \epsilon_j)} & \text{if } d(x_i, x_j) \leq \max(\epsilon_i, \epsilon_j) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

We argue that we can employ the heuristics for selecting constraints in the algorithm selection. In our problem, instead of using the before-mentioned strategies for selecting constraints, we employ the functions underlying the same approaches for estimating how informative is a given set of constraints. We thus select three well-known heuristics available in the literature to be used for constrained clustering algorithm selection. Table 1 shows the selected approaches. Each approach employs a distinct function to estimate the informativeness. For instance, ASC is designed for density-based constrained clustering algorithms as it can treat datasets with different cluster densities. Min-Max employes the radial basis function kernel to estimate the uncertainty of pairs of constraints. The last strategy, NPU, integrates a constrained clustering algorithm to refine iteratively the process of uncertainty estimation.

3 Proposed Approach

In this section, we explain our approach to building a meta-learning system for constrained clustering algorithm selection. We start from the assumption that a well-spread set of constraints can provide holistic information about the dataset in comparison to the density located set of constraints. Therefore, we propose a meta-feature that measures the distribution of shared k-nearest data instances from the set of constraints. In order to do that, we represent this meta-feature using a histogram, not only as a real number. A histogram can express the most knowledge possible about the dataset being characterized. In our case, the histogram characterizes the proportion of reachable data instances from the set of constraints. If constraints are close to each other, most of data instances share the same neighbourhood and the remaining data instances do not be computed in the histogram. On the other hand, if the set of constraints is well distributed in the data space, the overlap of neighbourhoods tend to be minimized and more data instances are considered, increasing the proportion of k-nearest data instances from the set of constraints.

Algorithm 1 presents how the histogram is built. The algorithm only requires the number of neighbours k and the set of constraints. It then builds a histogram of k bars in which each bar represents the proportion of shared k-nearest instances reachable from the set of constraints. We build a histogram for each set of constraints C, i.e., one histogram for ML and another one for CL. Each data instance that belongs to the set of constraints is processed in order to discover its k nearest neighbours. The algorithm adds the i-th neighbour to i-th bar and it counts the data instance only once. For example, Fig. 1 shows two examples of the obtained

histograms over different set of constraints w.r.t. the same dataset. One notes that, in the first example from top to bottom, the data instance b_2 is the shared neighbour of b and e. With our approach we have a global view of the number of data instances that is affected by the attraction power of must-links and the number of data instances is affected by the repulsion power of cannot-links.

We employ the Euclidean distance for extracting meta-features which depend on a distance function. We also compute the same distance between each pair of instances (x_i, x_j). We build three different histograms based on these distances and concatenate them afterwards. The first histogram is computed only from unconstrained pairs, the second is built only from pairs involved in a *must-link*, and the last one considers only *cannot-link* pairs.

Algorithm 1 Constraint neighbourhood-based histogram

1: $E \leftarrow \{\}, h \leftarrow [0, ..., 0]$
2: **for** $c \in C$ **do**
3: **for** $i \in [0, k]$ **do**
4: **for** $x \in c$ **do**
5: $N \leftarrow Nearestneighbours(x, i)$
6: $h[i] \leftarrow h[i] + \frac{|N-E|}{n}$
7: $E \leftarrow E \cup N$
8: **return** h

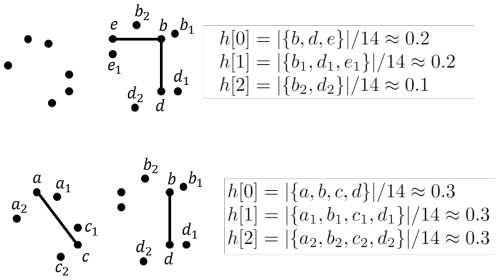


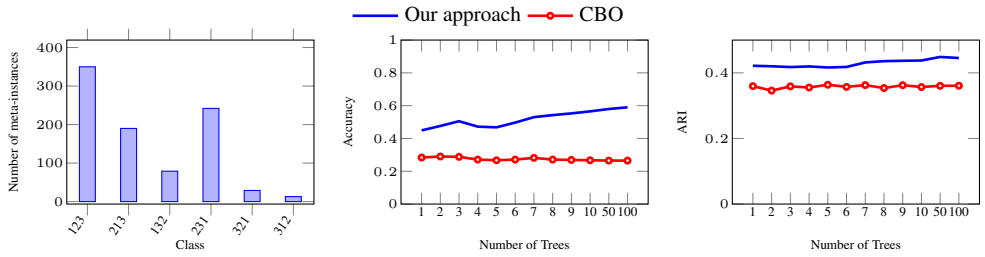
FIG. 1: The constraints arrangement in the dataset changes the obtained histogram ($k = 2$).

4 Experiments and Results

Experimental setup. In order to evaluate our approach, the experiments were conducted using 23 real datasets covering different domains from Open Machine Learning, an open scientific platform for standardizing and sharing datasets and empirical results. They are a subset from the datasets used by Pimentel and de Carvalho (2019) and Adam and Blockeel (2017). All these datasets have labeled data instances, which allows us to run our experiments. For each dataset, 5 different set of constraints were sampled according to a uniform distribution until the number of data instances were reached (0%,25%,50%,75%,100%). We also repeat the execution of each algorithm over each pair (unlabeled dataset, set of constraints) 5 times in order to catch the general behavior of the algorithms in each problem.

