Dealing with overlapping clustering: a constraint-based approach to algorithm selection

Antoine Adam and Hendrik Blockeel

KULeuven, Department of Computer Science, Celestijnenlaan 200A, 3001 Leuven, Belgium

Abstract. When confronted to a clustering problem, one has to choose which algorithm to run. Building a system that automatically chooses an algorithm for a given task is the algorithm selection problem. Unlike the well-studied task of classification, clustering algorithm selection cannot rely on labels to choose which algorithm to use. However, in the context of constraint-based clustering, we argue that using constraints can help in the algorithm selection process. We introduce CBOvalue, a measure based on must-link and cannot-link constraints that quantifies the overlapping in a dataset. We demonstrate its usefulness by choosing between two clustering algorithm, EM and spectral clustering. This simple method shows an average performance increase, demonstrating the potential of using constraints in clustering algorithm selection.

Keywords: clustering, algorithm selection, constraints

1 Introduction

Constraints have been used to improve clustering performance by incorporating some background knowledge in a clustering problem. In a study on constraint-based clustering, Davidson et al. [4] show that using constraints can sometimes decrease this performance. They introduce the notion of coherence between constraints, and show that the more incoherent a constraint set is, the more chance it has to decrease clustering performance. Two constraints are called incoherent if they carry information that is a priori contradictory. For instance, in figure 1, the must-link constraint (in blue) implies that the left area must be clustered with the right area, while the cannot-link constraint (in red) says the opposite.

Fig. 1. Incoherent constraints

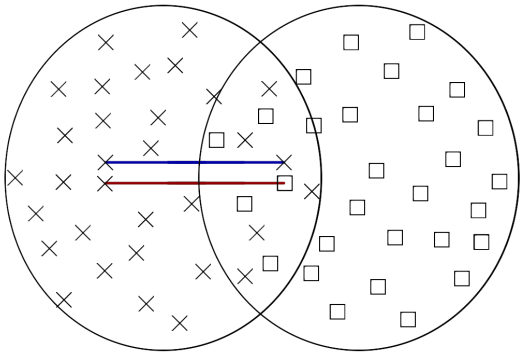

Fig. 2. Overlapping clusters can lead to incoherent constraints

Beyond the possible presence of noise in the data, a problem that we will ignore in this paper, we identified other circumstances where such incoherent constraints can appear: overlapping clusters, as shown in figure 2. Overlapping clusters is an issue that is not often tackled by clustering algorithms. Some state-of-the-art algorithms such as spectral clustering [18], which is very good at discovering arbitrary shaped clusters, will fail in the presence of overlapping. On the contrary, the EM [6] algorithm has a bias towards spherical clusters but can handle overlapping quite well as we show in section 3. As an example, we artificially created a cross dataset, see figure 3, where two clusters overlap in the middle. With a few random initialisations, EM is always able to find the correct clusters, while spectral clustering always fails. What is more, the model built by the EM algorithm incorporates the uncertainty about the cluster assignments in the overlapping area.

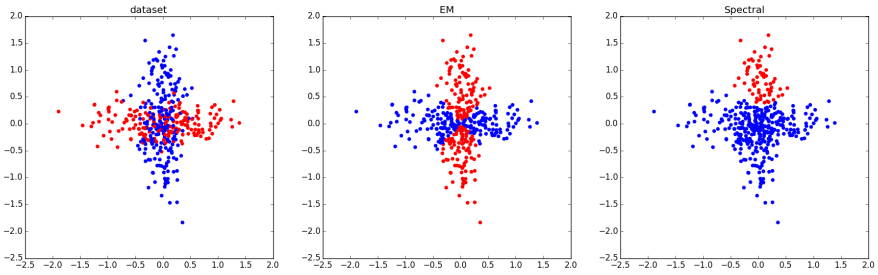

Fig. 3. Cross dataset. Colours are from left to right: the generated clusters, clusters found by EM, clusters found by spectral clustering.

This toy example illustrates the variety of clustering algorithms: different algorithms will produce different partitionings. Moreover, in a real clustering

problem, we cannot say one of these partitionings is better as we do not know the true labels. Even on the same dataset, two users might be interested in a different partitioning of the data. Only if some constraints are specified can we build a system that selects the algorithm best fitting a user requirements.

