Optimal Decision Trees For Interpretable Clustering with Constraints

Pouya Shati [1,3] , **Eldan Cohen** [2] and **Sheila McIlraith** [1,3]
[1]Department of Computer Science, University of Toronto, Toronto, Canada
[2]Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Canada
[3]Vector Institute, Toronto, Canada
{pouya, sheila}@cs.toronto.edu, ecohen@mie.utoronto.ca

Abstract

Constrained clustering is a semi-supervised task that employs a limited amount of labelled data, formulated as constraints, to incorporate domain-specific knowledge and to significantly improve clustering accuracy. Previous work has considered exact optimization formulations that can guarantee optimal clustering while satisfying all constraints, however these approaches lack interpretability. Recently, decision trees have been used to produce inherently interpretable clustering solutions, however existing approaches do not support clustering constraints and do not provide strong theoretical guarantees on solution quality. In this work, we present a novel SAT-based framework for interpretable clustering that supports clustering constraints and that also provides strong theoretical guarantees on solution quality. We also present new insight into the trade-off between interpretability and satisfaction of such user-provided constraints. Our framework is the first approach for *interpretable* and *constrained* clustering. Experiments with a range of real-world and synthetic datasets demonstrate that our approach can produce high-quality and interpretable constrained clustering solutions.

1 Introduction

Clustering is a core unsupervised machine learning problem that aims to partition a dataset into subgroups of similar data points. In practice, it is often used to discover meaningful sub-populations such as customer segments [Wu and Chou, 2011] or groups of correlated genes [Thalamuthu *et al.*, 2006]. Constrained clustering is a semi-supervised learning task that exploits small amounts of supervision, provided in the form of constraints, to incorporate domain-specific knowledge and to significantly improve clustering performance [Wagstaff and Cardie, 2000; Wagstaff *et al.*, 2001]. The most popular types of clustering constraints are so-called *instance-level pairwise must-link* constraints, *cannot-link* constraints, and additional types of constraints that can be translated to such pairwise constraints [Davidson and Ravi, 2005; Liu and Fu, 2015]. In the past decades, the topic of constrained clustering has received significant attention and different constrained clustering algorithms have been proposed [Bilenko *et al.*, 2004; Pelleg and Baras, 2007; Liu *et al.*, 2017; Cohen *et al.*, 2020]. In particular, approaches that are based on exact optimization formulations, such as integer programming, constraint programming, and satisfiability have obtained state-of-the-art performance [Davidson *et al.*, 2010; Berg and Järvisalo, 2017; Dao *et al.*, 2017; Babaki *et al.*, 2014].

Our interest is in developing an approach to constrained clustering that yields interpretable solutions with strong solution quality guarantees. Recent approaches to clustering via decision trees have yielded a degree of interpretability by exposing clustering rationale in the branching structure of the tree [Frost *et al.*, 2020; Moshkovitz *et al.*, 2020; Bertsimas *et al.*, 2021; Gamlath *et al.*, 2021]. However, these approaches do not support the incorporation of clustering constraints. Moreover, little is known about the compatibility of such clustering constraints with decision-tree clustering. While optimal decision trees have received significant attention in recent years, most of the literature is focused on classification tasks [Ignatiev *et al.*, 2021].

In this work, we present the first approach for *interpretable* and *constrained* optimal clustering based on decision trees. We make the following contributions:

1. We present a novel SAT-based encoding of optimal constrained clustering that is interpretable via a decision tree. Our formulation supports two well-known clustering objectives as well as pairwise instance-level clustering constraints. To the best of our knowledge, this is the first approach for inherently-interpretable constrained clustering.
2. We introduce novel techniques for efficient encoding of clustering problems and for improving search performance. Specifically, we introduce the notion of distance classes to support bounded suboptimal clustering, and a set of pruning rules to reduce the number of clauses.
3. We empirically evaluate our approach over real-world and synthetic datasets and show that it leads to high-quality, inherently-interpretable clustering solutions that satisfy a given set of pairwise constraints.
4. We present theoretical and empirical results on the trade-off between interpretability via decision trees and satisfaction of user-provided clustering constraints.

