Towards a Constrained Clustering Algorithm Selection

Guilherme Alves, Miguel Couceiro, Amedeo Napoli

Université de Lorraine, CNRS, Inria Nancy G.E., LORIA

Outline

- Introduction
- Background
- Proposed Approach
- Experiments and Results
- Future Directions

Context

- An increasing number of available algorithms
- Several parametrization and pre-processing approaches
- "No free lunch"

Save time by reducing the number of alternative algorithms tried out on a given problem

Meta-learning

- Appears in ML research in 90s

- A meta-learning *system* must include
  1. A learning subsystem, which **adapts with experience**.
  2. Experience is gained by exploiting meta-knowledge extracted
     - in a previous learning episode on a single dataset
     - or from **different domains or problems**.

- One case of meta-learning:
  - **Algorithm Selection** (to recommend an algorithm **automatically**)
    - The **classic** application: **classification**
    - Some research works on **clustering**

Algorithm Selection Problem (ASP)

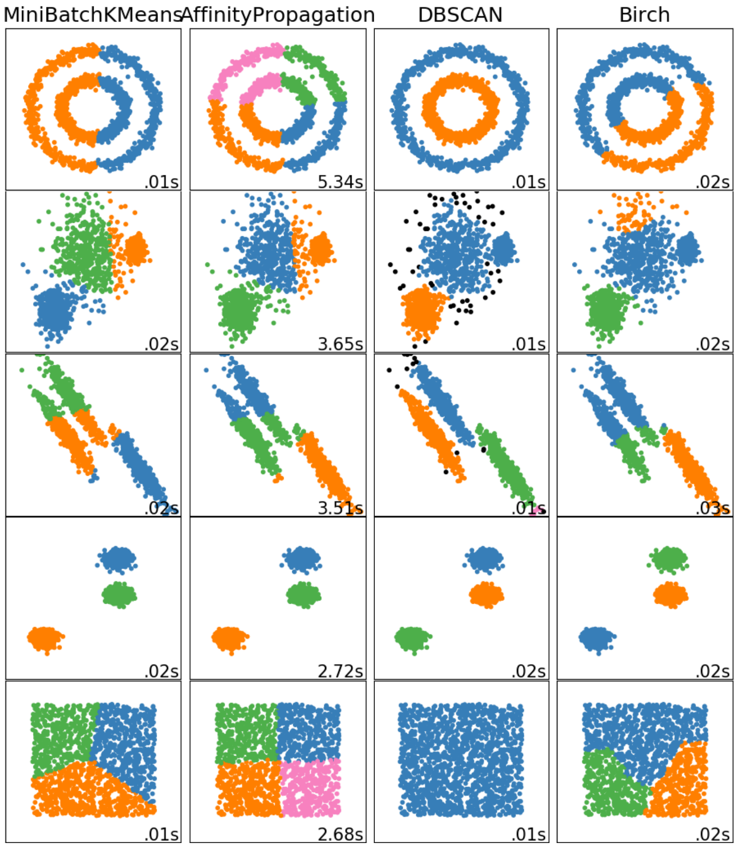
Given a set P of algorithms $p \in P$, a set D of datasets $d \in D$ and a cost metric $m: P \times D \to \mathbb{R}$

Goal: finding a mapping $s: D \to P$

such that the cost $\sum_{d \in D} m(s(d), d)$ across all datasets is optimized.

Rice, J. R. (1976). The algorithm selection problem. In Advances in computers (Vol. 15, pp. 65-118). Elsevier.

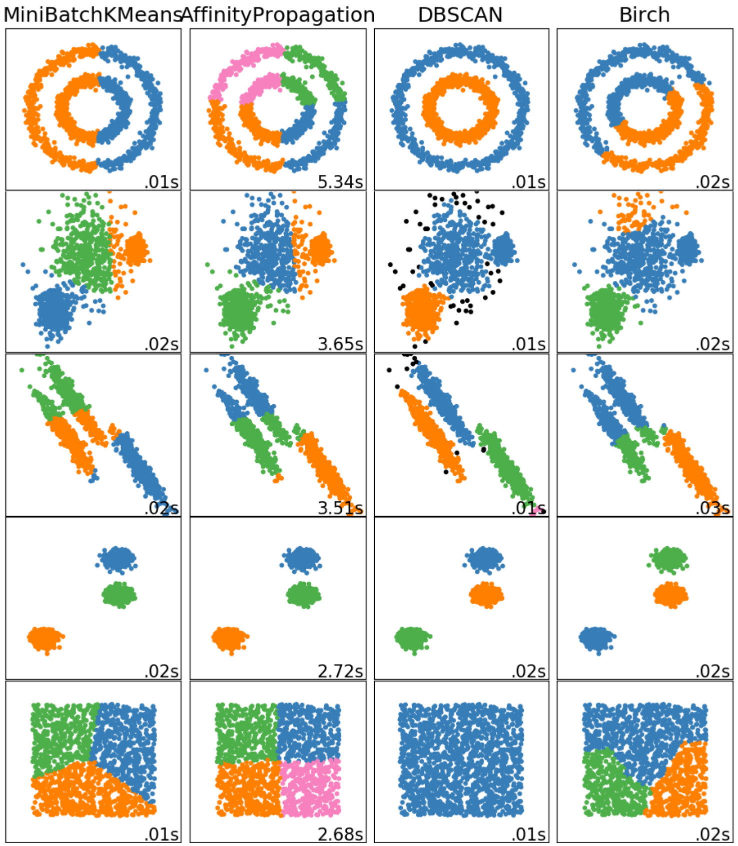
Clustering Algorithm Selection



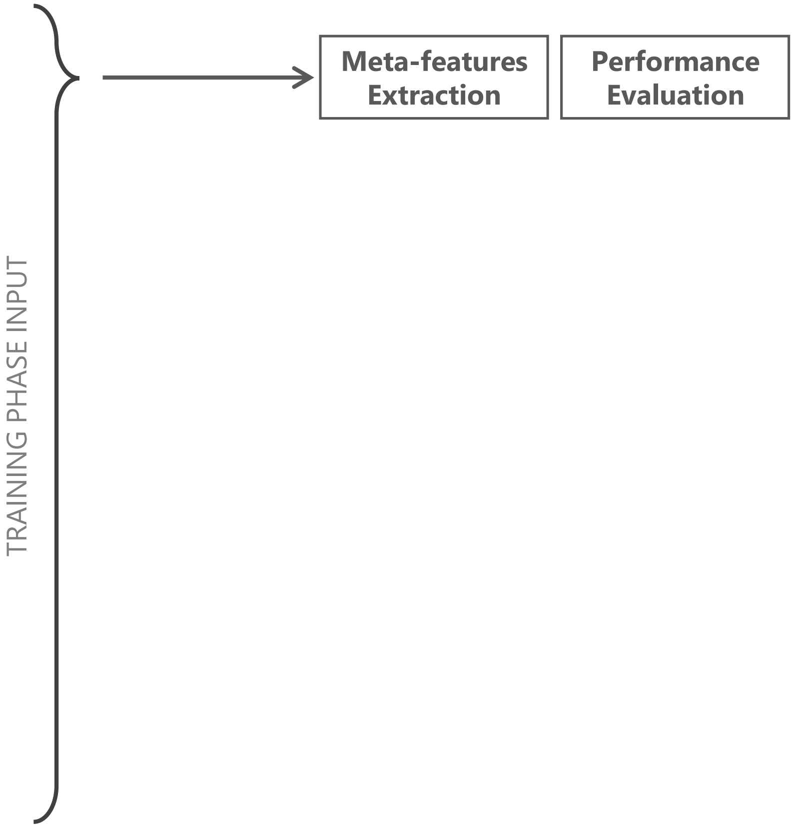
https://scikit-learn.org/stable/modules/clustering.html#clustering