In this paper, we present some preliminary results in this direction. We introduce the CBOvalue to measure the overlapping from must-link and cannot-link constraints. We use this measure as a meta-feature in a basic meta-learning system that chooses between EM and spectral clustering. The goal of the paper is not to present an advanced meta-learning system, but to show the potential of using constraints in clustering algorithm selection.

The content of the paper is organised as follows. In section 2, we present some related work. In section 3, we define more concretely what we call overlapping and show through experiments that EM performs better than spectral clustering when it occurs. In section 4, we introduce the CBOvalue, an overlapping measure based on equivalence constraints. In section 5, we show that a simple algorithm selection method based on this measure increases clustering performance. In section 6, we draw some conclusions and leads for future work.

2 Related work

2.1 Constraint-based clustering

Clustering is the unsupervised learning task of identifying groups of similar instances in a dataset. Although these groups are initially unknown, some information can be available as to what the desired solution is. This information takes the form of constraints on the resulting clusters. These constraints can be provided to the clustering algorithm to guide the search towards a more desirable solution. We then talk about constraint-based, constrained, or semi-supervised clustering.

Constraints can be defined on different levels. On a cluster level, one can ask for clusters that are balanced in size, or that have a maximum diameter in space. On an instance level, one might know some partial labelling of the data. A well-used type of constraints are must-link and cannot-link constraints, also called equivalence constraints. These are pair-wise constraints which state that two instances must be or cannot be in the same cluster.

Multiple methods have been developed to use these constraints, some of which are mentioned below. A metric can be learnt that complies with the constraints [2]. The constraints can be used in the algorithm for the cluster assignment in a hard [19] or soft way [13], [15], [20]. Some hybrid algorithms use constraints for both metric learning and clustering [3], [9]. Other approaches include constraints in general solver methods like constraint programming [7] or integer linear programming [1].

2.2 Algorithm selection for clustering

Not much research has been conducted on algorithm selection for clustering. Existing methods usually predict the ranking of clustering algorithms [5], [16],

[14] [8]. The meta-features used are unsupervised and/or domain-specific. None of these approaches use constraints.

3 Overlapping clustering

3.1 Overlapping

We talk about overlapping when two clusters are present in the same area of the data space. It is a local property of a dataset as it happens in some parts only. Several reasons can produce overlapping clusters: there might be noise in the data, the features may not capture all the necessary information to clearly separate clusters or the overlap may be inherent to the processes that produced the data. It is a problem for algorithms that assume a clear separation of the clusters, or at least a zone of lower density points. As already mentioned for the cross dataset, spectral clustering cannot cluster it correctly. With a few random initialisations, EM always finds the right partition and what is more, the model includes that the cluster assignment is uncertain in the overlapping area.

3.2 Rvalue

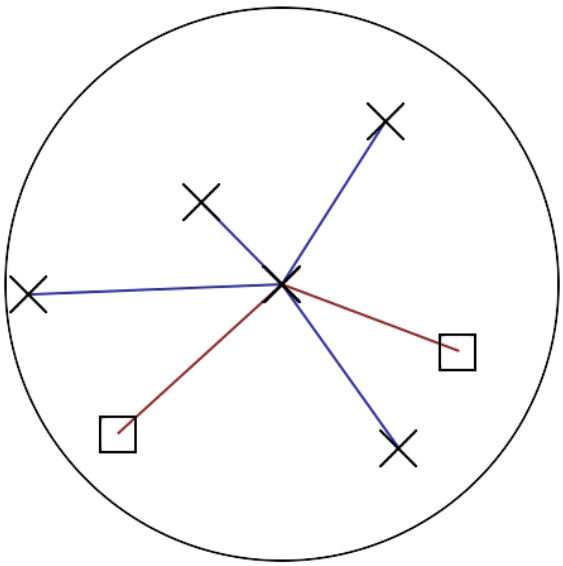

Fig. 4. Rvalue, $k = 6$, $\theta = 1$.

