Cluster Explanation via Polyhedral Descriptions

Connor Lawless [*1] **Oktay Günlük** [*1]

Abstract

This paper focuses on the cluster description problem where, given a dataset and its partition into clusters, the task is to explain the clusters. We introduce a new approach to explain clusters by constructing a polyhedron around each cluster while minimizing either the complexity of the resulting polyhedra or the number of features used in the description. We formulate the cluster description problem as an integer program and present a column generation approach to search over an exponential number of candidate half-spaces that can be used to build the polyhedra. To deal with large datasets, we introduce a novel grouping scheme that first forms smaller groups of data points and then builds the polyhedra around the grouped data, a strategy which out-performs the common approach of sub-sampling data. Compared to state of the art cluster description algorithms, our approach is able to achieve competitive interpretability with improved description accuracy.

1. Introduction

Machine learning (ML) is becoming an omnipresent aspect of the digital world. While ML systems are increasingly automating tasks such as image tagging or recommendations, there is increasing demand to use them as decision support tools in a number of settings such as criminal justice (Rudin & Ustun, 2018; Završnik, 2021; Berk, 2012), medicine (Rajkomar et al., 2019; Ustun & Rudin, 2016; Varol et al., 2017), and marketing (Ma & Sun, 2020; Hair Jr & Sarstedt, 2021; Dzyabura & Yoganarasimhan, 2018). Thus it is becoming increasingly critical that human users leveraging these ML tools understand and critique the outputs of the ML models to trust and act upon the recommendations. This is especially true for clustering, an unsupervised machine learning task, where a set of unlabelled data points are partitioned into groups (Xu & Wunsch, 2005). Clustering is often used in industry as a tool to find sub-populations in a dataset such as customer segments (Kansal et al., 2018), different media genres (Daudpota et al., 2019), or even patient subgroups in clinical studies (Wang et al., 2020). In these settings, practitioners often care less about the actual cluster assignments (i.e. which user is in which group) but rather a description of the groups found (i.e. a segment of users that consistently buy certain kinds of products). Unfortunately, many clustering algorithms only output cluster assignments, forcing users to work backwards to construct cluster descriptions.

This paper focuses on the cluster description problem, where a fixed clustering partition of a set of data points with real or integer coordinates is given and the goal is to find a compact description of the clusters. This problem occurs naturally in a number of settings where a clustering has already been performed either by a black-box system, or on unseen or complex data (for example a graph structure) and needs to be subsequently explained using features that may not have even been used in the initial cluster assignment.

In this paper we introduce a new method for cluster description that treats each data point as a vector in $\mathbb{R}^n$ and works by constructing a polyhedron around each cluster to act as its explanation, henceforth referred to as *polyhedral descriptions*. Each polyhedron is obtained by intersecting a (small) number of half spaces. We measure the interpretability of these polyhedra using two different notions: *complexity*, which is defined to be the number of half-spaces used plus the sum of the number of nonzero coefficients used to define each half-space, or *sparsity*, which is defined to be the number of features used across all half-spaces defining the polyhedra. If the convex hulls of the data points in each cluster do not intersect, then the half spaces defining the convex hull of the points in a cluster gives a polyhedral description for the cluster. However, such polyhedra may not have desirable interpretability characteristics as it might require a large number of half-spaces or involve many features. In this case a simpler explanation with some error might be more desirable. Furthermore, if the convex hulls of the clusters intersect then no error-free polyhedral description exists. Figures 1 and 2 show examples of the polyhedra associated with both cases.

In our setting, the accuracy of a cluster explanation is mea-

[*]Equal contribution [1]Operations Research and Information Engineering, Cornell University, USA. Correspondence to: Connor Lawless <cal379@cornell.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

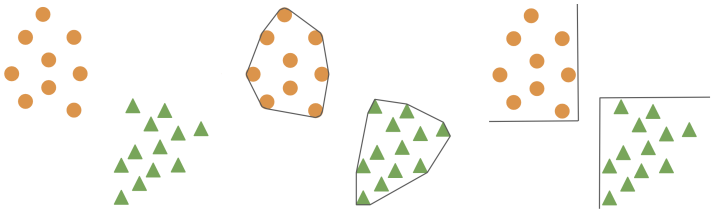

Figure 1. (Left) A sample set of clusters to be explained where the convex hulls do not intersect and perfect explanation is possible. (Middle) Polyhedral description using convex hull of clusters. (Right) Lower complexity polyhedral description.

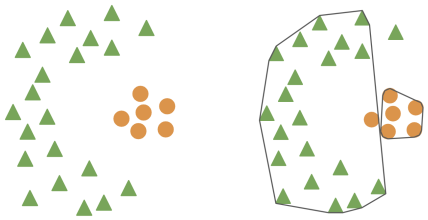

Figure 2. (Left) Sample set of clusters where convex hulls intersect and perfect explanation is impossible. (Right) Polyhedral description for clusters where convex hull intersects with best accuracy.

sured by the fraction of data points that are correctly explained (i.e. included in the polyhedron of their cluster and not included in other polyhedra). This framework allows the cluster explanation to trade-off accuracy with interpretability. Moreover, we can explicitly optimize the interpretability of the final polyhedral descriptions with respect to both complexity and sparsity. While a polyhedron may not initially seem like an interpretable model class, additional constraints placed on the half-spaces that construct the polyhedron allows cluster descriptions from popular interpretable model classes such as rule sets (Lawless et al., 2021; Wang et al., 2017; Rudin & Ertekin, 2018) and score cards (Ustun & Rudin, 2017).

1.1. Related Work

Existing work in interpretable clustering can be broadly divided into two groups: *cluster description*, where cluster assignments are given and the task is to explain them (our work builds on this line of research); or *interpretable clustering* approaches, where cluster assignments are generated using an interpretable model class.

A common approach for cluster description is to simply use a supervised learning algorithm to predict the cluster label assignments that are already given (Jain et al., 1999; De Koninck et al., 2017; Kauffmann et al., 2019). Broadly this can be seen as the application of multi-class classification (Aly, 2005) to cluster description. However, multi-class classification and cluster description differ in a few important characteristics. In multi-class classification the objective is to maximize classification accuracy, whereas the aim of cluster description is to explain the given clusters as simply as possible. In other words, cluster description aims to optimize interpretability with constraints on accuracy. Multi-class classification models are also expected to perform inference (i.e. make a prediction on new data). In cluster description there is no guarantee that the explanation is a partition of the feature space, and thus new data points can possibly fall outside all existing cluster descriptions.

In a recent work Carrizosa et al. (2022) introduce an integer programming (IP) framework for selecting a single prototype data point from each cluster and build a ball around it to act as a description for the cluster. While selecting a prototype point has an intuitive appeal, the resulting explanation can be misleading or uninformative if clusters are not compact or isotropic (i.e. have equal variance in all directions). Davidson et al. (2018) introduce a version of the cluster description problem where each data point has an associated set of tags coming from a discrete set. The goal of their formulation is to find a disjoint set of tags for each cluster such that each data point in a cluster is covered by at least one tag assigned to that cluster, which they call the disjoint-tag descriptor minimization problem (DTDM). If we interpret each half-space in a polyhedral description as a tag, our approach bares a superficial resemblance to the DTDM problem but it also differs in a number of ways. First, a data point satisfies a description in the polyhedral description setting only if it satisfies all the conditions in the description, whereas in the DTDM a data point only needs to satisfy one of the tags used to describe the cluster. Unlike the DTDM, our framework does not require data be provided with discrete tags and allows for real valued features. Finally, a data point is not considered correctly described in the polyhedral description problem if it meets a description for another cluster, a constraint not included in the DTDM. We note that this constraint ensures that cluster descriptions are informative (i.e. describe only a single cluster).

There has also been extensive work on constructing clusters using interpretable model classes such as decision trees with uni-variate splits (Bertsimas et al., 2021; Fraiman et al., 2013; Liu et al., 2000; Moshkovitz et al., 2020; Frost et al., 2020; Dasgupta et al., 2020), or rule sets (Chen et al., 2016; Chen, 2018; Pelleg & Moore, 2001; Carrizosa et al., 2022). The important distinction between this line of work and our setting is that this line of research assumes that the cluster assignment is not fixed. Most similar to our work is the use of multi-polytope machines to perform the clustering (Lawless et al., 2021). However, our approach differs from this line of work as the cluster assignments are fixed in the cluster description problem, and the aim is optimize

interpretability not the quality of the clustering itself. The cluster description setting can also be modeled as an IP as opposed to a mixed-integer non-linear program (MINLP) which allows our approach to scale to larger datasets.