2 Preliminaries

2.1 Decision Trees for Constrained Clustering

Definition 1 (Decision Tree). *Given a set of features F and number of clusters k, a decision tree $\mathcal{D}$ is a set of branching nodes $\mathcal{T}_B$, a set of leaf nodes $\mathcal{T}_L$, a root node $\delta \in \mathcal{T}_B$, a parent function $p : \mathcal{T}_B \cup \mathcal{T}_L \rightarrow \mathcal{T}_B$, left and right child functions $l, r : \mathcal{T}_B \rightarrow \mathcal{T}_B \cup \mathcal{T}_L$, a node feature selection function $\beta : \mathcal{T}_B \rightarrow F$ together with a threshold selection function $\alpha : \mathcal{T}_B \rightarrow dom(F)$, and finally a leaf labelling function $\theta : \mathcal{T}_L \rightarrow [1..k]$.*

Definition 2 (Tree Cluster Assignment). *Given a dataset X, and the number of clusters K, a decision tree $\mathcal{D}$ direct each point $x \in X \subset R^{|F|}$ to one of its leaves by starting from the root and recursively moving to its left or right child, depending on whether the value of the chosen feature is less than or equal to the threshold, or not. The point is then assigned the cluster label of the leaf, represented by $\Theta_{\mathcal{D}}(x)$.*

The tree clustering $\Theta_{\mathcal{D}}$ partitions X into k clusters such that each cluster is non-empty,

$$\sum_{x_i \in X} \mathbb{1}[\Theta_{\mathcal{D}}(x_i) = k] \geq 1 \qquad \forall k \in 1..K.$$

Consistent with recent work [Frost et al.*, 2020], each cluster is allowed to spread across multiple leaves to improve expressiveness and the ability to satisfy user-provided constraints. If two points are in the same cluster (resp. different clusters), they are said to be clustered together (resp. separately).*

Definition 3 (Pairwise Clustering Constraints). *Given a dataset X, a set of pairs of data points named must-link constraints ML, and a set of pairs of data points named cannot-link constraints CL, a clustering $\Theta_{\mathcal{D}}$ is said to respect the constraints if it clusters all of must-link pairs together and all of cannot-link pairs separately:*

$$\forall (x_1, x_2) \in ML : \Theta_{\mathcal{D}}(x_1) = \Theta_{\mathcal{D}}(x_1) \tag{1}$$

$$\forall (x_1, x_2) \in CL : \Theta_{\mathcal{D}}(x_1) \neq \Theta_{\mathcal{D}}(x_1) \tag{2}$$

Definition 4 (Minimum Split and Maximum Diameter). *Given a dataset X, the number of clusters k, and a clustering $\Theta_{\mathcal{D}} : X \rightarrow 1..k$, the minimum split is the shortest distance between two points that are clustered separately,*

$$MS_{\mathcal{D}} = min(\{|x_1 - x_2| \,|\, \Theta_{\mathcal{D}}(x_1) \neq \Theta_{\mathcal{D}}(x_2)\}, \tag{3}$$

and the maximum diameter is the longest distance between two points that are clustered together,

$$MD_{\mathcal{D}} = max(\{|x_1 - x_2| \,|\, \Theta_{\mathcal{D}}(x_1) = \Theta_{\mathcal{D}}(x_2)\}. \tag{4}$$

We assume $|x_i - x_j|$ to be the Euclidean distance, however any totally ordered measure of distance is applicable.

2.2 Problem Definitions

We consider two well-known clustering objectives [Dao *et al.*, 2017]: (1) minimizing the maximum diameter (MD); (2) a bi-criteria objective that consists of minimizing the maximum diameter and maximizing the minimum split ([MD,MS]). Specifically, we consider bounded approximations parameterized by ϵ such that $\epsilon = 0$ indicates an optimal solution (or Pareto optimal for [MD,MS]).