Introduction

Clustering Algorithm Selection



https://scikit-learn.org/stable/modules/clustering.html#clustering



Clustering Algorithm Selection

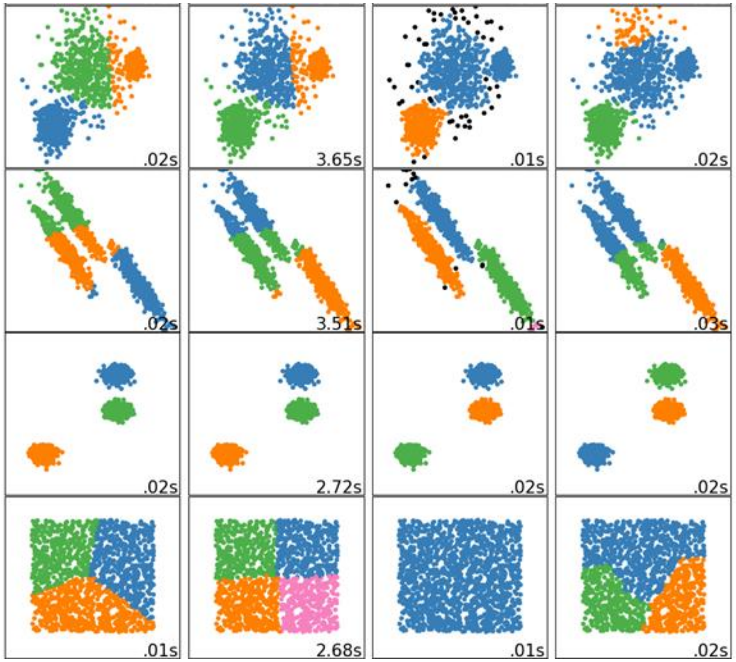
Examples: *mean, variance, std deviation, kurtosis, skewness*

Meta-data

Meta-features for D_1 | ***DBSCAN***

Examples:
a label representing the recommended algorithm,
a sequence of algorithms

Meta-features Extraction | **Performance Evaluation**

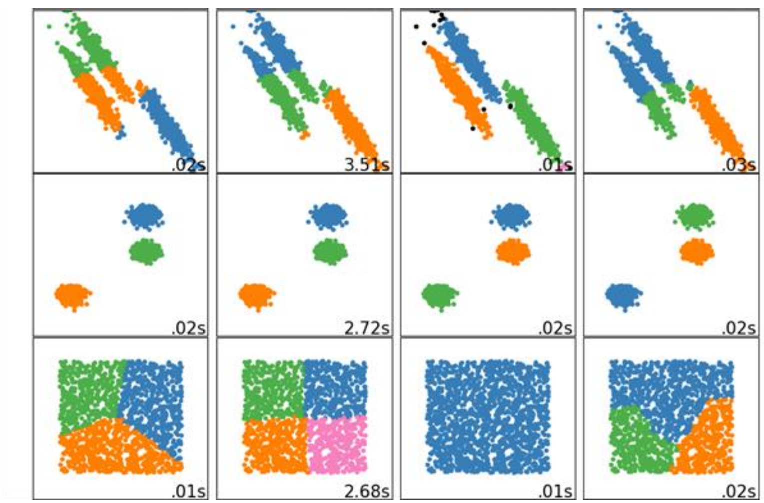


Clustering Algorithm Selection

Examples: *mean, variance, std deviation, kurtosis, skewness*

Meta-data

Meta-features for D_1	***DBSCAN***
Meta-features for D_2	***Affinity Propagation***



Examples:
a label representing the recommended algorithm,
a sequence of algorithms

Meta-features Extraction

Performance Evaluation

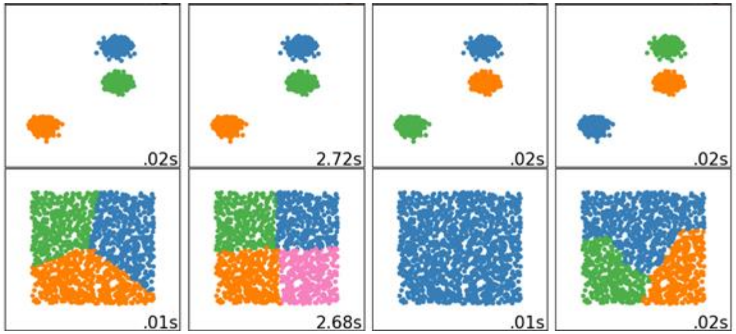
Introduction

Clustering Algorithm Selection

Examples: *mean, variance, std deviation, kurtosis, skewness*

Meta-data

Meta-features for D_1	***DBSCAN***
Meta-features for D_2	***Affinity Propagation***
Meta-features for D_3	***DBSCAN***



Examples:
a label representing the recommended algorithm,
a sequence of algorithms

Meta-features Extraction | **Performance Evaluation**

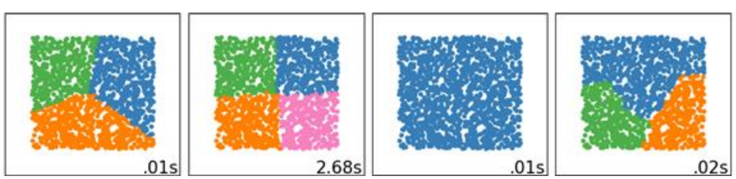
Introduction

Clustering Algorithm Selection

Examples: *mean, variance, std deviation, kurtosis, skewness*

Meta-data

Meta-features for D_1	***DBSCAN***
Meta-features for D_2	***Affinity Propagation***
Meta-features for D_3	***DBSCAN***
Meta-features for D_4	***BIRCH***



Examples:
a label representing the recommended algorithm,
a sequence of algorithms

Meta-features Extraction

Performance Evaluation

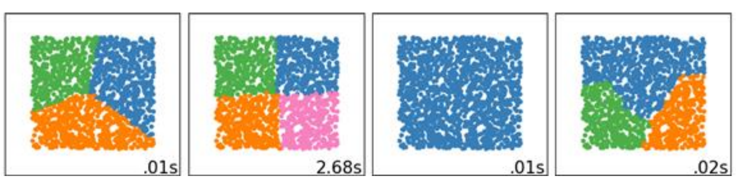
Introduction

Clustering Algorithm Selection

Examples: *mean, variance, std deviation, kurtosis, skewness*

Meta-data

Meta-features for D_1	***DBSCAN***
Meta-features for D_2	***Affinity Propagation***
Meta-features for D_3	***DBSCAN***
Meta-features for D_4	***BIRCH***



Examples:
a label representing the recommended algorithm,
a sequence of algorithms

Meta-features Extraction

Performance Evaluation

Introduction

Clustering Algorithm Selection

***Examples:** mean, variance, std deviation, kurtosis, skewness*

Meta-data

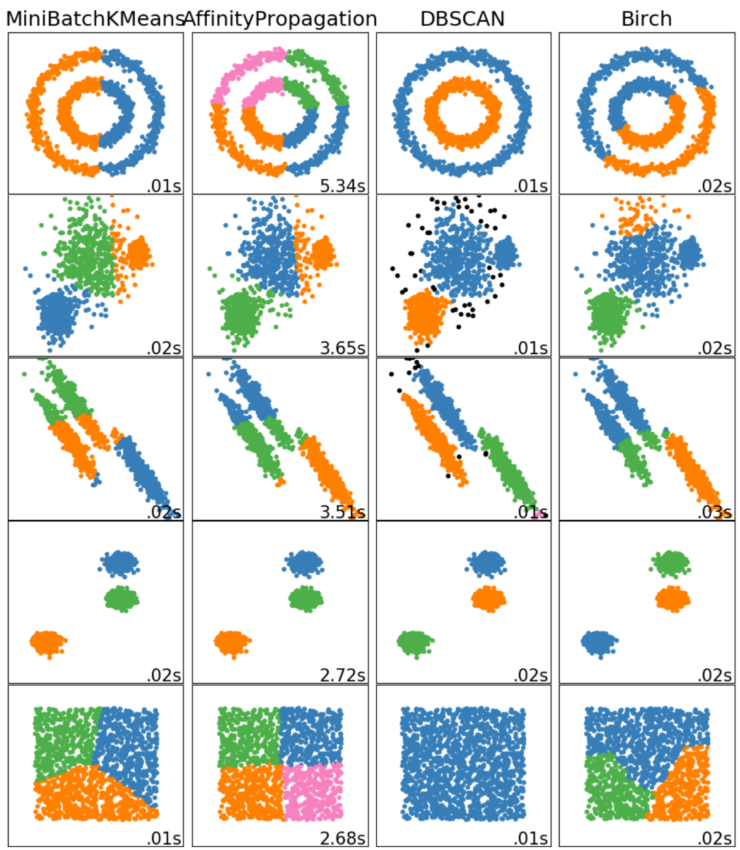
Meta-features for D_1	***DBSCAN***
Meta-features for D_2	***Affinity Propagation***
Meta-features for D_3	***DBSCAN***
Meta-features for D_4	***BIRCH***
Meta-features for D_5	***DBSCAN***