1.2. Main Contributions

We summarize our main contributions as follows:

- We introduce the polyhedral description problem which aims to explain why the data points in the same cluster are grouped together by building a polyhedron around them. We also show that this is an NP-Hard problem.
- We formulate the polyhedral description problem as an (exponential size) integer program where variables correspond to candidate half-spaces that can be used in the polyhedral description of the clusters. We present a column-generation algorithm to search over the (exponentially many) candidate half-spaces efficiently.
- We introduce a novel grouping scheme to summarize input data. This approach helps reduce the size of the IP instances and enables us to handle large datasets. We also present empirical results that show that our grouping scheme out-performs the commonly used approach of sub-sampling data points.
- We present numerical experiments on a number of real world clustering datasets and show that our approach performs favorably compared to state-of-the-art cluster description approaches.

The remainder of the paper is organized as follows. Section 2 formalizes the polyhedral description problem and presents an exponential sized IP formulation for constructing optimal polyhedral descriptions together with a column generation approach for solving it. Section 3 introduces a novel grouping scheme to enable the IP approach to deal with large scale data. Finally, Section 4 presents numerical results on a suite of UCI clustering data sets.

2. Problem Formulation

We now formally introduce the Polyhedral Description Problem (PDP). The input data for the problem consists of a set of n data points with m real-valued features $\mathcal{X} = \{x^i \in \mathbb{R}^m\}_{i=1}^n$, a partition of the data points in $\mathcal{X}$ into K clusters $C_1, \ldots, C_K$, where each C_k denotes the set of data points belonging to cluster k, and a set of candidate half-spaces $\mathcal{H}$ from which we can construct polyhedra. Assuming the data points to be real-valued is not a restrictive assumption in practice, as categorical data can be converted to real-valued features via a one-hot encoding scheme. Let x_d^i be the d-th feature of the data point x^i and k_i be its cluster assignment. For a given $w \in \mathbb{R}^m$ and $b \in \mathbb{R}$, the half-space associated with (w, b) is the set $h = \{x \in \mathbb{R}^m : w^T x \leq b\}$. For the remainder of the paper we refer to the half-space and the hyperplane defining a half-space interchangeably (i.e. refer to $\|w\|_0$ as the number of features used in a half-space). A polyhedron is the intersection of a finite number of half-spaces (Boyd et al., 2004).

A perfect solution to the PDP is a set of polyhedra $\{P_k\}_{k=1}^K$ such that $x \in C_k$ for all $x \in P_k$ and $x \notin C_k$ for all $x \notin P_k$. Note that when the convex hulls of the clusters intersect no such solution exists. In our formulation, we aim to find *good* solutions and allow the polyhedra to explain up to α data points incorrectly:

$$\left| \left\{ x^i \in \mathcal{X} : x^i \notin P_{k_i} \vee x^i \in \cup_{k' \neq k_i} P_{k'} \right\} \right| \leq \alpha$$

We say that a data point $x \in \mathcal{X}$ is correctly explained if $x \in P_{k_i}$ and $x \notin \cup_{k' \neq k_i} P_{k'}$. To improve the interpretability of the resulting descriptions we consider a restricted set of candidate half-spaces $\mathcal{H}$ that are defined by sparse hyperplanes with small integer coefficients. More precisely, we consider half-spaces that have the form $\{x \in \mathbb{R}^m : w^T x \leq b\}$ for integral w with maximum value W, $\max_d |w_d| \leq W$, and at most β non-zero values, $\|w\|_0 \leq \beta$. Note that these restrictions on the set of candidate half-spaces may cause the PDP to be infeasible, even if the convex hulls of the points in each cluster do not intersect.

It is important to note that this approach does not require the polyhedra to be non-intersecting, but rather penalizes data points that fall into multiple polyhedra. From a practical perspective, adding such a restriction on the polyhedra would lead to a computationally challenging problem. It may also be overly restrictive in settings where the intersection of polyhedra is unlikely to contain any data (see Appendix A for an illustrative example). In our computational experiments we observed only a small number of data points in the intersection of multiple polyhedra while there were many examples of polyhedra intersecting. We consider two variations of the PDP that add additional restrictions on the polyhedral descriptions to help improve interpretability.

Low-Complexity PDP (LC-PDP): This variant restricts the complexity of the polyhedral description. Similar to previous work on rule sets (Lawless et al., 2021), we define complexity of a half-space as the number of non-zero terms in the half-space plus one, and the complexity of the polyhedron as the sum of the complexities of the half-spaces that compose it.

***Sparse* PDP (Sp-PDP):** The second variant we consider puts a limit on the total number of features in all the half-

spaces used in the polyhedral descriptions (i.e. sparsity). Unfortunately, both variants of the PDP are strongly NP-Hard (see Appendix B for proof).

Theorem 2.1. *Both the Low Complexity and Sparse Polyhedral Description Problems are strongly NP-Hard.*

2.1. Integer Programming Formulation for the PDP

Given a set of candidate half-spaces $\mathcal{H}$ that can be used in a polyhedral description, we next formulate the optimization version of both variants of the PDP as an integer program. In practice, enumerating all possible candidate half-spaces, even in this restricted setting, is computationally impractical and we describe a column generation approach to handle this in the subsequent section. Let $\mathcal{H}_i = \{(w, b) \in \mathcal{H} : w^T x^i > b\}$ be the set of half-spaces that data point i falls outside, and in a slight abuse of notation let $\mathcal{H}_d = \{(w, b) \in \mathcal{H} : w_d \neq 0\}$ be the set of half-spaces that use feature d. The complexity of a half-space $h = (w, b)$ is defined to be $c_h = \|w\|_0 + 1$.

Let z_{hk} be the binary decision variable indicating whether half-space h is used in the polyhedral description of cluster $k \in \mathcal{K} = \{1, \dots, K\}$. Note that we can recover the polyhedral description for cluster k from these binary variables as $P_k = \bigcap_{h \in I_k} h$ where $I_k = \{h \in \mathcal{H} : z_{hk} = 1\}$. We use a binary variable ξ_i to indicate whether data point i is mis-classified (i.e. either not included in its cluster's polyhedron or is incorrectly included in another cluster's polyhedron). Let y_d be a binary variable indicating whether feature $d \in \mathcal{D} = \{1, \dots, m\}$ is used in any of the half-spaces chosen for the polyhedral descriptions. With these definitions, an IP formulation for the PDP is as follows:

$$\textbf{min} \quad \theta_1 \sum_{k \in \mathcal{K}} \sum_{h \in \mathcal{H}} c_h z_{hk} + \theta_2 \sum_{d \in \mathcal{D}} y_d \tag{1}$$

$$\textbf{s.t.} \quad \xi_i + \sum_{h \in \mathcal{H}_i} z_{hk} \geq 1 \qquad \forall x^i \in \mathcal{X}, \forall k \neq k_i \tag{2}$$

$$M\xi_i - \sum_{h \in \mathcal{H}_i} z_{hk_i} \geq 0 \qquad \forall x^i \in \mathcal{X} \tag{3}$$

$$\sum_{k \in \mathcal{K}} \sum_{h \in \mathcal{H}_d} z_{hk} \leq M y_d \qquad \forall d \in \mathcal{D} \tag{4}$$

$$\sum_{x^i \in \mathcal{X}} \xi_i \leq \alpha \tag{5}$$

$$\xi_i, z_{hk}, y_d \in \{0, 1\} \tag{6}$$

where M is a suitably large constant. A natural choice is the smallest upper bound for the total number of half-spaces used (if an existing heuristic solution exists), or simply $|\mathcal{H}|$. Note that in practice the choice of M can be chosen independently for constraints (3) and (4). The objective consists of two terms that capture both variants of the PDP. The first term captures the complexity of the half-spaces used (LC-PDP), and the second captures the sparsity (Sp-PDP). θ_1 and θ_2 control the relative importance of each term. Note that if $\theta_1 = 1, \theta_2 = 0$ we get the LC-PDP, and similarly if $\theta_1 = 0, \theta_2 = 1$ we get the Sp-PDP.

Constraint (2) tracks false positives (i.e. data points that are included in a wrong cluster's polyhedron) and constraint (3) tracks false negatives (i.e. data points that are not included in their respective cluster's polyhedron). Constraint (4) tracks which features are used in the polyhedral descriptions. If $\theta_2 = 0$ (i.e. sparsity is not a consideration) then constraint (4) can be removed and the problem can be decomposed into a separate problem for each cluster. Constraint (5) sets an upper bound α on the number of data points that are not properly explained. We denote the problem (1)-(6) as the master integer program (MIP), and its associated linear relaxation, taken by relaxing constraint (6) to allow for non-integer values, as the master LP (MLP).

2.2. Column Generation

Enumerating every possible half-space is computationally intractable and thus it is not practical to solve the MIP using standard branch-and-bound techniques (Land & Doig, 1960). Instead, we use column generation (Gilmore & Gomory, 1961) to solve the MLP by searching over the best possible candidate half-spaces to consider in the master problem. Once we solve the MLP to (near) optimality or exceed a computational budget, we then use the set of candidate half-spaces generated during column generation to find a solution to the MIP. To solve the MLP we start with a restricted initial set of half-spaces $\hat{\mathcal{H}} \subset \mathcal{H}$. We denote the MLP solved using only $\hat{\mathcal{H}}$ the restricted master linear program (RMLP). In other words, the RMLP is the MLP where all variables corresponding to $\mathcal{H} \setminus \hat{\mathcal{H}}$ are set to 0. Once this small instance of the MLP is solved, we use the optimal *dual* solutions to the problem to identify a missing variable (i.e. half-space) that has a negative reduced cost. The problem to find such a half-space is called the *pricing problem* and can be solved by another integer program. If a new half-space with a negative reduced cost is found then we add it to the set $\hat{\mathcal{H}}$ and this process is repeated until either no such half-space can be found, which represents a certificate of optimality for the MLP, or a given computational budget is exceeded.