Problem 1 (MD ϵ-Optimal Tree Clustering). *Given a dataset X, number of clusters k, a set of must-links constraints ML, a set of cannot-links constraints CL, decision tree depth d, and approximation parameter ϵ, find a complete decision tree $\mathcal{D}$ of depth d such that:*

1. *$\Theta_{\mathcal{D}}$ respects the constraints ML and CL;*
2. *$MD_{\mathcal{D}} \leq MD_{\mathcal{D}^*} + \epsilon$,*

where $\mathcal{D}^*$ is an optimal solution with respect to the objective of minimizing the maximum diameter.

Problem 2 (MS-MD ϵ-Pareto Optimal Tree Clustering). *Given a dataset X, number of clusters k, a set of must-links constraints ML, a set of cannot-links constraints CL, decision tree depth d, and approximation parameter ϵ, find a complete decision tree $\mathcal{D}$ of depth d such that:*

1. *$\Theta_{\mathcal{D}}$ respects the constraints ML and CL;*
2. *$MS_{\mathcal{D}} \geq MS_{\mathcal{D}^*} - \epsilon$;*
3. *$MD_{\mathcal{D}} \leq MD_{\mathcal{D}^*} + \epsilon$,*

where $\mathcal{D}^*$ is a Pareto optimal solution with respect to the bi-criteria objective of maximizing the minimum split and minimizing the maximum diameter.

Note that Problem 1 and Problem 2 are not always feasible since a decision tree of a given depth cannot represent all possible clusterings, and may not be able to represent any clustering that satisfies the ML and CL constraints. However in Proposition 1 we show that for every clustering there exists a tree of certain depth that can represent it.[1]

Proposition 1 (Decision Tree Completeness). *Given a dataset X and an arbitrary clustering Θ_C such that*

$$\forall x_1, x_2 : (\Theta_C(x_1) \neq \Theta_C(x_2)) \rightarrow (\exists j \in F : x_1[j] \neq x_2[j])$$

there exists a complete decision tree $\mathcal{D}$ of sufficiently high depth d, which partitions X into the same clusters $\forall x \in X : \Theta_{\mathcal{D}}(x) = \Theta_C(x)$.[2]

3 SAT-based $\epsilon-$Optimal Tree Clustering

In this section, we present our approach for solving the problems in Section 2.2. First, we present a way to simplify our handling of pairs while still maintaining the approximation guarantee. Then, we present a novel SAT-based encoding for interpretable and constrained tree clustering.

3.1 Distance Classes

To encode the ϵ-approximation in Problem 1 and Problem 2, we divide the set of all pairs of data points based on their distance into μ non-overlapping intervals called distance classes $\mathcal{D} = \{D_1, D_2, ..., D_\mu\}$ such that the smallest and largest distances in each class are less than ϵ apart. Specifically, we employ a greedy procedure that sorts all pairs from the smallest distance to the largest and greedily adds them one by one,

[1] We only consider cases where the ML and CL constraints are consistent, i.e., where there exists a clustering that satisfies these constraints. For example, if the same pair of points is found in both the ML and CL constraints, there exists no clustering that would satisfy these constraints.

[2] All proofs appear in the extended version [Shati *et al.*, 2023].

creating new classes as needed to guarantee that the distances in each class are at most ϵ apart. All pairs in the same class are treated similarly w.r.t. being clustered together or not.