Examples:
a label representing the recommended algorithm,
a sequence of algorithms

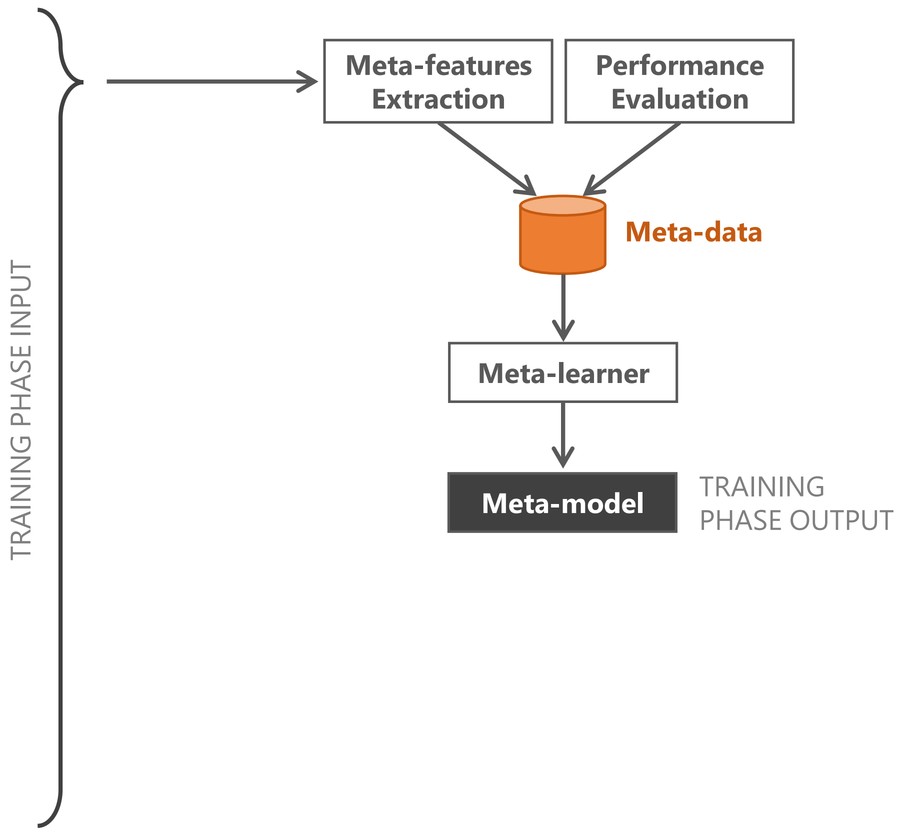
Meta-features Extraction

Performance Evaluation

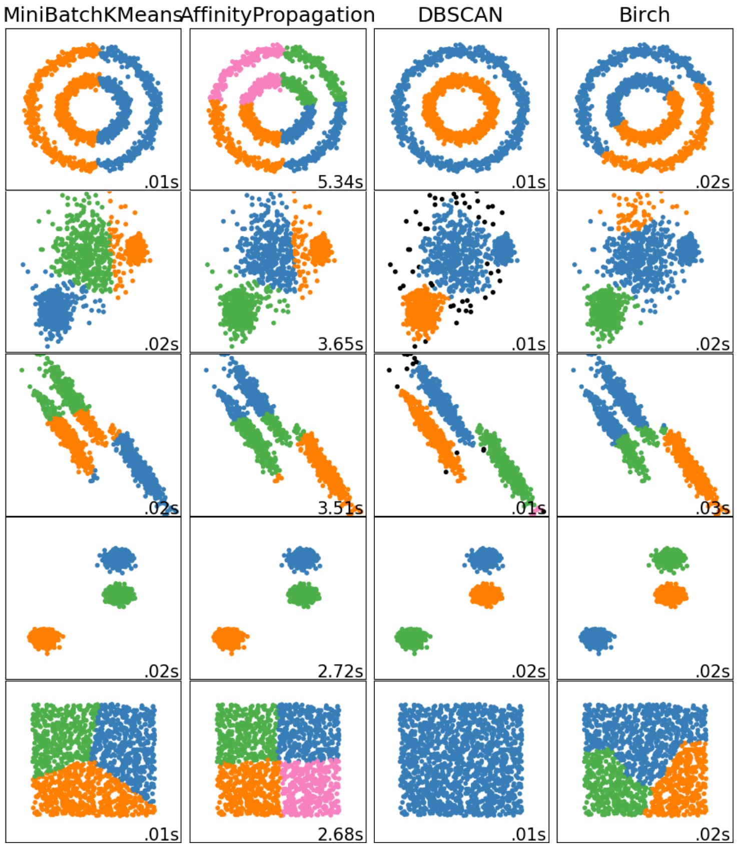
Clustering Algorithm Selection



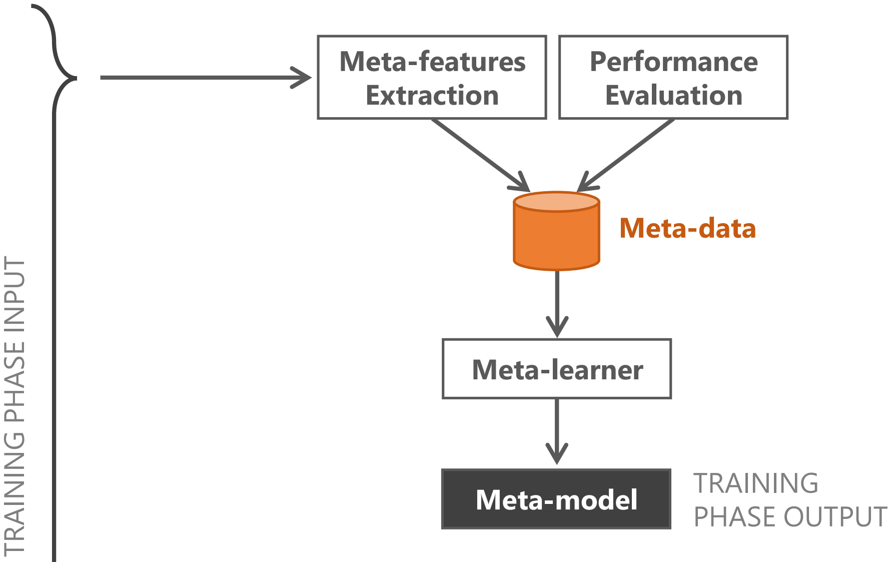
https://scikit-learn.org/stable/modules/clustering.html#clustering