To numerically measure overlapping, we use the Rvalue introduced by [11]. The principle is illustrated in figure 4. For each object of a dataset, the labels of

its k neighbours are checked. If stricly more than θ are from another class, it is counted as overlapped. The Rvalue of the dataset is the proportion of overlapped objects. It is a local measure that requires two parameters, k and θ. In all our experiments, we use $k = 6$ and $\theta = 1$, i.e. we allow one neighbour to be of another class. This limits the false overlapping measurement when two clusters are next to each other but not overlapping. As an example, the cross dataset figure 3 has an Rvalue of 0.41 which means that 41% of the data points are overlapping. Figure 5 shows the distribution of the Rvalue for 14 datasets from the UCI repository, namely *iris*, *glass*, *ionosphere*, *wine*, *vertebral*, *ecoli*, *seeds*, *students*, *yeast*, *zoo*, *breast cancer wisconsin*, *mammographic*, *banknote*, *haberman*. Each feature of these datasets is normalised to an average of 0 and standard value of 1 and the metric used is the euclidean distance. This normalisation and metric are kept throughout all experiments. We can see from this figure that overlapping is not uncommon in real world datasets.

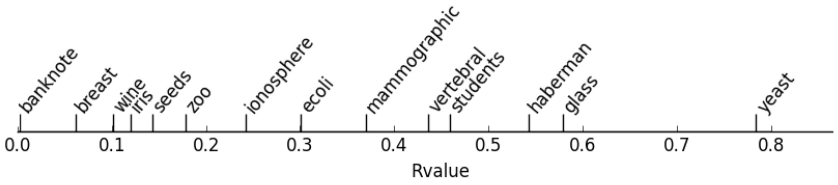

Fig. 5. Rvalue of UCI datasets, $k = 6$, $\theta = 1$.

3.3 Clustering performance

We now compare two clustering algorithms, namely EM [6] and spectral clustering [18] that we will call SC. EM is run 10 times with randomly initialised gaussians while SC is run with various parameter settings. The right number of clusters was given to both algorithms, whose performances were measured in terms of ARI (Ajusted Rand Index, [10]) and AMI (Ajusted Mutual Information, [17]). The best run was kept for comparison, as we want to compare the potential of each algorithm. On figure 6, we show the ARI of EM (in red) and SC(in blue) on the same datasets, as well as on 22 artificially made datasets.

As expected, both algorithms lose performance when overlapping increases. However, EM decreases more slowly than SC, as it is presented in table 1. These results show that EM can handle overlapping better than SC.

4 Detecting overlapping from constraints

In a clustering problem, the Rvalue cannot be directly computed as the labels are unknown. However, a user might have some partial knowledge of the clustering

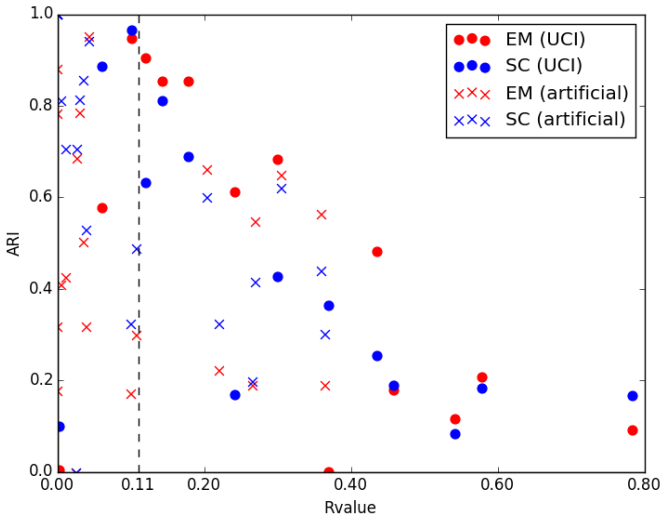

Fig. 6. Clustering performance vs Rvalue.

UCI	EM	SC
$Rvalue < 0.11$	0.509	**0.65**
$Rvalue > 0.11$	**0.452**	0.36
ALL	EM	SC
$Rvalue < 0.11$	0.511	**0.728**
$Rvalue > 0.11$	**0.443**	0.38

Table 1. Average clustering performance measured with ARI.

he is looking for. This is the setting of semi-supervised clustering, presented in section 2. We now present our method to detect overlapping based on these constraints. Like the Rvalue, it is based on the idea that overlapping is a local property.

4.1 CBOvalue: Constraint-Based Overlapping value

Overlapping translates in two cases in terms of equivalence constraints: one short cannot-link constraint or two close parallel must-link and cannot-link constraints.

CLOvalue: Cannot-Link Overlapping value. A short cannot-link means that in a close neighbourhood, two points are in two distinct clusters. Figure 7 illustrates the principle. For a cannot-link constraint cl between points x_1 and x_2, we define

$$CLOvalue(cl) = exp(-\frac{1}{2}(\frac{dist(x_1, x_2)}{max(\epsilon_1, \epsilon_2)})^p)$$

where ϵ_i the distance between x_i and it's k^{th} nearest neighbour.