Let (μ, γ, ϕ) be the optimal dual solution to the RMLP where $\mu_{ik} \geq 0$ is the dual value corresponding to constraint (2) for data point i and cluster k, $\gamma_i \geq 0$ is the dual value corresponding to constraint (3) for data point i, and $\phi_d \leq 0$ is the dual value corresponding to constraint (4) for feature d, respectively. Since the decision variables z_{hk} in the MIP are defined for a half-space and a specific cluster k, we

define a separate pricing problem for each cluster, which can be solved in parallel. Using the optimal dual solution, the reduced cost $\rho_{(h,k)}$ for a missing variable z_{hk} corresponding to a half-space $h \notin \hat{\mathcal{H}}$ for a cluster k is:

$$\rho_{(h,k)} = \theta_1 c_h - \sum_{x^i \in \mathcal{X} \setminus C_k} \mu_{ik} \mathbb{1}(w^T x_i > b) + \sum_{x^i \in C_k} \gamma_i \mathbb{1}(w^T x_i > b) - \sum_{d \in \mathcal{D}} \phi_d \mathbb{1}(w_d \neq 0)$$

Where $\mathbb{1}(x)$ is the indicator function and equals 1 if the literal x is true, and 0 otherwise. Note that the reduced cost is non-negative for all half-spaces already included in the restricted master problem by the optimality of the dual solution (i.e. $\rho_{(h,k)} \geq 0 \ \ \forall h \in \hat{\mathcal{H}}$). For a given cluster k let $w \in \mathbb{Z}^m$ and $b \in \mathbb{R}$ be the decision variables representing the hyperplane used to construct a candidate half-space. We also introduce variables $w^+, w^- \in \mathbb{Z}_{\geq 0}$ that represent the positive and negative components of the hyperplane (i.e. $w_d^+ = \max(0, w_d)$ and $w_d^- = \max(0, -w_d)$). Let y_d be the binary variable indicating whether feature d is used in the hyperplane, and similarly y_d^+, y_d^- represent whether a positive or negative component of feature d is used. Finally let δ_i be the binary variable indicating whether data point $x^i \in \mathcal{X}$ is correctly included, for data points in C_k, or excluded, for data points in $\mathcal{X} \setminus C_k$, in the half-space. With these decision variables in mind, the pricing problem to find a candidate half-space for cluster k can be formulated as follows:

$$\textbf{min} \ \ \theta_1 (\sum_{d \in \mathcal{D}} (y_d^+ + y_d^-) + 1) - \sum_{x^i \in \mathcal{X} \setminus C_k} \mu_{ik}(1 - \delta_i) + \sum_{x^i \in C_k} \gamma_i \delta_i - \sum_{d \in \mathcal{D}} \phi_d (y_d^+ + y_d^-) \tag{7}$$

s.t.

$$(w^+ - w^-)^T x^i - b \leq M \delta_i \qquad \forall x^i \in C_k \tag{8}$$

$$(w^+ - w^-)^T x^i - b \geq \epsilon - M \delta_i \qquad \forall x^i \in \mathcal{X} \setminus C_k \tag{9}$$

$$y_d^+ \leq w_d^+ \leq W y_d^+ \qquad \forall d \in \mathcal{D} \tag{10}$$

$$y_d^- \leq w_d^- \leq W y_d^- \qquad \forall d \in \mathcal{D} \tag{11}$$

$$\sum_{d \in \mathcal{D}} (y_d^+ + y_d^-) \leq \beta \tag{12}$$

$$y_d^+ + y_d^- \leq 1 \qquad \forall d \in \mathcal{D} \tag{13}$$

$$\sum_{d \in \mathcal{D}} (w_d^+ + w_d^-) \geq 1 \tag{14}$$

$$w_d^+, w_d^- \in \mathbb{Z}_{\geq 0} \qquad \forall d \in \mathcal{D} \tag{15}$$

$$y_d, \delta_i \in \{0, 1\} \qquad \forall d \in \mathcal{D}, \ x^i \in \mathcal{X} \tag{16}$$

The objective of the problem is to minimize the reduced cost of the new column. Note that c_h is defined by $\|w\|_0 + 1$ which can be represented by the y_d variables in the objective. Constraint (8) tracks whether a data point in C_k is included in the half-space and similarly Constraint (9) tracks whether or not each data point outside of C_k is not included in the half-space. M is a suitably large constant that can be computed based on the data set and settings for W, β. In the latter constraint ϵ is a small constant to ensure the constraint is a strict inequality. Constraints (10) and (11) put a bound on the maximum integer coefficient size of the hyperplane, and constraint (12) puts a bound on the ℓ_0 norm of the hyperplane. Finally, constraints (13) and (14) exist to exclude the trivial solution where $w = 0$.

3. Grouped Data for Scalability

For problems with a large number of data points it can be computationally challenging to solve the IP formulation introduced in the preceding section. A standard approach for clustering or cluster description for large datasets is to simply sub-sample data points to consider in the optimization problem (see Carrizosa et al. (2022) for an example of the approach). While this approach has intuitive appeal, it fails to leverage all the information present in the given problem. Instead, we use a novel technique where we create smaller groups of data points that we treat as a single entity and perform the cluster description on the grouped data. This approach also effectively reduces the size of the problem instance without discarding any data points.

3.1. Description Error in Grouped Data

In this section we formalize the notion of grouping data points and present results on its impact on the accuracy of the resulting cluster description. We start by partitioning each cluster C_k into a set of smaller groups $\mathcal{G}_k$ where each data point is assigned to a single group and $C_k = \cup_{G \in \mathcal{G}_k} G$, and define $\mathcal{G} = \cup_{k=1}^K \mathcal{G}_k$. We say that a group $G \in \mathcal{G}$ is correctly explained if all data points $x \in G$ are correctly explained. Let $\mathbf{P} = \{P_k\}_{k=1}^K$ be a solution to the PDP (i.e. a set of polyhedral descriptions). We define the *true* cost $COST(\mathbf{P}) = \sum_{k=1}^K \sum_{x \in C_k} \mathbb{1}((x \notin P_k) \vee (x \in \bigcup_{k' \neq k} P_{k'}))$ to be the number of data points incorrectly explained by the solution. For simplicity we exclude the explicit dependence of the dataset $\mathcal{X}$ and the cluster assignments $\mathcal{C}$ from the inputs to the cost function. The scheme by which the groups are constructed can be viewed as a separate clustering task that can be performed by a clustering algorithm. In practice, compared to both k-means and DBSCAN we found that using a hierarchical clustering algorithm with a bound on the maximal linkage of each group performed the best empirically in preliminary experiments.

We define the grouped cost $COST_G(\mathbf{P}) =$

$\sum_{k=1}^{K} \sum_{G \in \mathcal{G}_k} |G| \mathbb{1}(\exists x \in G \text{ s.t. } (x \notin P_k) \vee (x \in \bigcup_{k' \neq k} P_{k'})))$, as the mis-classification cost of each group weighted by the size of the group. A natural corollary of this definition is that for any solution $\mathbf{P}$ the grouped cost *over-estimates* the true cost (i.e. $COST_G(\mathbf{P}) \geq COST(\mathbf{P})$). Let $\mathbf{P}^*_G = \text{argmin}_{\mathbf{P}} COST_G(\mathbf{P})$ and $\mathbf{P}^* = \text{argmin}_{\mathbf{P}} COST(\mathbf{P})$ be the optimal solutions to the grouped problem and original problem respectively. We now show that solving the PDP over groups versus the individual data points leads to mis-classifying at most $|G_{max}|$ times the optimal number of data points, where $|G_{max}|$ is the size of the largest group (see Appendix C for proof).

Theorem 3.1. *The optimal solution to the grouped problem, with **any grouping scheme**, incurs a cost no more than $|G_{max}|$ times the cost of the optimal solution to the full problem instance. Formally:*

$$COST(\boldsymbol{P}^*_G) \leq |G_{max}| COST(\boldsymbol{P}^*)$$

While $|G_{max}|$ may seem like a relatively large factor, it is important to note that even creating small groups can have large impacts on the size of problem instances that can be solved via integer programming (i.e. even groups of size 2 reduces the size of the IP formulation significantly). Note that Theorem 3.1 places no assumption on how the groups were formed (i.e. the grouping scheme), and thus provides a general bound for any grouping approach. A natural question is whether placing additional restrictions on how groups are formed can lead to a stronger guarantee. One such possible restriction is to ensure that the grouping is optimal with respect to a clustering evaluation metric. Silhouette coefficient is a popular clustering evaluation metric that has been used in a line of recent work on optimal interpretable cluster (Lawless et al., 2021; Bertsimas et al., 2021).