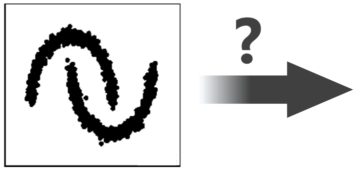


Clustering Algorithm Selection



https://scikit-learn.org/stable/modules/clustering.html#clustering





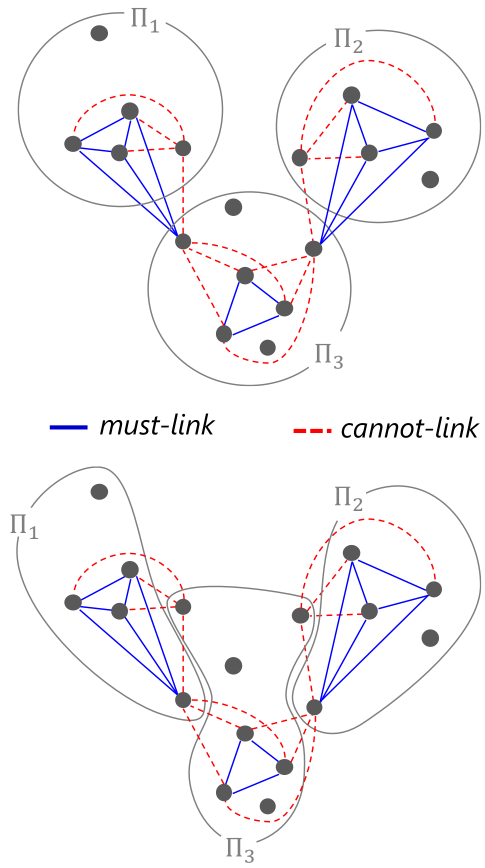
- Mini Batch K-Means
- Affinity Propagation
- DBSCAN
- Birch

Introduction

Constrained Clustering

- **Must-link**: instances must be assigned to the same cluster
- **Cannot-link**: instances should be assigned to the distinct clusters

- Extensions
  - K-means → COP-Kmeans, MPC-Kmeans



Active Learning

- Getting constraints is costly
  - **without guarantees of improvements** in terms of quality of obtained clusters

- Strategies to select informative constraints based on
  - Uncertainty: **NPU** (Normalized Point-based Uncertainty)
  - *k*-nearest neighbor graph: **ASC** (Ability to Separate between Clusters)

Problem Statement

Constrained Clustering Algorithm Selection Problem (CCASP)

Given a set P of algorithms $p \in P$, a set D of datasets $d \in D$, **the additional knowledge K** and a cost metric $m'\colon P \times D \times K \to \mathbb{R}$,

Goal: finding a mapping $s'\colon D \times K \to P$

such that the cost $\sum_{d \in D} m'(s'(d, k), d)$ across all datasets is optimized.

Literature

Constrained Clustering

- **2001** COP-K-means
- **2004** MPC-K-means
- **2008** Min-Max
- **2010** ASC (Ability to Separate between Clusters)
- **2014** NPU (Normalized Point-based Uncertainty)

Clustering Algorithm Selection

- [Ferrari & de Castro] **2015**
- (Constraint-Based Overlap) CBO **2017**
- [Pimentel & de Carvalho] **2019**

*"The question remains **open** to which extent this and other features can be derived from constraints, and to what extent this can lead to better clustering algorithm selection."* [Adam & Blockeel 2017]

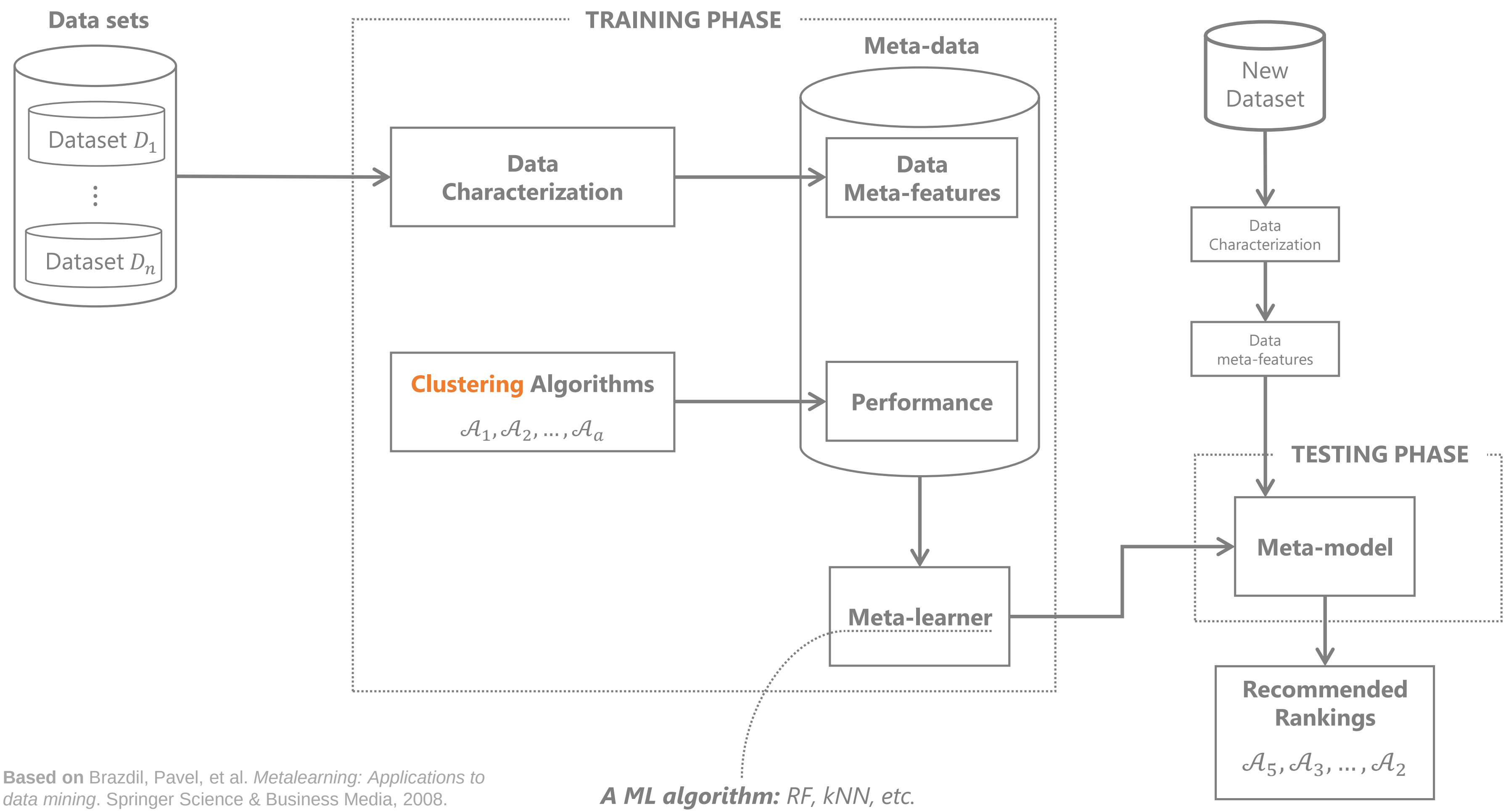
Hypothesis

Combining CBO with other constraints meta-features and our proposed meta-feature can help on providing accurate predictions in CCASP

Background

Background

Meta-learning System for Clustering



Based on Brazdil, Pavel, et al. *Metalearning: Applications to data mining*. Springer Science & Business Media, 2008.

