A branch-and-price algorithm for the hyper-rectangular clustering problem with axis-parallel clusters and outliers

Diego Delle Donne[1], Javier Marenco[2]

[1] ESSEC Business School, France
delledonne@essec.edu
[2] Business School, Universidad Torcuato Di Tella, Argentina
javier.marenco@utdt.edu

Keywords : *clustering, hyper-rectangle, outliers, integer programming, branch-and-price*

1 Introduction

Given a set $\mathcal{X} = \{x^1, \ldots, x^n\}$ of n points in $\mathbb{R}^d$ and an integer $p \geq 1$, a *p-clustering* of $\mathcal{X}$ is a collection of subsets $C_1, \ldots, C_p \subseteq \mathcal{X}$, in such a way that $\mathcal{X} = C_1 \cup \cdots \cup C_p$ and $C_i \cap C_j = \emptyset$ for $i, j \in \{1, \ldots, p\}$, $i \neq j$. Each set from $\{C_i\}_{i=1}^p$ is called a *cluster* in this context. The *span* of a cluster $C \subseteq \mathcal{X}$ over the coordinate t is $\text{span}_t(C) = \max\{x_t : x \in C\} - \min\{x_t : x \in C\}$ if $C \neq \emptyset$ and $\text{span}_t(C) = 0$ otherwise, and the *total span* of C is $\text{span}(C) = \sum_{t=1}^d \text{span}_t(C)$. Finally, the *total span* of a clustering $\mathbb{C} = \{C_1, \ldots, C_p\}$ is defined as the sum of the total spans of its constituent clusters, i.e., $\text{span}(\mathbb{C}) = \sum_{i=1}^p \text{span}(C_i)$. Given a set $\mathcal{X}$ of points, the integer p specifying the number of clusters, and an integer $q \geq 0$, the *hyper-rectangular clustering problem with axis-parallel clusters and outliers* (HRC-APO) consists in determining a p-clustering of a subset of points $\mathcal{X}' \subseteq \mathcal{X}$ with cardinality $|\mathcal{X}'| \geq n - q$, minimizing the total span. In other words, we seek a clustering of $\mathcal{X}$ into p clusters with minimum total span, and we are allowed to discard up to q points, which are thereby declared to be *outliers*.

Hyper-rectangular clustering has been proposed as a model for *explainable clustering*, since it is straightforward to describe the obtained clusters by the bounds defining each hyper-rectangle. Indeed, if each coordinate corresponds to a relevant parameter in the application generating the given points, then clusters are specified by a lower and an upper bound on each parameter, and this is easier to communicate than a distance-based clustering [1].

The first integer programming approach for hyper-rectangular clustering was proposed in [4] for the case $q = 0$, and integer programming concepts also appear in [3] for $p = 2$. A branch-and-cut procedure for the general case was introduced in [2]. In this work we propose an extended formulation and a branch-and-price (B&P) procedure for it, and we show how this approach outperforms the ones from the literature.

2 The extended formulation and the branch and price

Let $\mathcal{C} \subset 2^{\mathcal{X}}$ denote the set of all nonempty subsets of $\mathcal{X}$. For $C \in \mathcal{C}$, we use the binary variable y_C representing whether the cluster C is included in the solution or not. For $i \in [n] := \{1, \ldots, n\}$, we use the binary variable w_i representing whether the point x^i is clustered or not, i.e., $w_i = 1$ if the solution includes at least one cluster containing x^i. In this setting, HRC-APO can be solved by minimizing $\sum_{C \in \mathcal{C}} \text{span}(C)\, y_C$, subject to the following constraints:

$$\sum_{C \in \mathcal{C}: i \in C} y_C \geq w_i \qquad \forall i \in [n] \tag{1}$$

$$\sum_{C \in \mathcal{C}} y_C \leq p \tag{2}$$

$$\sum_{i\in[n]} w_i \quad \geq \quad n - q \tag{3}$$

As the number of variables in the formulation grows exponentially with the number n of points, we solve the *linear relaxation* (LP) of this formulation by resorting to the *column generation* techniques. The reduced cost of variable y_C is $\text{span}(C) - \sum_{i\in C} \lambda_i + \mu$, where $\lambda_i \in \mathbb{R}$ and $\mu \in \mathbb{R}$ are the dual variable associated with Constraints (1) and (2), respectively. Given an optimal solution for the RMP, the *pricing problem* consists in determining whether there exists a cluster C with negative reduced cost or not. We propose two different MIP formulations to tackle this problem and we empirically show that one of these slightly outperforms the other.

In order to deal with fractional optimal solutions on the nodes of the branching tree, we propose two branching rules. The first one seeks to rule out solutions in which a point $x^i \in \mathcal{X}$ is partially considered to be an outlier, i.e., when $0 < w_i < 1$, for some $i \in [n]$.

Branching Rule 1 ($\mathbf{BR_1}$) *A given point $x \in \mathcal{X}$ is either an outlier or not.*

Unfortunately, rule BR_1 is not *complete*; even when every variable w_i takes an integer value, the solution can still have fractional values for some cluster variables y_C. To deal with this situation, we propose a second branching rule (BR_2).

Branching Rule 2 ($\mathbf{BR_2}$) *Given a point $x \in \mathcal{X}$, a coordinate axis $t \in [d]$, and a value $\delta \in \mathbb{R}$, any hyper-rectangle covering x must have lower (resp. upper) limit either lesser-or-equal or greater-or-equal than δ in the coordinate axis t.*

If a point $x \in \mathcal{X}$ is partially covered by two different clusters C_1 and C_2, then these clusters should differ in at least one bound (lower or upper) of one dimension t. In this case, we find a value δ separating C_1 and C_2 in this bound and we impose a branching rule to cut-off the fractional solution. Unfortunately, this branching modifies the pricing problem, however, our pricing MIP models can be easily modified in order to cope with the branching constraints.

3 Concluding remarks

We implemented the B&P in Java using Cplex as the linear programming solver[1]. We use the instance generator introduced in [2] for the experiments. Our experimentation evidences that our B&P procedure tends to outperform the branch-and-cut procedure from [2], particularly as the number of clusters p is increased, both in running times to optimality and achieved dual gaps when the time limit is reached. We also show that the lower bounds obtained by the continuous relaxation (LP) of our formulation tends to be very tight, in particular when the number of points n is increased.

References

[1] Aviruch Bhatia, Vishal Garg, Philip Haves, and Vikram Pudi. Explainable clustering using hyper-rectangles for building energy simulation data. *IOP Conference Series: Earth and Environmental Science*, 238(1):012068, feb 2019.

[2] Javier Marenco. An integer programming approach for the hyper-rectangular clustering problem with axis-parallel clusters and outliers. *Discrete Applied Mathematics*, 341:180–195, 2023.

[3] Sung Hee Park. Classification with axis-aligned rectangular boundaries. In Vijay K Mago and Nitin Bhatia, editors, *Cross-Disciplinary Applications of Artificial Intelligence and Pattern Recognition*, 2012.

[4] Sung Hee Park and Jae-Young Kim. Unsupervised clustering with axis-aligned rectangular regions. Technical report, Stanford University, 2009.

[1] Our source code is available at `https://github.com/jmarenco/clusterswithoutliers`.