Unfortunately, the following result shows that the bound in Theorem 3.1 is tight in the sense that there exists an instance where the grouped cost is equal to $|G_{max}|$ times the optimal cost on the full problem even when a large number of groups are used via an optimal grouping scheme with respect to the silhouette coefficient (see Appendix D for proof).

Theorem 3.2. *Even for $|\mathcal{G}_k| = |C_k| - 2 \ \forall k \in \mathcal{K}$, $K = 2$, and an optimal grouping scheme with respect to silhouette coefficient, there exists an instance where:*

$$COST(\boldsymbol{P}^*_G) = |G_{max}| COST(\boldsymbol{P}^*)$$

Note that although this theorem uses silhoeutte coefficient, we believe that the bound is also tight for any other cluster evaluation metric. The emphasis of this result is that even when groups are constructed in a reasonable manner, there still exists an instance where the upper bound is tight. We also note that these are worst-case bounds and in practice grouping performs much better. Unlike accuracy, grouping data points can have ambiguous affects on the interpretability of the final solution (i.e. can lead to solutions that are simpler or more complex).

3.2. Integer Programming Formulation with Grouping

We next describe how to integrate the grouped data into the original IP formulation presented in Section 2.1. The goal of the approach is to summarize the information about each group in such a way that the resulting integer program scales linearly with the number of groups. For this purpose we start with constructing the smallest hyper-rectangle that contains all the data points in each group. Let $x^H_{G,d} = max_{x \in G} x_d$ and $x^L_{G,d} = min_{x \in G} x_d$ be the maximum and minimum value for coordinate d for the points in group G. The hyper-rectangle R_G for the group G is defined as the set $R_G = \left\{x \in \mathbb{R}^m : x^H_{G,d} \geq x_d \geq x^L_{G,d} \ \forall d = 1, \ldots, m\right\}$. In our new formulation we consider a group to be mis-classified if any part of the hyper-rectangle is mis-classified. Note that this is a stronger condition than the previous section where a group is mis-classified if any data point is mis-classified. However, modelling the pricing problem to track whether each individual data point is correctly classified would not reduce the problem size of the pricing problem, eliminating the computational benefit of leveraging grouping. It is also worth noting this difference only occurs for non-axis parallel half-spaces (i.e. $\beta > 1$).

Let w^+ and w^- again represent the positive and negative components of the hyperplane (i.e. $w^+_d = \max(w_d, 0)$, $w^-_d = \max(-w_d, 0)$). A hyper-rectangle for group G is fully inside a half-space $h = (w, b)$ (i.e. $R_G \subset h$) if $(w^+)^T(x^H_G) - (w^-)^T(x^L_G) \leq b$. Similarly, a hyper-rectangle for a group G is fully outside a half-space (i.e. $R_G \cap h = \emptyset$) if $(w^+)^T(x^L_G) - (w^-)^T(x^H_G) > b$. Note these conditions are akin to ensuring the worst-case corner of the hyper-rectangle is within a given half-space.

We can now integrate the hyper-rectangle approach into the IP formulation as follows. In the master problem, let $\mathcal{H}^+_G$ and $\mathcal{H}^-_G$ represent the set of half-spaces that group G does not fully fall within or fall outside respectively. Formally $\mathcal{H}^+_G = \{h \in \mathcal{H} : (w^+)^T(x^H_G) - (w^-)^T(x^L_G) > b\}$ and $\mathcal{H}^-_G = \{h \in \mathcal{H} : (w^+)^T(x^L_G) - (w^-)^T(x^H_G) > b\}$. Constraints (2), (3), and (5) in the MLP/MIP are thus updated to the following:

$$\xi_G + \sum_{h \in \mathcal{H}^-_G} z_{hk} \geq 1 \qquad \forall k \neq k_G, \ \forall G \in \mathcal{G} \tag{17}$$

$$M\xi_G - \sum_{h \in \mathcal{H}^+_G} z_{hk} \geq 0 \qquad \forall k = k_G, \ \forall G \in \mathcal{G} \tag{18}$$

$$\sum_{i \in \mathcal{G}} |G_i| \xi_i \leq \alpha \tag{19}$$

where k_G is the cluster of group G. Note that constraints (17) and (18) are nearly identical to the non-grouped version except the sets of hyperplanes are now defined for hyper-rectangles. Constraint (19) now weights the error of the solution by the size of the group.

To alter the pricing problem for the grouped setting we update the constraints that check whether or not a data point is correctly included in the half-space to check the entire hyper-rectangle. Specifically we update constraints (8) and (9) to the following:

$$(w^+)^T(x_G^H) - (w^-)^T(x_G^L) - b \leq M\delta_i \qquad \forall G \in \mathcal{G}_k \tag{20}$$

$$(w^+)^T(x_G^L) - (w^-)^T(x_G^H) - b \geq \epsilon - M\delta_i \quad \forall G \in \mathcal{G} \setminus \mathcal{G}_k \tag{21}$$

3.3. Empirical Evaluation

To evaluate the performance of our grouped data approach versus sub-sampling data points we ran a sequence of experiments on synthetic data. Data was generated using a Gaussian mixture model where cluster centers were sampled uniformly from $[-1, 1]^m$, and n data points were generated around the sampled center for each cluster with a covariance matrix of σI where I is the $m \times m$ identity matrix. The parameter σ controls the difficulty of the description problem as larger values of σ lead to clusters with considerable overlap making a perfect explanation unlikely. To construct the groups for our approach we use hierarchical clustering with a limit on the maximal linkage distance χ, which is akin to setting a maximum diameter on the size of the groups. We tested a range of different χ values to get different number of groups. To provide a fair comparison between the two approaches we sub-sampled the same number of data points (uniformly at random) as the number of groups. The same set of candidate half-spaces, generated by considering all possible uni-variate splits, is also used for both approaches. For all of the following results we created 50 random instances using the above simulation procedure with $K = 3$, $m = 10$, and $n = 10000$ and then ran both approaches and averaged the performance over the 50 instances, and 5 random sub-samples. Figure 3 shows the results of the synthetic experiments. The results show that for an equivalent number of samples (i.e. groups for the grouped data and data points for the sub-sampled data) the grouping approach is able to find explanations with a lower error rate.

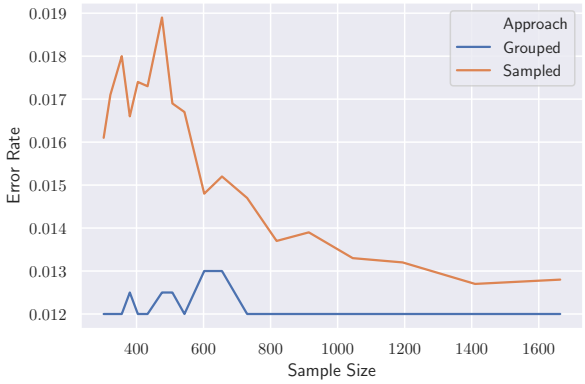

Figure 3. Relative performance of grouping data versus sub-sampling. Error rate is what percentage of dataset is not properly explained by the explanation generated using each approach. Sample size is the number of groups or data points for the grouped and sub-sampled approach respectively.

4. Numerical Results

To evaluate our approach we ran experiments on a suite of clustering datasets from the UCI Machine Learning repository (Asuncion & Newman, 2007). Details on how the data was pre-processed and other implementation details can be found in the Appendix. Note that for certain choices of α the MIP and MLP may be infeasible. As α is not given as a constraint for the application a priori in these datasets, we use a two-stage procedure to first find a feasible α then optimize for interpretability. In the first phase we replace the objective in the MIP (1) with α which we take as a continuous decision variable and solve via column generation. The goal of the first stage is thus to optimize for the accuracy of the descriptions. We then take the optimal α^* from the first stage and multiply it by a tolerance factor (i.e. $(1 + \kappa)\alpha^*$ for a small κ) and use it in constraint (5) in the second stage to optimize for interpretability of the descriptions. For the following experiments we used $\kappa = 0.05$. We benchmark our approach against three common algorithms for cluster description: Classification and Regression Trees (CART) (Breiman et al., 2017), Iterative Mistake Minimization Trees (IMM) (Frost et al., 2020), and Prototype Descriptions (PROTO) (Carrizosa et al., 2021). We do not compare against the Disjoint-Tag Minimization Model (Davidson et al., 2018) as the approach requires data in a different form to the preceding algorithms.