In order to maximize the minimum split, we consider the index λ^+ such that any pair of data points in distance classes $D_1..D_{\lambda^+}$ must be clustered together (Eq. (5)). Similarly, to minimize the maximum diameter, we consider the index λ^- such that any pair in distance classes $D_{\lambda^-+1}..D_\mu$ must be clustered separately (Eq. (6)).

$$((x_1, x_2) \in D_w, w \le \lambda^+) \rightarrow \Theta_C(x_1) = \Theta_C(x_2) \quad (5)$$
$$((x_1, x_2) \in D_w, w > \lambda^-) \rightarrow \Theta_C(x_1) \neq \Theta_C(x_2) \quad (6)$$

Note that in any feasible clustering, for all λ^+ and λ^- values that satisfy (Eqs. (5) and (6)), we have that MS $\geq$ $min(\{||x_1 - x_2|| \mid (x_1, x_2) \in D_{\lambda^++1}\})$ and MD $\leq$ $max(\{||x_1 - x_2|| \mid (x_1, x_2) \in D_{\lambda^-}\})$. We therefore focus on optimizing the indices λ^+ and λ^- to obtain ϵ-optimal solutions to Problem 1 and Problem 2 (Proposition 2).

Proposition 2 (MD and MS-MD ϵ-approximation)**.** *Let Θ_D be a tree cluster assignment. We have that:*

1. *If Θ_D is an optimal solution w.r.t minimizing λ^- then Θ_D is an ϵ-optimal solution w.r.t the MD objective.*
2. *If Θ_D is a Pareto-optimal solution w.r.t minimizing λ^- and maximizing λ^+ then Θ_D is an ϵ-Pareto optimal solution w.r.t the bi-criteria MD-MS objective.*

3.2 MaxSAT Encoding

A SAT formula is a conjunction of clauses, a clause is a disjunction of literals, and a literal is either a Boolean variable or its negation. A clause is *satisfied* if at least one of its literals is true. The SAT problem consists of finding an assignment of the variables that satisfies all clauses in a formula [Biere *et al.*, 2009]. We model the construction of ϵ-optimal constrained clustering trees as Partial MaxSAT, an optimization variant of SAT that divides clauses into *hard* and *soft* clauses, requiring variable assignments to satisfy all hard clauses and to maximize the number of satisfied soft clauses.

Variables

Table 1 describes the set of Boolean variables.

$a_{t,j}$	Feature j is chosen for the split at branching node t
$s_{i,t}$	Point i is directed towards the left child, if it passes through branching node t
$z_{i,t}$	Point i ends up at leaf node t
$g_{t,c}$	The cluster assigned to leaf t is or comes after c
$x_{i,c}$	The cluster assigned to point i is or comes after c
b^-_w	(The negation of) whether the pairs in distance class w should be clustered separately
b^+_w	The pairs in class w should be clustered together

Table 1: Boolean Variables in our Model

Clauses

We encode the construction of the decision tree (Eqs. (7)-(15)) following Shati *et al.* [2021], a state-of-the-art SAT-based decision-tree classifier that naturally supports numeric features that are prevalent in clustering problems. These clauses guarantee that exactly one feature is chosen at each branching node (Eqs. (7)-(8)), the points are directed to the left or right child of each branching node based on their value of the chosen feature (Eqs. (9)-(10)), the appearance of points at leaves correctly correspond to the path that they are directed through within the tree (Eqs. (11)-(13)), and thresholds are non-trivial (Eqs. (14)-(15)).[3][4]

$$(\neg a_{t,j}, \neg a_{t,j'}) \quad \forall t \in \mathcal{T}_B, j \neq j' \in F \quad (7)$$
$$(\bigvee_{j \in F} a_{t,j}) \quad \forall t \in \mathcal{T}_B \quad (8)$$
$$(\neg a_{t,j}, s_{i,t}, \neg s_{i',t}) \quad \forall t \in \mathcal{T}_B, j \in F, (i, i') \in O_j(X) \quad (9)$$
$$(\neg a_{t,j}, \neg s_{i,t}, s_{i',t}) \quad \forall t \in \mathcal{T}_B, j \in F, (i, i') \in O_j(X), x_i[j] = x_{i'}[j] \quad (10)$$
$$(\neg z_{i,t}, s_{i,t'}) \quad \forall t \in \mathcal{T}_L, x_i \in X, t' \in A_l(t) \quad (11)$$
$$(\neg z_{i,t}, \neg s_{i,t'}) \quad \forall t \in \mathcal{T}_L, x_i \in X, t' \in A_r(t) \quad (12)$$
$$(z_{i,t}, \bigvee_{t' \in A_l(t)} \neg s_{i,t'}, \bigvee_{t' \in A_r(t)} s_{i,t'}) \quad \forall t \in \mathcal{T}_L, x_i \in X \quad (13)$$
$$(\neg a_{t,j}, s_{\#^1_j,t}) \quad \forall t \in \mathcal{T}_B, j \in F \quad (14)$$
$$(\neg a_{t,j}, \neg s_{\#^{|X|}_j,t}) \quad \forall t \in \mathcal{T}_B, j \in F \quad (15)$$