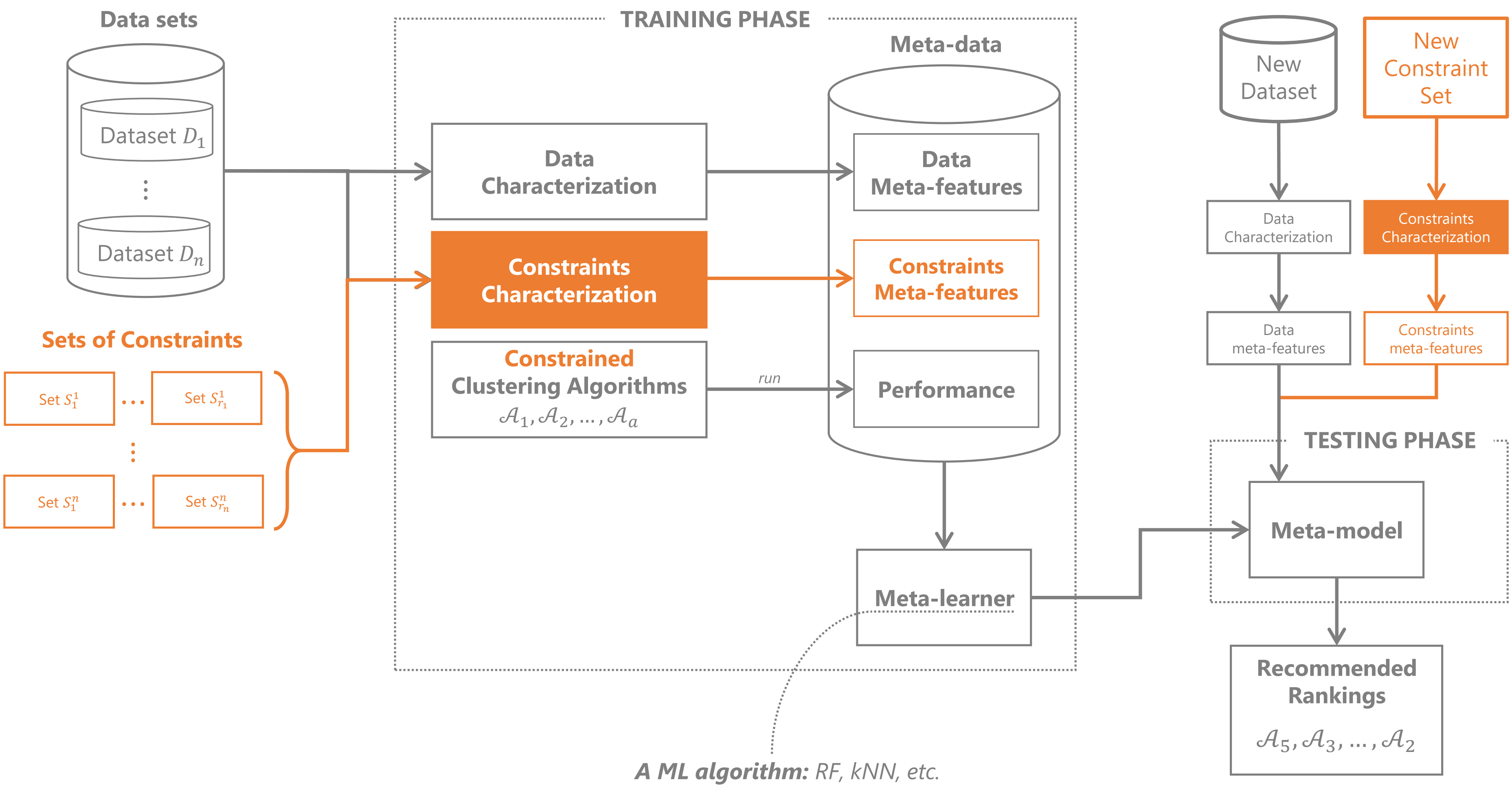
Meta-learning System for Constrained Clustering

Let’s now add one more variable in our previous scenario…

For each dataset we may have *one or more set of constraints*.

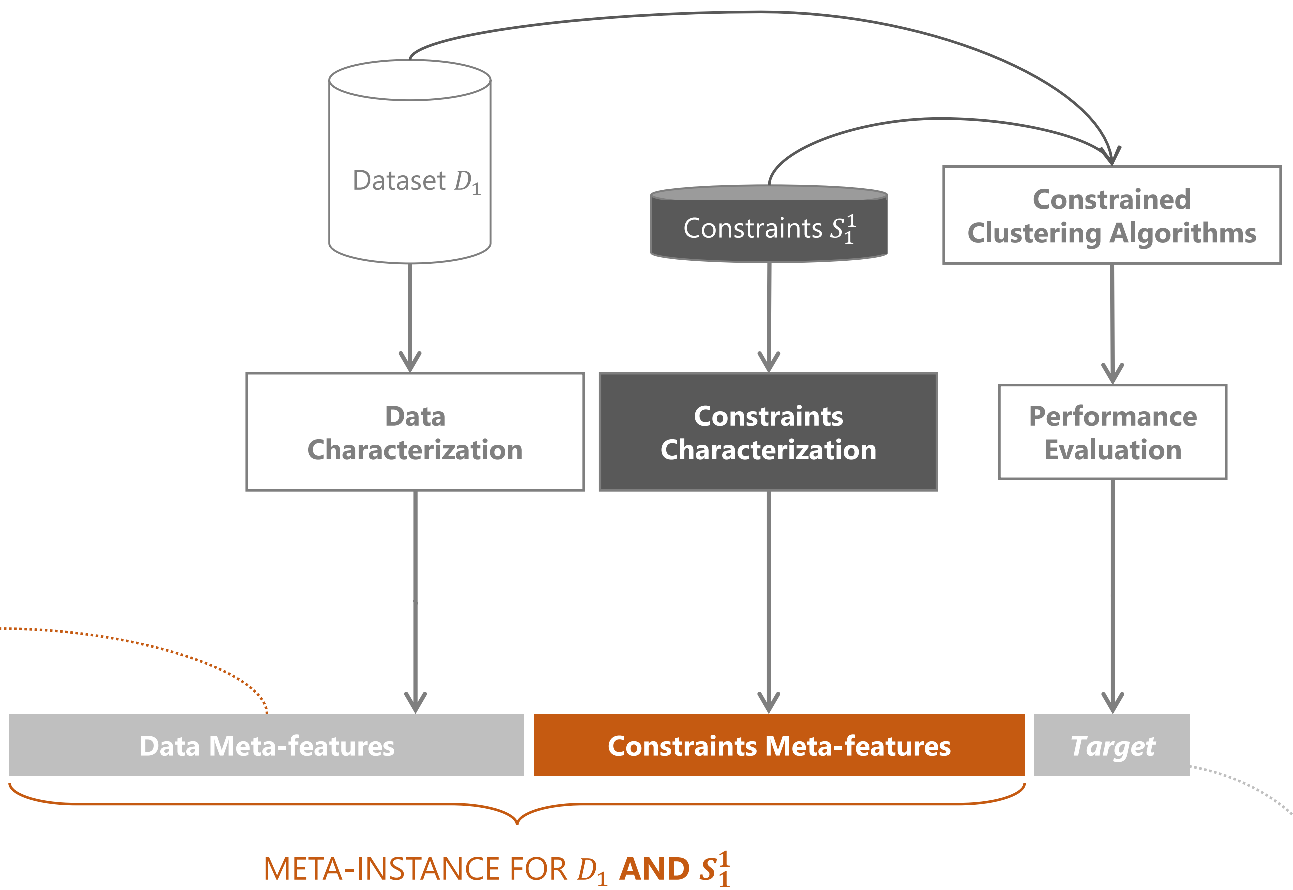
Several ways to specify constraints.
Here, we only consider pairwise constraints **must-links** and **cannot-links**

Meta-learning System for Constrained Clustering



Background

Meta-instance for CCASP



Examples:
mean, variance,
std deviation,
kurtosis,
skewness

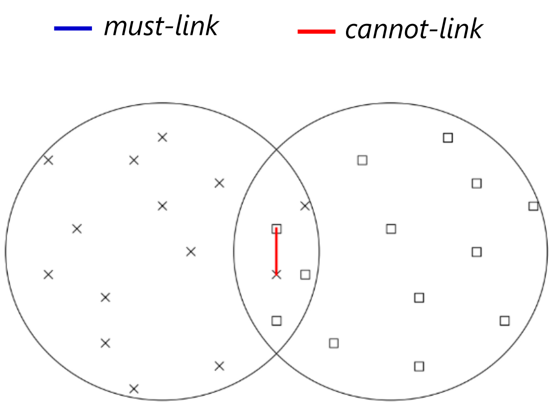
Examples:
a label representing the recommended algorithm,
a sequence of algorithms