We present results for both the low complexity (LC-PDP) and sparse (Sp-PDP) variants of our algorithm. We also consider two different settings for β and W: PDP-1 which has $W = \beta = 1$ and PDP-3 which has $W = 10$, $\beta = 3$. For the following results, the pre-fix of the algorithm denotes the objective used and the suffix denotes the setting for W and β. To construct an initial set of candidate half-spaces, for each cluster we enumerate the p maximum and minimum values for each feature ($p = 10$ for the following experiments) and construct half-spaces with uni-variate splits at each of the

Table 1. Cluster description accuracy (%). The percentage of data points in the original reference clustering that are correctly explained. Bolded numbers indicate best accuracy for each dataset.

dataset	n	m	K	IMM	CART	PROTO	PDP-1	PDP-3
adult	32561	108	3	99.93	99.63	66.40	**99.95**	**99.95**
bank	4521	51	7	**97.74**	92.79	80.1	**97.74**	**97.74**
default	30000	23	2	**100.00**	**100.00**	99.2	**100.00**	**100.00**
seeds	210	7	2	98.57	98.57	98.10	99.05	**100.00**
zoo	101	17	4	**100.00**	**100.00**	95.05	**100.00**	**100.00**
iris	150	4	2	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
framingham	3658	15	8	**100.00**	**100.00**	82.8	**100.00**	**100.00**
wine	178	13	2	97.19	97.19	96.63	**98.88**	**98.88**
libras	360	90	10	82.50	78.06	78.61	**98.06**	**98.06**
spam	4601	57	2	**99.98**	**99.98**	94.07	**99.98**	**99.98**

Table 2. Cluster description sparsity and complexity for explanation. Bolded numbers indicate best sparsity and complexity respectively for each dataset.

	Sparsity				Complexity			
dataset	IMM	CART	Sp-PDP-1	Sp-PDP-3	IMM	CART	LC-PDP-1	LC-PDP-3
adult	2	2	**1**	**1**	**10**	**10**	**10**	**10**
bank	6	6	**5**	**5**	44	42	**40**	**40**
default	**1**	**1**	**1**	**1**	**4**	**4**	**4**	**4**
seeds	**1**	**1**	2	3	**4**	**4**	**4**	**4**
zoo	**3**	**3**	**3**	**3**	18	18	**14**	**14**
iris	**1**	**1**	**1**	**1**	**4**	**4**	**4**	**4**
framingham	**3**	**3**	**3**	**3**	48	48	48	**44**
wine	**1**	**1**	4	2	**4**	**4**	10	6
libras	**9**	**9**	18	18	98	82	84	**80**
spam	**1**	**1**	**1**	**1**	**4**	**4**	**4**	**4**

values. For all results we set a 300 second time limit on the overall column generation procedure and a 30 second time limit on solving an individual pricing problem. We add all solutions found during the execution of the pricing problem with negative reduced cost to the master problem. We also use the grouping approach outlined in Section 3 for all datasets with more than 4000 data points, with the aim of getting the number of groups within $30000/K$. We perform the grouping by using hierarchical clustering with a maximum linkage of $\chi = 0.05$. All models were implemented in python using Gurobi 9.1 and run on a computer with 16 GB of RAM and a 2.7 GHz processor.

Table 1 shows the performance of each algorithm with respect to cluster description accuracy. Overall PDP is able to dominate the other benchmark algorithms, achieving the best accuracy on every benchmark dataset. Surprisingly, PDP-1 and PDP-3 perform almost identically, with PDP-3 only outperforming PDP-1 on the seeds dataset. Overall, PROTO is the least competitive approach, likely due to being the most restrictive function class relative to decision trees and polyhedra. Table 2 shows the number of features used in the cluster descriptions and their complexity. Note that PROTO does not appear in this table or the complexity table as the output for each cluster is simply a representative data point and a radius, and thus has no natural analog for sparsity or complexity. For CART and IMM we compute the complexity by considering each internal branching node as a half-space and report the total complexity of half-spaces needed to explain each cluster, counting a half-space multiple times if it is used to describe multiple clusters to provide a fair comparison to polyhedra. We report results for Sp-PDP as it directly optimizes this metric, whereas we report complexity for the LC-PDP. Sp-PDP performs competitively with IMM and CART getting the best sparsity in all but three datasets. Of the three datasets where it is outperformed by CART it is important to note that Sp-PDP achieves considerably better accuracy highlighting that the gains in explanation accuracy can come at a cost to the interpretability of the explanation. LC-PDP also performs competitively with the decision tree based approaches only being outperformed on datasets where it achieves higher cluster accuracy. A natural question is whether Sp-PDP and LC-PDP can achieve equal sparsity and low-complexity by sacrificing the increased accuracy. Figure 4 shows the Pareto curve of cluster description accuracy versus explanation sparsity on the wine dataset where CART and IMM achieve lower sparsity. The curve shows that PDP dominates the decision tree approaches, achieving equal sparsity at the same level of accuracy. Results in Appendix H show similar results for every dataset where PDP achieves lower

Table 3. Computation Time (s) for each algorithm. F-PDP represents the time to do the first phase solve (i.e. minimizing α).

Data	CART	IMM	PROTO	F-PDP-1	F-PDP-3	LC-PDP-1	LC-PDP-3	Sp-PDP-1	Sp-PDP-3
adult	0.06	0.34	900.00	300.00	300.00	300.00	300.00	300.00	300.00
bank	0.01	0.04	2100.00	300.00	300.00	300.00	300.00	300.00	300.00
default	0.04	0.11	600.00	2.98	3.19	2.80	4.53	300.00	300.00
framingham	0.01	0.02	2400.00	300.00	300.00	300.00	300.00	300.00	300.00
iris	0.00	0.00	135.91	0.11	0.09	0.18	0.16	0.59	0.56
libras	0.02	0.02	3000.00	300.00	300.00	300.00	300.00	300.00	300.00
seeds	0.00	0.00	600.00	0.63	1.66	300.00	300.00	300.00	300.00
spam	0.01	0.02	118.00	300.00	300.00	300.00	300.00	300.00	300.00
wine	0.00	0.00	489.52	300.00	300.00	300.00	300.00	300.00	300.00
zoo	0.00	0.00	72.55	0.18	0.13	300.00	300.00	300.00	300.00

sparsity or complexity.

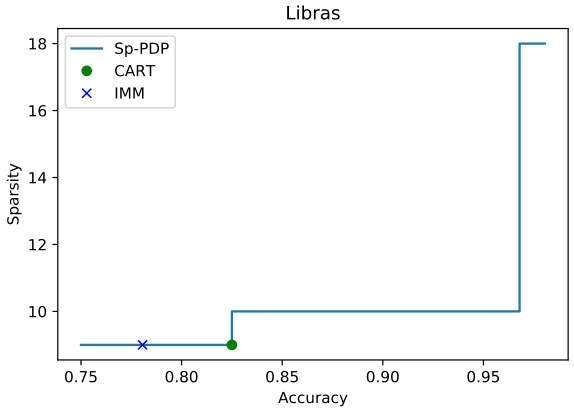

Figure 4. Pareto curve of cluster description accuracy and sparsity of the explanation on the LIBRAS dataset.

Table 3 summarizes the total computation time for all the methods. F-PDP-1/3 are the times needed for the first stage of our procedure (i.e. minimizing α). Although both decision tree methods run significantly faster than PDP, all results reported are run within a practical five minute time limit. PDP also outperforms the other IP based method (PROTO) on many datasets. It may appear unusual that PDP takes the full time limit on small datasets like wine. However, in these datasets there tends to be a high degree of degeneracy (i.e. many equivalent solutions with the same objective) and the column generation procedure continues until the certification of optimality. Thus even though PDP reaches the time limit, in many cases running the algorithm for a shorter time limit will give the same quality solution.

5. Conclusion

In this paper we introduced a novel approach for cluster description that works by constructing a polyhedron around each cluster. As opposed to existing approaches, our algorithm is able to explicitly optimize for the complexity or sparsity of the resulting explanations. We formulated the problem as an integer program and present both a column generation procedure to deal with an exponential number of candidate half-spaces and a grouping scheme to help the approach scale to large datasets. Compared to state of the art cluster description algorithms our approach is able to achieve competitive performance in terms of explanation accuracy and interpretablity when measured by sparsity and complexity.

Acknowledgements

This work was generously supported by Office of Naval Research (ONR) Grant N00014-21-1-2575.

References

Aly, M. Survey on multiclass classification methods. *Neural Networks*, 19(1):9, 2005.

Asuncion, A. and Newman, D. Uci machine learning repository, 2007.

Berk, R. *Criminal justice forecasts of risk: A machine learning approach*. Springer Science & Business Media, 2012.

Bertsimas, D., Orfanoudaki, A., and Wiberg, H. Interpretable clustering: an optimization approach. *Machine Learning*, 110(1):89–138, 2021.

Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and regression trees*. Routledge, 2017.

Carrizosa, E., Kurishchenko, K., Marın, A., and Romero, D. On clustering and interpreting with rules by means of mathematical optimization. *Unpublished Manuscript.*, 2021.