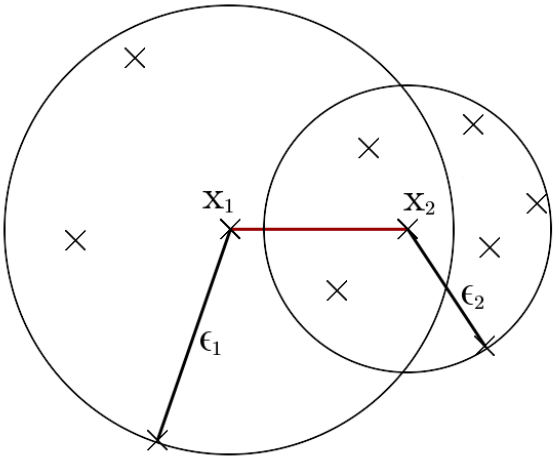

Fig. 7. CLOvalue for k=6.

Unlike for the Rvalue, we take a soft approach with the exponential because experience showed that for a limited number of constraints a hard approach was too sensitive to noise. However, the usual $p = 2$ of a gaussian turned out to be a bit too soft hence we also experiment with $p = 4$. This provides a soft neighbourhood with still a major drop at the epsilon. Using $k = 6$ produced relatively low values, so we also consider a broader neighbourhood by raising k to 10.

With CL the set of cannot-link constraints, we define

$$CLOvalue = \frac{1}{|CL|} \sum_{cl \in CL} CLOvalue(cl)$$

MLCLOvalue: Must-Link and Cannot-link Overlapping value. The case of two close must-link and cannot-link constraints was shown figure 2. Figure 8 illustrates the principle of the measure. It is defined for a cannot-link constraint cl between points x_1 and x_2 and a must-link constraint ml between two other points. We name these points x_3 and x_4 such that $dist(x_1, x_3) + dist(x_2, x_4) \leq dist(x_1, x_4) + dist(x_2, x_3)$. This ensures that we pair up neighbour points together. For instance in figure 8, we want to compare x_3 with x_1 and not x_2. We then define

$$MLCLOvalue(ml, cl) = \frac{exp(-\frac{1}{2}(\frac{d_1+d_2}{max(\epsilon_1,\epsilon_3)+max(\epsilon_2,\epsilon_4)})^p)}{2}$$

where ϵ_i the distance between x_i and its k^{th} neighbour, $d_1 = dist(x_1, x_3)$ and $d_2 = dist(x_2, x_4)$.

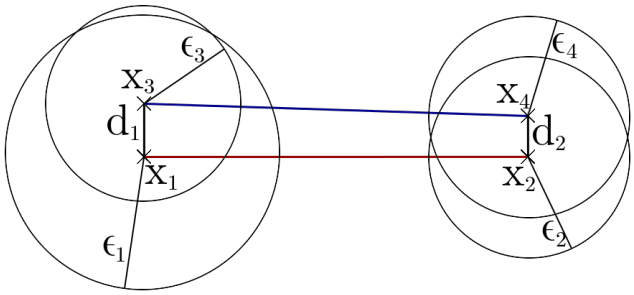

Fig. 8. MLCLOvalue.

If CL is the set of cannot-link constraints and ML the set of must-link constraints, we define

$$MLCLOvalue = \frac{1}{|CL| \times |ML|} \sum_{cl \in CL, ml \in ML} MLCLOvalue(ml, cl)$$

$$CBOvalue = \frac{CLOvalue + MLCLOvalue}{2}$$

For each dataset, we randomly generated 100 equivalence constraints from the real classes and we computed the CBO value for $k \in \{6, 10\}$. Figure 9 plots the CBO-value versus the Rvalue. The correlation is not perfect, but is enough for the algorithm selection as we will see in the next section.