Constraint Based Overlap (CBO)

- First proposed meta-feature for characterizing constraints
- How the clusters overlap based on a given set of constraints

Constraint Based Overlap (CBO)

- First proposed meta-feature for characterizing constraints
- How the clusters overlap based on a given set of constraints
  - the overlap among short cannot-links

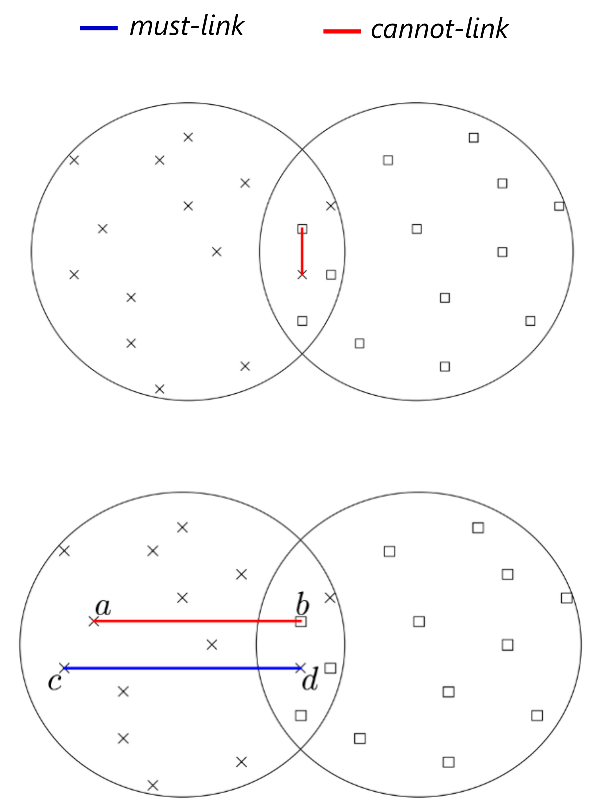


Adam, A., & Blockeel, H. (2017). Constraint-based measure for estimating overlap in clustering. In *Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning* (Vol. 6, pp. 54–61).

Background

Constraint Based Overlap (CBO)

- First proposed meta-feature for characterizing constraints
- How the clusters overlap based on a given set of constraints
  - the overlap among short cannot-links
  - the overlap among pairs of parallel must-link and cannot-link

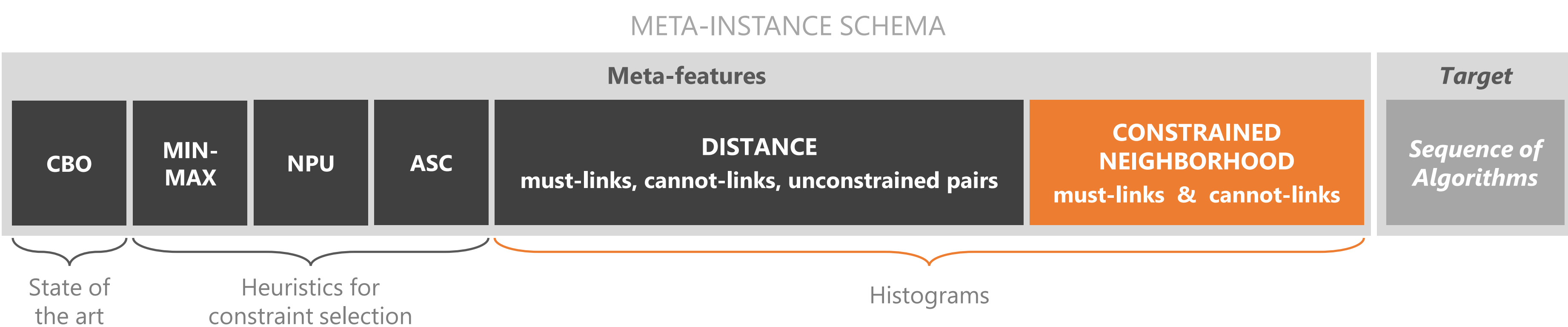


Adam, A., & Blockeel, H. (2017). Constraint-based measure for estimating overlap in clustering. In *Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning* (Vol. 6, pp. 54–61).

Proposed Approach

Meta-features Schema

- CBO
- Features computed from heuristics for selecting constraints
- Histograms built from distances between all pair of instances (constrained and unconstrained)
- Our proposed meta-feature: Constrained Neighborhood



Proposed Meta-feature

Main Idea

- Assumption:

  *a **well-spread set of constraints** can provide*
  ***holistic** information about the dataset*

- Histograms capture the most information possible about the problem being characterized [Kalousis 2002]

Algorithm

Algorithm CONSTRAINT NEIGHBORHOOD-BASED HISTOGRAM

INPUT:

$constraint\_set$: must-link set (or cannot-link set),

k: maximum number of neighbors

OUTPUT:

h: the histogram in which each bar represents the proportion of shared k-nearest instances