Carrizosa, E., Kurishchenko, K., Marín, A., and Romero Morales, D. Interpreting clusters via prototype optimization. *Omega*, 107:102543, 2022. ISSN 0305-0483. doi: https://doi.org/10.1016/j.omega.2021.102543. URL https://www.sciencedirect.com/science/article/pii/S0305048321001523.

Chen, J. *Interpretable Clustering Methods*. PhD thesis, Northeastern University, 2018.

Chen, J., Chang, Y., Hobbs, B., Castaldi, P., Cho, M., Silverman, E., and Dy, J. Interpretable clustering via discriminative rectangle mixture model. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 823–828. IEEE, 2016.

Dasgupta, S., Frost, N., Moshkovitz, M., and Rashtchian, C. Explainable k-means clustering: theory and practice. In *XXAI Workshop. ICML*, 2020.

Daudpota, S. M., Muhammad, A., and Baber, J. Video genre identification using clustering-based shot detection algorithm. *Signal, Image and Video Processing*, 13(7): 1413–1420, 2019.

Davidson, I., Gourru, A., and Ravi, S. The cluster description problem-complexity results, formulations and approximations. *Advances in Neural Information Processing Systems*, 31, 2018.

De Koninck, P., De Weerdt, J., et al. Explaining clusterings of process instances. *Data mining and knowledge discovery*, 31(3):774–808, 2017.

Dzyabura, D. and Yoganarasimhan, H. Machine learning and marketing. In *Handbook of Marketing Analytics*. Edward Elgar Publishing, 2018.

Fraiman, R., Ghattas, B., and Svarc, M. Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7(2):125–145, 2013.

Frost, N., Moshkovitz, M., and Rashtchian, C. Exkmc: Expanding explainable k-means clustering. *arXiv preprint arXiv:2006.02399*, 2020.

Gilmore, P. C. and Gomory, R. E. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859, 1961. ISSN 0030364X, 15265463. URL http://www.jstor.org/stable/167051.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022. URL https://www.gurobi.com.

Hair Jr, J. F. and Sarstedt, M. Data, measurement, and causal inferences in machine learning: opportunities and challenges for marketing. *Journal of Marketing Theory and Practice*, 29(1):65–77, 2021.

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

Kansal, T., Bahuguna, S., Singh, V., and Choudhury, T. Customer segmentation using k-means clustering. In *2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)*, pp. 135–139. IEEE, 2018.

Kauffmann, J., Esders, M., Montavon, G., Samek, W., and Müller, K.-R. From clustering to cluster explanations via neural networks. *arXiv preprint arXiv:1906.07633*, 2019.

Land, A. H. and Doig, A. G. An automatic method for solving discrete programming problems. *ECONOMETRICA*, 28(3):497–520, 1960.

Lawless, C., Kalagnanam, J., Nguyen, L. M., Phan, D., and Reddy, C. Interpretable clustering via multi-polytope machines. *arXiv preprint arXiv:2112.05653*, 2021.

Liu, B., Xia, Y., and Yu, P. Clustering through decision tree construction. *In Proceedings of the ACM International Conference on Information and Knowledge Management*, 10 2000. doi: 10.1145/354756.354775.

Ma, L. and Sun, B. Machine learning and ai in marketing–connecting computing power to human insights. *International Journal of Research in Marketing*, 37(3):481–504, 2020.

Moshkovitz, M., Dasgupta, S., Rashtchian, C., and Frost, N. Explainable k-means and k-medians clustering. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7055–7065. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/moshkovitz20a.html.

Pelleg, D. and Moore, A. Mixtures of rectangles: Interpretable soft clustering. In *ICML*, volume 2001, pp. 401–408, 2001.

Rajkomar, A., Dean, J., and Kohane, I. Machine learning in medicine. *New England Journal of Medicine*, 380(14): 1347–1358, 2019.

Rudin, C. and Ertekin, Ş. Learning customized and optimized lists of rules with mathematical programming. *Mathematical Programming Computation*, 10(4):659–702, 2018.

Rudin, C. and Ustun, B. Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice. *Interfaces*, 48(5):449–466, 2018.

Ustun, B. and Rudin, C. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

Ustun, B. and Rudin, C. Optimized risk scores. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1125–1134, 2017.

Varol, E., Sotiras, A., Davatzikos, C., Initiative, A. D. N., et al. Hydra: revealing heterogeneity of imaging and genetic patterns through a multiple max-margin discriminative analysis framework. *Neuroimage*, 145:346–364, 2017.

Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., and MacNeille, P. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, 18(1):2357–2393, 2017.

Wang, Y., Zhao, Y., Therneau, T. M., Atkinson, E. J., Tafti, A. P., Zhang, N., Amin, S., Limper, A. H., Khosla, S., and Liu, H. Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records. *Journal of biomedical informatics*, 102:103364, 2020.

Xu, R. and Wunsch, D. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

Završnik, A. Algorithmic justice: Algorithms and big data in criminal justice settings. *European Journal of criminology*, 18(5):623–642, 2021.

A. Illustrative Example on Non-Intersection of Polyhedral Descriptions

Consider a simple example where we have two clusters representing dogs and cats and two binary features - one indicating whether the animal barks and the other if it meows. A simple polyhedral description for these clusters is BARKS = TRUE for the dog cluster and MEOWS = TRUE for the cat cluster. However, the two polyhedra intersect in the improbable region where an animal both barks and meows. For this simple example, a solution could be to use BARKS = FALSE for the cat cluster. However, if we increase the number of animals, each with their own new binary feature for the noise they make (i.e. a frog cluster with a binary feature for ribbets), then our polyhedral descriptions can either intersect, with one simple half-space per cluster, or the description needs to add additional conditions (i.e. BARKS = FALSE and RIBBETS = FALSE for the cat cluster) which make the resulting description harder to interpret solely for scenarios that are unlikely to occur in real world data.

B. Proof of Theorem 2.1

Proof. We prove the hardness of the LC-PDP and Sp-PDP by showing the associated decision version is NP-Complete. Formally, the decision version of the PDP checks whether there exists a polyhedral description subject to a bound on the objective (i.e. complexity or sparsity respectively). This is in contrast to the optimization version of the PDP which involves minimizing the complexity or sparsity of the solution.

We start by noting that membership in NP is straightforward. Given a solution it is verifiable in polynomial time whether or not the given polyhedra correctly explain the given clusters and this certificate has encoding length polynomially bounded by that of the input. We prove NP-Hardness by a reduction from 3-SAT. Specifically, we consider yes-instances of the 3-SAT problem and show that if a given 3-SAT instance is a yes-instance a constructed LC-PDP instance is feasible.

Consider a 3-SAT problem with n variables $v_1, v_2, \ldots, v_n$ and m clauses $K_1, K_2, \ldots K_m$. Each clause K_i consists of three conditions $(v_{i1} \vee v_{i2} \vee v_{i3})$ where v_{ij} corresponds to either one of the original variables or its complement. We now construct a LC-PDP instance with $2n$ candidate half-spaces in $2n$ dimensional feature space with $m+n+1$ data points. We focus specifically on the simplest form of the problem - explaining only one cluster. Clearly if explaining one cluster is NP-Hard, explaining multiple cluster will also be NP-Hard. Let $\mathcal{C}_0$ be the cluster to be explained.

For each variable v_i in the 3-SAT instance, the PDP instance has two dimensions d_{v_i} and $d_{\bar{v}_i}$. The set of candidate half-spaces consist of $h_{v_i} = \{x : x_{d_{v_i}} \leq 0.5\}$ and $h_{\bar{v}_i} = \{x : x_{d_{\bar{v}_i}} \leq 0.5\}$ for each v_i . We next describe how to construct $n + m + 1$ data points for the PDP instance:

- We generate one data point x^0 in $\mathcal{C}_0$ that has a value of 0 for every feature.
- For each variable v_i in the original 3-SAT problem we add one new data point x^{v_i} outside the cluster to be explained that has 1s for features d_{v_i} and $d_{\bar{v}_i}$, and 0s otherwise.
- We also add one data point x^{K_i} for every clause K_i in the original 3-SAT problem, which has 1 for the features corresponding to the original conditions in the clause $d_{v_{i1}}, d_{v_{i2}}, d_{v_{i2}}$ and 0s otherwise. (For instance if $K_i = (v_1 \vee \bar{v}_2 \vee v_3)$, then the associated data point has 1s for features $d_{v_1}, d_{\bar{v}_2}, d_{v_3}$ and 0s for the rest.)

Finally we add a bound on the complexity of the instance of $2n$. Note that because each half-space uses one feature, this is equivalent to adding a constraint that at most n half-spaces can be used.

The above instance can clearly be set-up in polynomial time. It now suffices to show that solving the associated PDP yields a valid solution to the 3-SAT problem.