We compare the state of the art meta-learning system (that only uses CBO as meta-feature) with our approach, which comprises CBO, Min-Max, ASC, NPU, the constraint neighbourhood-based histogram, and the distance-based histograms. In order to evaluate both predictions, we use the leave-p-out protocol, where p is the number of meta-instances yielded from one dataset. The idea is to avoid sharing information among meta-instances that comes from the same dataset. The pool of considered algorithms were: the constrained clustering algorithms COP-KMEANS (1) and K-MEANS (2), and the traditional clustering algorithm K-MEANS (3).

Following the research works in algorithm selection, we adopted Random Forest (RF) (Breiman, 2001) as meta-learner. Therefore, we run RF over our meta-dataset where meta-instances were composed of the above-mentioned meta-features and were labeled according to Adjusted Rand Index (ARI). For example, given partitions obtained from a dataset and its set of constraints, if we have the following values of ARI: COP-KMEANS = 0.6, MPCK-MEANS



(a) (b) (c)

FIG. 2: (a) Class distribution, (b) Classification assessment, (c) Partition quality assessment.

= 0.7, and K-MEANS = 0.5, the assigned class to the associated meta-instance is "213". It means that the recommended order for employing the available algorithms is MPCK-MEANS, COP-KMEANS, and then K-MEANS. Fig. 2a shows the class distribution. Note that we do not have the best algorithm in all scenarios, which matches with we have mentioned earlier.

Empirical Results. We designed a set of experiments intending to evaluate how would be the variation of accuracy when we change the number of trees of RF. Figure 2b shows the assessment in terms of accuracy of the built meta-model using only CBO and using our approach. We can observe that for the smaller number of trees, the two approaches are competitive, yielding results with minor differences. However, the main advantages of our approach over CBO can be noted at the larger number of trees, as we have more meta-features for describing the same clustering problems.

Furthermore, we can also observe improvements in terms of ARI (see Figure 2c) . ARI is calculated based on the first position indicated in the predicted class. For instance, if the predicted class is "213", it means that algorithm 2 (MPCK-MEANS) is highly recommended and thus we run MPCK-MEANS over the dataset to compute its ARI afterwards. Therefore, the increase in the average of ARI corroborates that our meta-features contribute to a better decision of which clustering algorithm should be employed.

5 Conclusion

In this paper, we proposed an approach for constrained clustering algorithm selection using the set of meta-features: CBO, heuristics for selecting constraints, and our proposed constraint neighbourhood-based histogram. We evaluate our approach over real datasets and we achieved results that indicate improvements with respect to existing state of the art.

This work opens several avenues for future research. Our work could be extended to select the most informative meta-instances. Another interesting directions are to deal with this problem as a learning of ranking task and extend it to the online setting.

References

Adam, A. and H. Blockeel (2017). Constraint-based measure for estimating overlap in clustering. In *Benelux Conference on Machine Learning*, Volume 6, pp. 54–61.

Bilenko, M., S. Basu, and R. J. Mooney (2004). Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, pp. 11. ACM.

Brazdil, P., C. G. Carrier, C. Soares, and R. Vilalta (2008). *Metalearning: Applications to data mining*. Springer Science & Business Media.

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32.

Cachada, M., S. M. Abdulrahman, and P. Brazdil (2017). Combining feature and algorithm hyperparameter selection using some metalearning methods. In *AutoML@PKDD/ECML*, pp. 69–83.

Davidson, I., K. L. Wagstaff, and S. Basu (2006). Measuring constraint-set utility for partitional clustering algorithms. In *PKDD*, pp. 115–126. Springer.

Mallapragada, P. K., R. Jin, and A. K. Jain (2008). Active query selection for semi-supervised clustering. In *ICPR*, pp. 1–4. IEEE.

Pimentel, B. A. and A. C. de Carvalho (2019). A new data characterization for selecting clustering algorithms using meta-learning. *Information Sciences 477*, 203–219.

Ruiz, C., M. Spiliopoulou, and E. Menasalvas (2007). *C-DBSCAN: Density-Based Clustering with Constraints*, Volume 4482 of *LNCS*. Berlin, Heidelberg: Springer.

Vu, V., N. Labroche, and B. Bouchon-Meunier (2010). Boosting Clustering by Active Constraint Selection. In *ECAI*, Lisbon, Portugal.

Wagstaff, K., C. Cardie, S. Rogers, S. Schrödl, et al. (2001). Constrained k-means clustering with background knowledge. In *ICML*, Volume 1, pp. 577–584.

Wang, G., Q. Song, X. Zhang, and K. Zhang (2014). A generic multilabel learning-based classification algorithm recommendation method. *ACM TKDD 9*(1), 7.

Xiong, S., J. Azimi, and X. Z. Fern (2014). Active Learning of Constraints for Semi-Supervised Clustering. *IEEE TKDE 26*(1), 43–54.

Résumé

Le succès des approches d'apprentissage automatique pour résoudre les problèmes du monde réel a motivé une pléthore de nouveaux algorithmes. Cependant, cela soulève le problème de la sélection des algorithmes, puisqu'il n'ya pas un seule algorithme qui soit toujours plus performant que tous les autres. Les approches permettant de prédire quels algorithmes fournissent les meilleurs résultats pour un problème donné deviennent utiles, en particulier dans le cadre des workflows avec plusieurs algorithmes. Les connaissances du domaine (sous forme de contraintes et de préférences) doivent également être prises en compte et utilisées pour guider le processus et pour améliorer les résultats. Dans ce travail, nous proposons une approche de méta-apprentissage qui caractérise des ensembles de contraintes pour décider quel algorithme de clustering contraint doit être utilisé. Nous présentons une étude empirique sur des ensembles de données réels utilisant trois algorithmes de clustering (un non supervisé et deux semi-supervisés) et qui montre l'amélioration de la qualité des clusters obtenus par rapport aux méthodologies semi-supervisées existantes.