5 Algorithm selection

Now that we have an overlapping measure from the constraints, we can build a system that picks which algorithm to use based on this measure. For each parameter setting, we put a threshold at the optimal position in terms of ARI. For example on figure 10 where the CBOvalue is computed with k=6 and p=4, we put a threshold at 0.011. If the CBOvalue is bigger, we use EM, otherwise we use SC. We call this method AS for Algorithm Selection. To provide an upper bound, we compute the performance of an oracle that would always pick the algorithm with highest performance. Table 2 compares the average performance of EM, SC, AS, and oracle. To visualise the improvement of the algorithm selection method, we plot on figure 11 the loss of each method for the UCI datasets. The

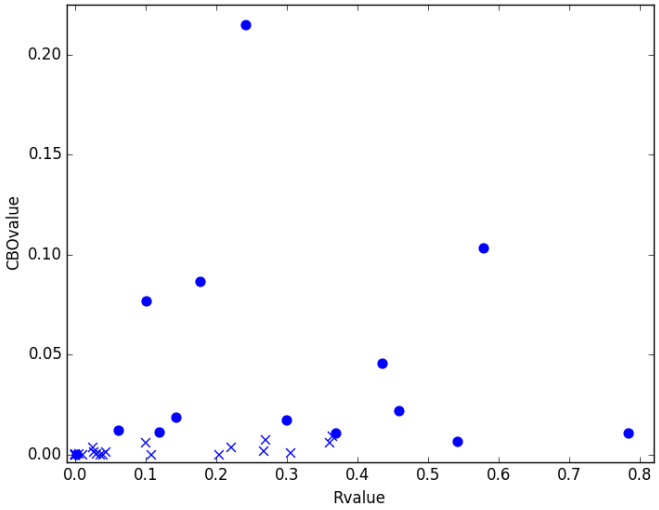

Fig. 9. CBOvalue with k=6 and p=4 vs Rvalue with k=6 and th=1.

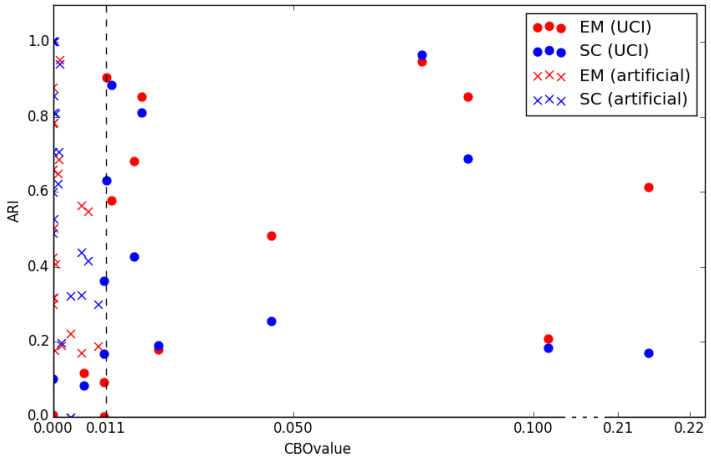

Fig. 10. Clustering performance vs CBOvalue for k=6 and p=4.

		EM	SC	AS				oracle
				k=6		k=10		
				p=2	p=4	p=2	p=4	
UCI	*ARI*	0.464	0.422	0.497	0.5	0.497	**0.522**	0.526
	AMI	0.481	0.47	0.508	**0.514**	0.487	0.487	0.534
ALL	*ARI*	0.477	0.554	0.58	0.585	0.58	**0.593**	0.605
	AMI	0.522	0.614	0.626	0.631	0.626	**0.638**	0.642

Table 2. Average clustering performance of EM, SC (Spectral Clustering), AS (a selection between the two based on the CBOvalue for several parameters), and oracle(an ideal system that would keep the best between the EM and SC).

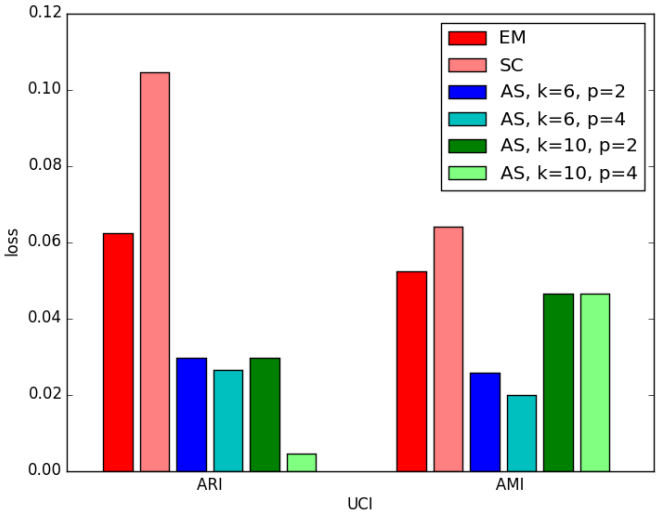

Fig. 11. Performance loss to oracle.

loss is simply the difference between the average performance of a method and the oracle average performance.