We start be claiming that the solution to the LC-PDP yields solutions where exactly one of h_{v_i} and $h_{\bar{v}_i}$ is used. Assume not, then the solution to the LC-PDP must have a solution where either both half-spaces h_{v_i} and $h_{\bar{v}_i}$ or neither one of them are used. However, at least one of h_{v_i} and $h_{\bar{v}_i}$ must be used, otherwise x^{v_i} would not be classified correctly. Moreover, a feasible solution cannot use multiple half-spaces corresponding to one variable, given that each variable has at least one half-space used, because it would contradict the complexity bound that at most n half-spaces used. Thus the claim must be true.

We can now interpret the half-spaces selected as the variable settings in the original 3-SAT problem (i.e. $v_i = T$ if h_{v_i} is selected and $v_i = F$ if $h_{\bar{v}_i}$ is selected). We now claim that any feasible solution to the LC-PDP corresponds to a solution of the 3-SAT instance. Note that for each data point outside $\mathcal{C}_0$ there exists at least one half-space selected that excludes it. By construction we know for every clause in the original 3-SAT problem there is an associated data point x^{K_i} outside the cluster to be explained that is only excluded by the half-spaces corresponding to the conditions in the clause $h_{v_{1i}}, h_{v_{2i}}, h_{v_{13}}$. Thus at least one of the half-spaces corresponding to the conditions must be used, and by extension every clause must be satisfied. Given that the PDP has no numerical data, the problem is also strongly NP-Complete. An identical proof also works if we replace the complexity bound with a sparsity bound (as each half-space uses a new dimension) thus also completing the claim for Sp-PDP. □

C. Proof of Theorem 3.1

Proof. We start by noting some properties of $COST_G(\mathbf{P})$ and $COST(\mathbf{P})$. First, for a fixed solution $\mathbf{P}$ we have $COST_G(\mathbf{P}) \geq COST(\mathbf{P})$, which follows from the fact that the grouped cost over-estimates error (i.e. counts all members of group as mis-classified if any individual data point in the group is mis-classified). By the optimality of the solutions $\mathbf{P}^*_G$, and $\mathbf{P}^*$, for the grouped and un-grouped problems respectively, we also have that $COST_G(\mathbf{P}^*_G) \leq COST_G(\mathbf{P}^*)$ and $COST(\mathbf{P}^*) \leq COST(\mathbf{P}^*_G)$ respectively. Rearranging the three inequalities we get:

$$COST_G(\mathbf{P}^*) \geq COST(\mathbf{P}^*_G) \geq COST(\mathbf{P}^*)$$

This implies that if we can get a bound on the difference between the grouped cost and full cost of $\mathbf{P}^*$ we can get a bound on the sub-optimality of $\mathbf{P}^*_G$ for the full problem.

Take $\mathbf{P}^*$ and consider the grouped cost relative to the original cost. Looking at each group G individually there are three possible cases: All the data points in a group are correctly classified, all data points in the group are misclassified, and the group has both data points that are both classified correctly and incorrectly. In the former two cases, the grouped cost is identical to the original cost, so it suffices to consider the last case. Note that the additional increase in cost for that group is equal to the number of correctly classified data points in the group. In the worst case, there are at most $|G| - 1$ such points. Thus, the cost in the grouped setting is at most $|G|$ times the original cost for data points in that group. Overall, in the worst case this is the only case in the dataset and every group it affects is the largest possible size $|G_{max}|$ completing the proof. Note that no aspect of the proof uses how the groups were constructed, so the result holds for any grouping scheme. □

D. Proof of Theorem 3.2

For reference, we start with the definition of the silhouette coefficient for a given clustering.

Definition D.1 (Silhouette Coefficient)**.** Consider data point $x^i \in C_k$, and a distance matrix d where entries d_{ij} capture distance between data points x^i and x^j.
Let $r(x^i)$ denote the average distance between data point x^i and every other data point in the same cluster:

$$r(x^i) = \frac{1}{|C_k| - 1} \sum_{x^j \in C_k} d_{ij}$$

Let $q(x^i)$ denote the average distance between data point x^i and every data point in the second closest cluster:

$$q(x^i) = \min_{l=1,\ldots,K: l \neq k} \frac{1}{|C_l|} \sum_{x^j \in C_l} d_{ij}$$

For data point x^i the silhouette score $s(x^i)$ is defined as:

$$s(x^i) = \frac{q(x^i) - r(x^i)}{\max(q(x^i), r(x^i))}.$$

The silhouette score for a set of cluster assignments is the average of the silhouette scores for all the data points. The possible values range from -1 (worst) to +1 (best).

Proof. Consider the following simple example with two clusters and a single feature x:

- For the first cluster C_1 there are three data points at the origin ($x = 0$) and m data points placed individually at increments of $-d_2$ (i.e. one data point at $x = -d_2$, one data point at $x = -2d_2$ and so on).
- For the second cluster C_2 there is one data point at the origin, 2 data points at $x = d_1$, and m data points

placed at increments of d_2 after d_1 (i.e. one data point at $x = d_1 + d_2$, one data point at $x = 2d_2 + d_1$ and so on).

- We set $d_1 < d_2$.

Figure 5 shows a visualization of the setting.

Consider the following groupings which we claim are optimal with respect to the silhouette coefficient. For C_1 all three data points at the origin form one group and every other data point is in its own group. Evidently this is the optimal grouping for $|G_1| = |C_1| - 2$ as every group has an intra-cluster distance of 0 and an inter-cluster distance of d_2 giving a silhouette score for the grouping of 1. For C_2 we group the one data point at the origin and the 2 data points at $x = d_1$ together, and every other data point is in its own group. Suppose this was not optimal with respect to the silhouette coefficient for $|G_2| = |C_2| - 2$. Clearly an optimal grouping will have the two points at $x = d_1$ together as they have an intra-cluster distance of 0. Thus the only scenarios are that the point at $x = d_1 + d_2$ is included in that group or two of the m points spaced at increments of d_2 are grouped together. Simple arithmetic shows that both scenarios result in a silhouette coefficient larger than the given grouping, proving its optimality.

An optimal solution to the original problem is to use a single half-space $\{x \in \mathbb{R} : x \leq 0\}$ for C_1 and $\{x \in \mathbb{R} : x \geq d_1\}$ for C_2 respectively, which incurs a cost of 1. Note that under the optimal grouping scheme outlined above one group with 3 points from C_1 overlaps with one group with 3 points from C_2. Thus an optimal solution to the grouped problem is to use a single half-space $\{x \in \mathbb{R} : x \leq d_1\}$ for C_1 and $\{x \in \mathbb{R} : x \geq d_1 + \epsilon\}$, where $\epsilon < d_2$, for C_2 respectively as no solution will incur a grouped cost less than 3. This optimal solution to the grouped problem incurs a true cost of 3 (as the three points in 3 point group in C_2 are misclassified), completing our claim.

□

E. Implementation Details

We pre-process all datasets by using a min-max scaler to normalize numeric feature values between 0 and 1, encode all categorical features using one-hot encoding, and for all supervised learning datasets remove the target variable. To create a reference cluster assignment we use k-means clustering using k-means++ initialization scheme with 100 random restarts. To select the number of clusters we tune k between 2 and 10 and select the k with the best silhouette score.

For all approaches we used the same k-means clustering as a reference cluster assignment to be explained. For CART we used the cluster assignments as labels for the classifier. For both CART and IMM we set the number of leaf nodes to be the number of clusters to provide a fair comparison to the polyhedral description approach. While IMM is an algorithm for generating new clusters not explaining the reference clustering, we interpreted the resulting tree as an explanation for the initial clustering. While in principle IMM should under-perform CART which explicitly optimizes for classification accuracy we found that IMM outperformed CART with respect to explanation accuracy on a number of datasets. We implemented the Prototype description IP model using Gurobi 9.1 (Gurobi Optimization, LLC, 2022) and Python, and placed a 300 second time limit on the solution time. To allow the prototype description model to scale to larger datasets we implemented the sub-sampling scheme outlined in the original paper and sub-sampled 125 candidate prototypes and 500 data points for each cluster.

F. Qualitative Comparison

Figure 6 shows three sample cluster descriptions for the zoo dataset to compare each model class's interpretability. For this example we use the best reference k-means clustering which resulted in four clusters, and describe the second cluster (which is composed primarily of birds). The prototype explanation for the cluster is a ladybird. While having a representative animal is easy to understand, without the added context that the cluster is primarily birds it is not obvious what are the defining characteristics of the cluster. For instance, ladybirds are also predators and have eggs, which could also define clusters. The decision tree description requires that the cluster has no tail, is a predator, and is not domestic. Compared to both the decision tree and prototype explanation, the polyhedral description, simply that the cluster is all airborne, provides a parsimonious summary of the cluster that gives intuition about its defining characteristic. This further underscores that a full partition of the feature space for a description, as necessary for a decision tree, may lead to more complicated descriptions.

G. Additional Computation Time Results

To give a sense for how PDP scales with n, m and K, Tables 4 and 5 show the computation time to solve the final restricted master integer program and the pricing problem respectively. The results in Table 4 show that the algorithm scales reasonably with n, m and K. Only one result exceeds the 300s time limit - Libras for Sp-PDP-3. This is due to the fact that the this dataset has 90 features and the reference clustering has 10 clusters and our formulation scales with both the number of candidate half-spaces (impacted by m) and the number of clusters. We note that the two largest datasets, adult and default, both have solve times within approximately one minute.