$\mathcal{E} = \{\},\ h = [0, \dots, 0]$

for each $constraint \in set\_of\_constraint$ **do**

  for $i \in [0, k]$ **do**

    for $x \in constraint$ **do**

      $\mathcal{N} = NearestNeighbors(x, i)$

      $h\,[i] = h[i] + \frac{|\mathcal{N} - \mathcal{E}|}{n}$

      $\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{N}$

    end_for

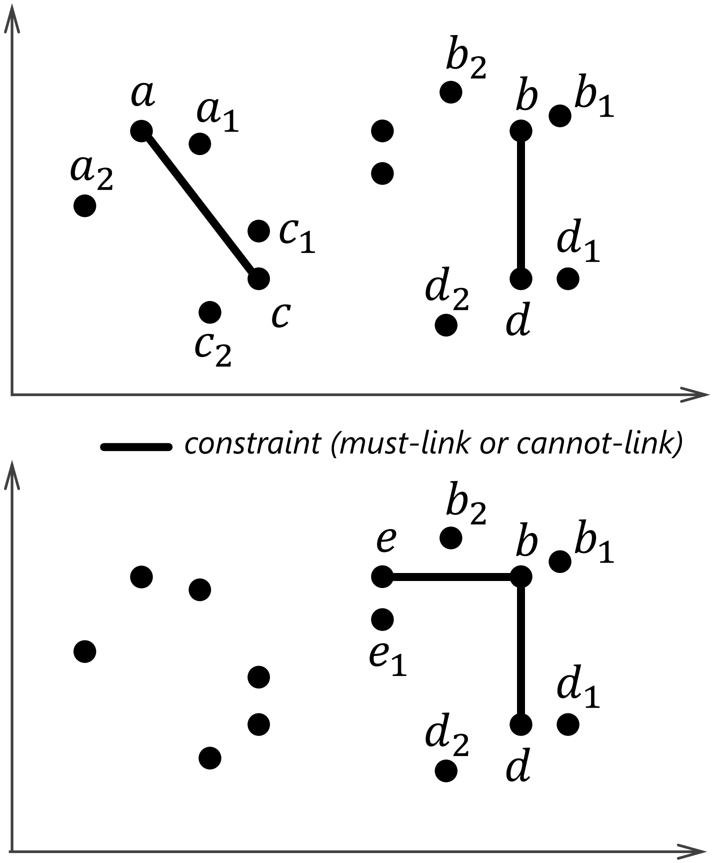
  end_for

end_for

return h

Examples

- Same datasets and number of constraints

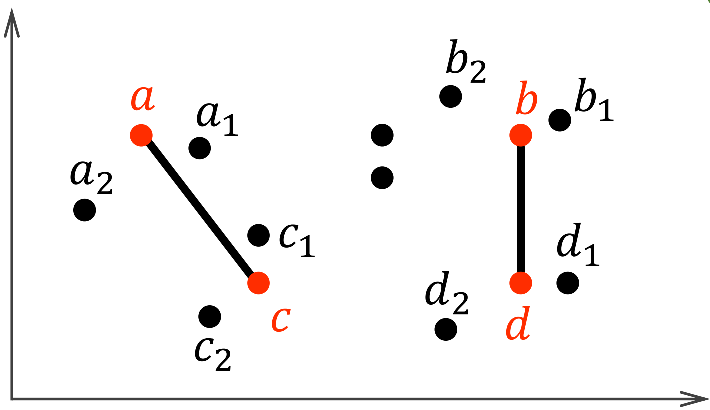


Proposed Meta-feature

Examples

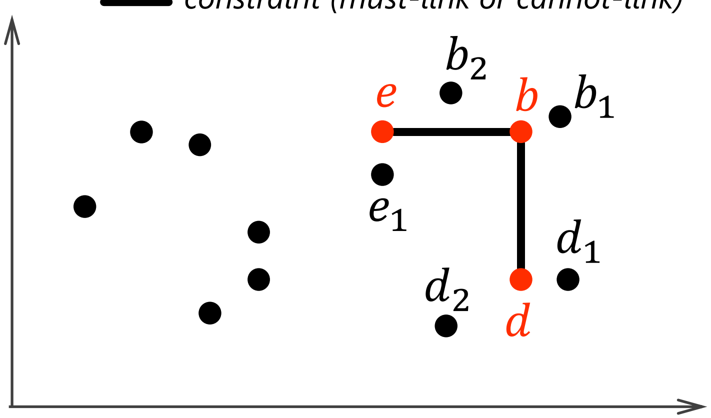
- Same datasets and number of constraints
- **Different** constraints organization → **Different** histograms

$k = 2$



$$h[0] = \frac{|\{a, b, c, d\}|}{14} \approx 0.3$$

— *constraint (must-link or cannot-link)*



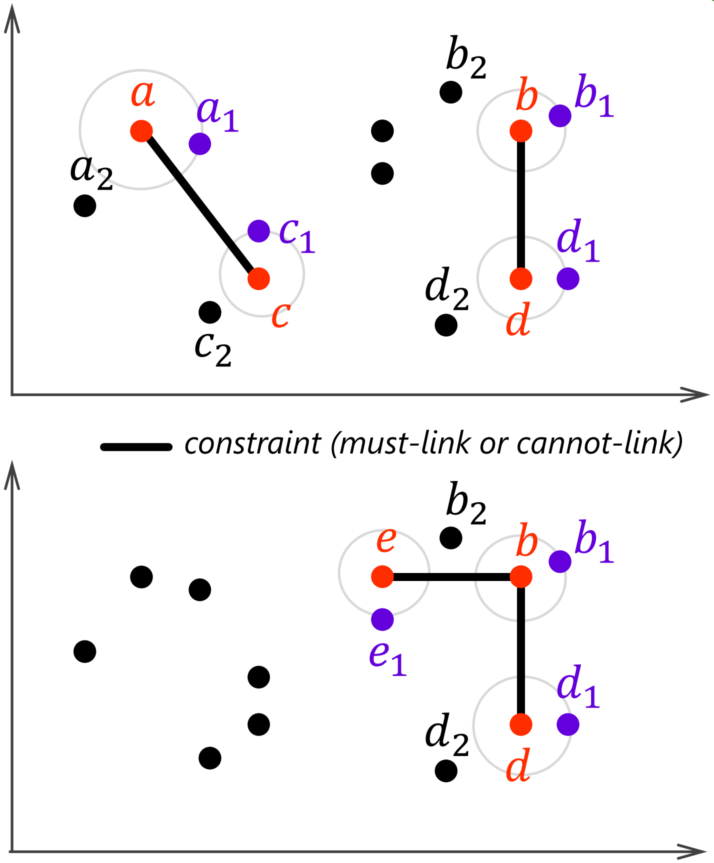
$$h[0] = \frac{|\{b, d, e\}|}{14} \approx 0.2$$

Proposed Meta-feature

Examples

- Same datasets and number of constraints
- **Different** constraints organization → **Different** histograms

$k = 2$



$$h[0] = \frac{|\{a, b, c, d\}|}{14} \approx 0.3$$

$$h[1] = \frac{|\{a_1, b_1, c_1, d_1\}|}{14} \approx 0.3$$

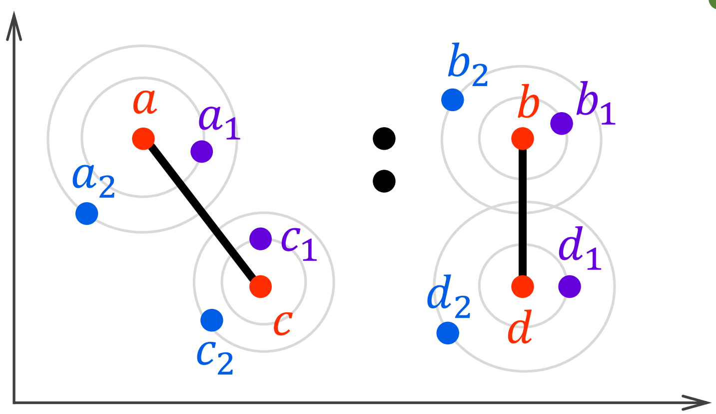
$$h[0] = \frac{|\{b, d, e\}|}{14} \approx 0.2$$

$$h[1] = \frac{|\{b_1, d_1, e_1\}|}{14} \approx 0.2$$

Examples

- Same datasets and number of constraints
- **Different** constraints organization → **Different** histograms

$k = 2$

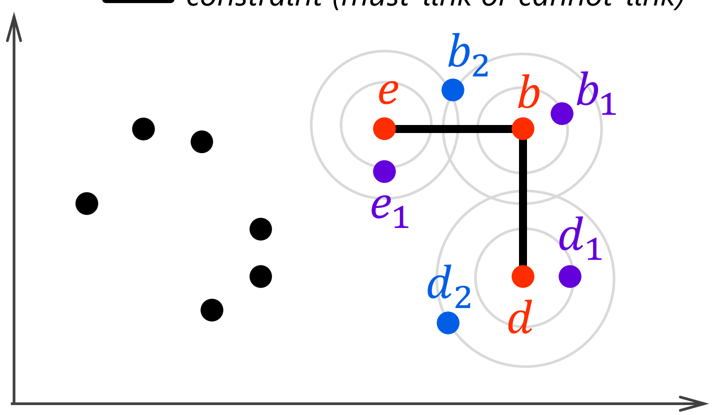


$$h[0] = \frac{|\{a, b, c, d\}|}{14} \approx 0.3$$

$$h[1] = \frac{|\{a_1, b_1, c_1, d_1\}|}{14} \approx 0.3$$

$$h[2] = \frac{|\{a_2, b_2, c_2, d_2\}|}{14} \approx 0.3$$

constraint (must-link or cannot-link)



$$h[0] = \frac{|\{b, d, e\}|}{14} \approx 0.2$$

$$h[1] = \frac{|\{b_1, d_1, e_1\}|}{14} \approx 0.2$$

$$h[2] = \frac{|\{b_2, d_2\}|}{14} \approx 0.1$$

Proposed Meta-feature

Examples

- Same datasets and number of constraints
- **Different** constraints organization → **Different** histograms