In all experiments, AS performs on average better than EM and SP, in terms of ARI or AMI. The meta-learning system used here is very simplistic: we consider only one meta-feature and two clustering algorithms. However, the goal here is not so much to build a very elaborate system, but to show the potential of using constraints in clustering algorithm selection. We see here that despite the simplicity of the selection process, the Constraint-Based Overlapping value increases the average clustering performance.

6 Conclusion

In this paper, we introduced the CBOvalue to measure the amount of overlapping in a dataset based on must-link and cannot-link constraints. On the basis that the EM algorithm handles overlapping better than spectral clustering, we select which algorithm to run depending on the CBOvalue. This simple algorithm selection system shows an increase in average performance compared to the two algorithms. Through this promising result, we demonstrate the potential of using constraints in clustering algorithm selection.

More in-depth research on the CBOvalue still has to be conducted to answer remaining questions: How robust is this measure? How sensitive is it with respect to the constraint set? How does high dimensionality affect it? We should also integrate the CBOvalue in a more complex meta-learner that uses constrained and unconstrained features.

The approach we used can be generalised as follows. A first step is to identify the strong and weak point of different algorithms, in our case the fact that EM can produce overlapping clusters. In a second step, a measure is engineered based on constraints and/or data to discriminate situations where algorithms perform differently. Finally, these measures can be used as meta-features in an algorithm selection system which can then make use of the strong points of each algorithm. Despite the remaining questions on the CBOvalue, we believe the encouraging results promote the validity of this approach for the problem of clustering algorithm selection.

Acknowledgements This work is funded by the KU Leuven Research Fund (project IDO/10/012). Experiments have been implemented in python with the scikit-learn package [12].

References

1. Behrouz Babaki, Tias Guns, and Siegfried Nijssen. Constrained clustering using column generation. In *Integration of AI and OR Techniques in Constraint Programming*, pages 438–454. Springer, 2014.
2. Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6(6):937–965, 2005.
3. Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM, 2004.
4. Ian Davidson, Kiri L Wagstaff, and Sugato Basu. *Measuring constraint-set utility for partitional clustering algorithms*. Springer, 2006.
5. Marcilio CP De Souto, Ricardo BC Prudencio, Rodrigo GF Soares, Rodrigo GF De Araujo, Ivan G Costa, Teresa B Ludermir, Alexander Schliep, et al. Ranking and selecting clustering algorithms using a meta-learning approach. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 3729–3735. IEEE, 2008.

6. Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
7. Khanh-Chuong Duong, Christel Vrain, et al. Constrained clustering by constraint programming. *Artificial Intelligence*, 2015.
8. Daniel Gomes Ferrari and Leandro Nunes de Castro. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. *Information Sciences*, 301:181–194, 2015.
9. Pan Hu, Celine Vens, Bart Verstrynge, and Hendrik Blockeel. Generalizing from example clusters. In *Discovery Science*, pages 64–78. Springer, 2013.
10. Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.
11. Sejong Oh. A new dataset evaluation method based on category overlap. *Computers in Biology and Medicine*, 41(2):115–122, 2011.
12. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
13. Dan Pelleg and Dorit Baras. K-means with large and noisy constraint sets. In *Machine Learning: ECML 2007*, pages 674–682. Springer, 2007.
14. Ricardo BC Prudêncio, Marcilio CP De Souto, and Teresa B Ludermir. Selecting machine learning algorithms using the ranking meta-learning approach. In *Meta-Learning in Computational Intelligence*, pages 225–243. Springer, 2011.
15. Carlos Ruiz, Myra Spiliopoulou, and Ernestina Menasalvas. C-dbscan: Density-based clustering with constraints. In *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, pages 216–223. Springer, 2007.
16. Rodrigo GF Soares, Teresa B Ludermir, and Francisco AT De Carvalho. An analysis of meta-learning techniques for ranking clustering algorithms applied to artificial data. In *Artificial Neural Networks–ICANN 2009*, pages 131–140. Springer, 2009.
17. Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.
18. Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
19. Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.
20. Xiang Wang and Ian Davidson. Flexible constrained spectral clustering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 563–572. ACM, 2010.