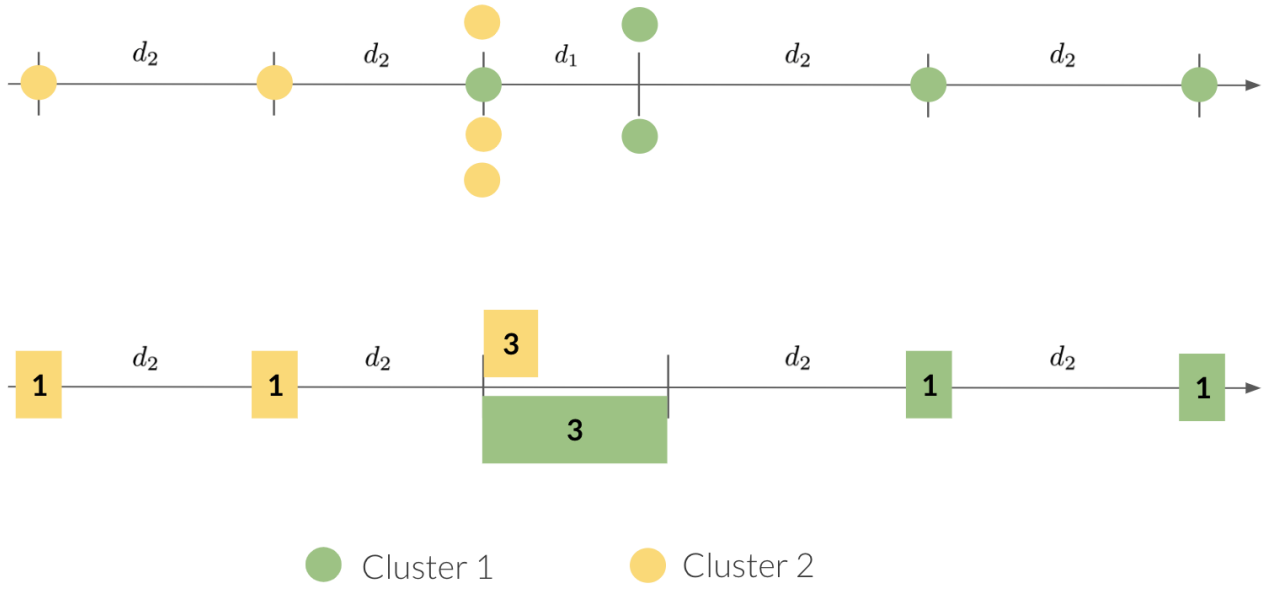

Figure 5. (Top) Visualization of points to be explained in instance for Theorem 3.2. (Bottom) Optimal grouping with respect to silhouette coefficient for $|G_k| = |C_k| - 1$.

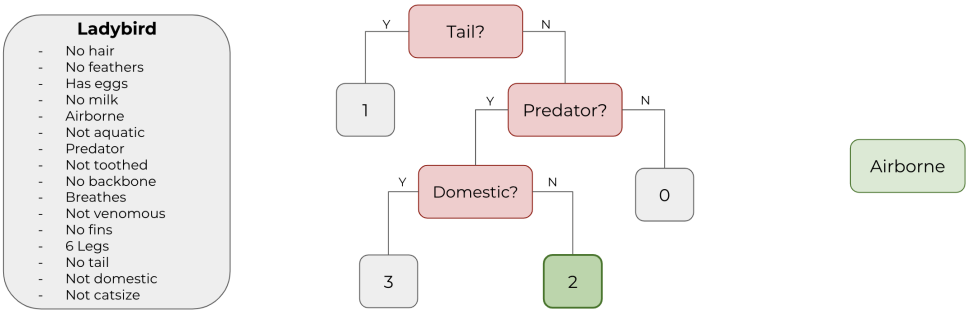

Figure 6. Sample cluster descriptions for the same cluster on the zoo dataset. (Left) A prototype. (Middle) A decision tree. (Right) A polyhedral description.

Table 5 shows the average time to solve the pricing problem program during column generation. For the most part, all the pricing problems can be solved in under 5 seconds. The exception again being libras which has 90 features and 10 reference clusters and reaches the 30 second time limit on multiple occasions.

H. Pareto Curves

For the majority of datasets evaluated in Section 4, PDP either dominates the benchmark algorithms (i.e. achieves better accuracy and/or interpretability) or achieves the same performance. However, there are three datasets where the comparison is inconclusive. In wine, PDP achieves better accuracy but higher complexity. Likewise in the seeds and libras datasets PDP achieves better accuracy but higher sparsity. A natural question is whether PDP can achieve comparable accuracy by sacrificing accuracy. Figures 7 and 4 show the pareto curves for accuracy versus complexity and sparsity. The results show that PDP in fact dominates the existing algorithms on all three datasets, achieving the same interpretability at the same level of accuracy while also being able to achieve better accuracy at lower interpretability.

Table 4. Computation Time (s) to solve the Master Integer Program after column generation.

dataset	n	m	K	Feas-PDP-1	Feas-PDP-3	LC-PDP-1	LC-PDP-3	Sp-PDP-1	Sp-PDP-3
zoo	101	17	4	0.02	0.01	0.03	0.05	0.07	0.13
iris	150	4	2	0.00	0.00	0.00	0.00	0.00	0.00
wine	178	13	2	0.09	0.16	0.23	0.29	0.35	0.49
seeds	210	7	2	0.02	0.02	0.08	0.03	0.09	0.03
libras	360	90	10	28.64	42.03	33.39	112.95	191.76	300.00
framingham	3658	15	8	3.87	4.84	1.95	1.62	2.72	2.33
bank	4521	51	7	88.51	85.60	107.78	115.11	176.62	221.22
spam	4601	57	2	2.22	2.12	2.96	2.65	3.21	5.17
default	30000	23	2	0.81	0.84	0.25	0.25	1.07	0.53
adult	32561	108	3	52.37	60.41	11.13	10.60	18.40	20.85

Table 5. Average computation time (s) to solve the Pricing Problem during execution of column generation framework.

dataset	n	m	K	Feas-PDP-1	Feas-PDP-3	LC-PDP-1	LC-PDP-3	Sp-PDP-1	Sp-PDP-3
zoo	101	17	4	0.03	0.02	0.02	0.02	0.02	0.02
iris	150	4	2	0.02	0.01	0.01	0.01	0.01	0.01
wine	178	13	2	0.02	0.02	0.02	0.02	0.04	0.11
seeds	210	7	2	0.04	0.03	0.03	0.02	0.06	0.18
libras	360	90	10	1.72	30.00	13.53	30.00	21.83	30.00
framingham	3658	15	8	1.10	1.06	1.08	1.03	1.09	1.05
bank	4521	51	7	2.04	4.17	2.30	4.10	2.30	7.68
spam	4601	57	2	0.39	0.39	0.42	0.40	0.56	0.46
default	30000	23	2	1.62	1.82	2.30	4.03	1.89	1.96
adult	32561	108	3	3.00	3.61	2.52	6.67	2.93	9.03

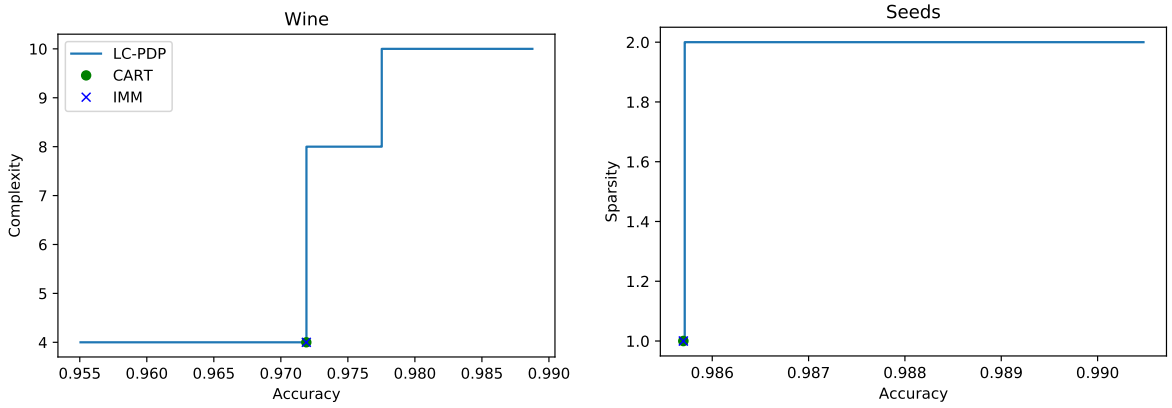

Figure 7. Pareto curves for wine and seeds with respect to accuracy and complexity and sparsity respectively. Note that for both plots the marker for CART and IMM are at the same point.