$$h = [0.3, 0.3, 0.3]$$

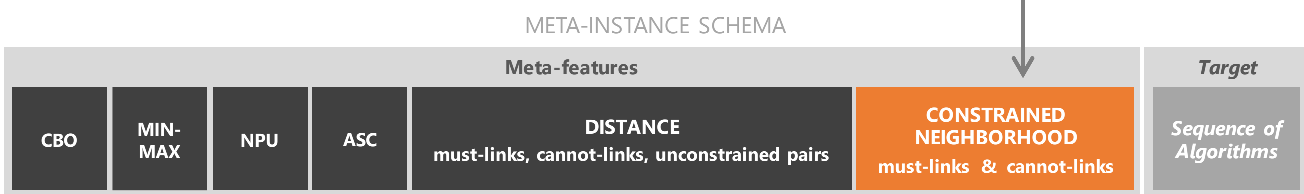
$$h = [0.2, 0.2, 0.1]$$

Proposed Meta-feature

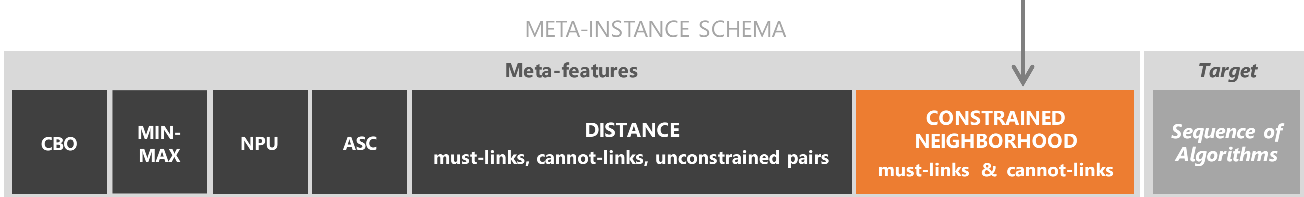
Examples

- Same datasets and number of constraints
- **Different** constraints organization → **Different** histograms

$$h = [0.3, 0.3, 0.3]$$



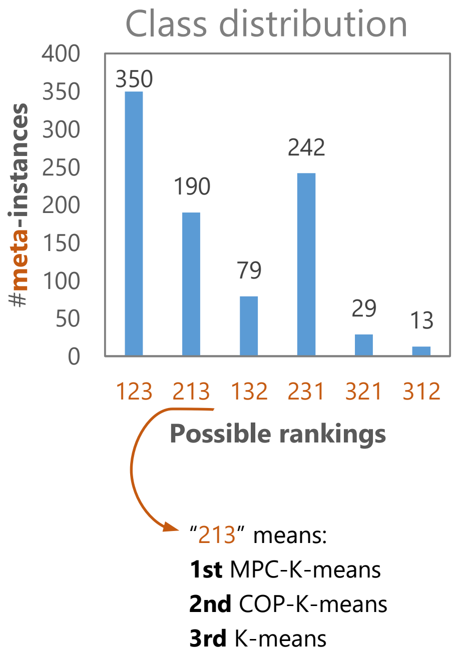
$$h = [0.2, 0.2, 0.1]$$



Experiments and Results

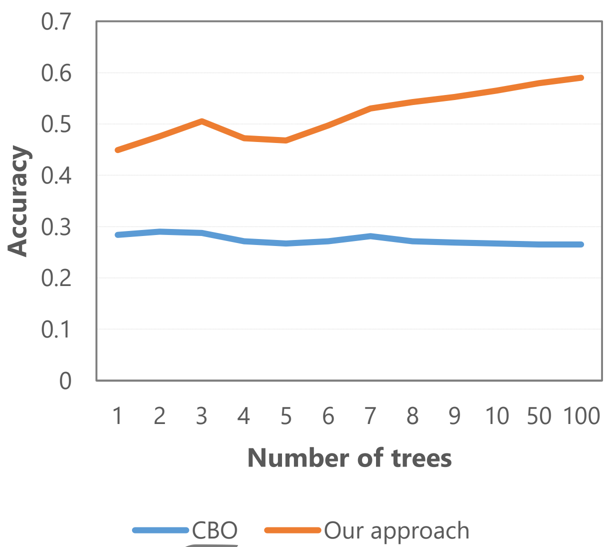
Experimental Setup

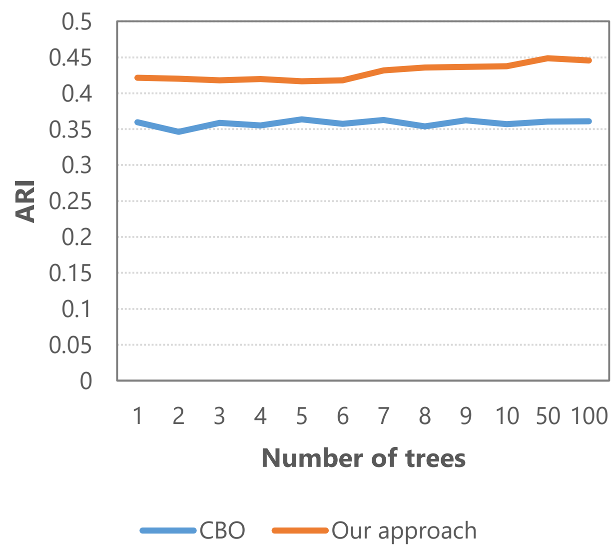
- **Datasets**: 23 (available on *openml.org*)
- **Constraints generation**
  - randomly selected (uniform distribution)
  - different sets for the same dataset
  - different number of constraints
    - 0%, 25%, 50% ,100% over the number of instances
- **Protocol**: leave-one-dataset-out
- **Clustering algorithms**:
  - Constrained: COP-K-means (**1**), MPC-K-means (**2**),
  - Unsupervised: K-means (**3**)



Experimental Results

Meta-learner: Random Forest





The state of the art

Conclusion

- The larger number of trees: the **main advantages of our approach over CBO**
  - we have more meta-features for describing the same clustering problems.
- ARI improvement
  - our meta-features contribute to a better decision of which clustering algorithm should be employed (**hypothesis**).

Future Directions

Future Work

- Incorporate more algorithms
- Select most informative meta-instances (training phase)
- Online learning

Thank you

Questions?

References

- Adam, A., & Blockeel, H. (2017). **Constraint-based measure for estimating overlap in clustering**. In *Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning* (Vol. 6, pp. 54–61). Retrieved from https://core.ac.uk/download/pdf/95683794.pdf
- Bilenko, M., Basu, S., & Mooney, R. J. (2004). **Integrating constraints and metric learning in semi-supervised clustering**. In *ICML* (p. 11). New York, New York, USA: ACM Press. https://doi.org/10.1145/1015330.1015360
- Brazdil, P., Carrier, C. G., Soares, C., & Vilalta, R. (2009). ***Metalearning Applications to Data Mining***. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-73263-1
- Breiman, L. (2001). **Random Forests**. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324
- Cachada, M. V., Abdulrahman, S. M., & Brazdil, P. (2017). **Combining feature and algorithm hyperparameter selection using some metalearning methods**. In *AutoML@PKDD/ECML* (Vol. 1998). Retrieved from https://repositorio.inesctec.pt/handle/123456789/7126
- Mallapragada, P. K., Jin, R., & Jain, A. K. (2008). **Active query selection for semi-supervised clustering**. In *ICPR* (pp. 1–4). IEEE. https://doi.org/10.1109/ICPR.2008.4761792
- Pimentel, B. A., & de Carvalho, A. C. P. L. F. (2019). **A new data characterization for selecting clustering algorithms using meta-learning**. *Information Sciences*, *477*, 203–219. https://doi.org/10.1016/J.INS.2018.10.043
- Ruiz, C., Spiliopoulou, M., & Menasalvas, E. (2007). **C-DBSCAN: Density-Based Clustering with Constraints**. *11th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing* (Vol. 4482). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72530-5
- Vu, V., Labroche, N., & Bouchon-Meunier, B. (2010). **Boosting Clustering by Active Constraint Selection**. In *ECAI*. Lisbon, Portugal.
- Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). **Constrained k-means clustering with background knowledge**. In *ICML* (Vol. 1, pp. 577-584).
- Wang, G., Song, Q., Zhang, X., & Zhang, K. (2014). **A generic multilabel learning-based classification algorithm recommendation method**. *ACM TKDD*, *9*(1), 7.
- Xiong, S., Azimi, J., & Fern, X. Z. (2014). **Active Learning of Constraints for Semi-Supervised Clustering**. *IEEE TKDE*, *26*(1), 43–54. https://doi.org/10.1109/TKDE.2013.22