The following clauses extend the basic decision tree encoding to support ϵ-optimal clustering trees that satisfy the ML and CL constraints. The clauses in Eq. (16) guarantee well-formed unary encoding of cluster labels in each leaf.[5]

$$(g_{t,c}, \neg g_{t,c+1}) \quad \forall t \in \mathcal{T}_L, c \in [1..k-2] \quad (16)$$

Eqs. (17)-(18) guarantee that the label assigned to each data point matches the label of leaf the data point reaches.

$$(\neg z_{i,t}, \neg g_{t,c}, x_{i,c}) \quad \forall t \in \mathcal{T}_L, x_i \in X, c \in [1..k-1] \quad (17)$$
$$(\neg z_{i,t}, g_{t,c}, \neg x_{i,c}) \quad \forall t \in \mathcal{T}_L, x_i \in X, c \in [1..k-1] \quad (18)$$

Eqs. (19)-(20) break the ties between the equivalent cluster assignments. We consider two cluster assignments Θ^1 and Θ^2 equivalent if there exists a relabelling, $\Gamma : K \rightarrow K$, such that $\Gamma \circ \Theta^1 = \Theta^2$. To break ties, we force each x_i in the ascending order to be assigned to the first empty cluster, if it needs a new one. This guarantees that there are no two feasible solutions that are relabellings of each other. Note that we do not eliminate viable solutions, since any clustering can be renamed into an equivalent one that respects this property.

$$(\neg x_{c,c}) \quad \forall c \in [1..k-1] \quad (19)$$
$$(\neg x_{i,c}, \bigvee_{i' < i} x_{i',c-1}) \quad \forall x_i \in X, c \in [2..k-1], c < i \quad (20)$$

[3]The set $O_j(X)$ contains all consecutive pairs of points when ordered according to feature j and the set $A_l(t)$ ($A_r(t)$) contains all nodes that have t as descendent of their left (right) child.

[4]For a detailed description of the clauses in Eqs. (7)-(15), we refer the reader to Shati *et al.* [2021].

[5]A unary encoding is well-formed if it does not include the sequence 01 at any point.

The clause in Eq. (21) alongside the tie-breaking ones guarantee that all clusters are non-empty, assuming $|X| \geq K$. If we substitute k with $k' < k$ in Eq. (21), k' clusters are guaranteed to be non-empty. Thus, we can enforce minimum k' and maximum k clusters. In our experiments we focus on the setting where all clusters are non-empty.

$$(\bigvee_i x_{i,k-1}) \tag{21}$$

Eqs. (22)-(24), namely the unconditional separating clauses, guarantee that pairs in CL are clustered separately.

$$(x_{i,1}, x_{i',1}) \qquad \forall(i,i') \in CL \tag{22}$$

$$(\neg x_{i,k-1}, \neg x_{i',k-1}) \qquad \forall(i,i') \in CL \tag{23}$$

$$(\neg x_{i,c}, \neg x_{i',c}, x_{i,c+1}, x_{i',c+1}) \quad \forall(i,i') \in CL, c \in [1..k-2] \tag{24}$$

Eqs. (25)-(26), namely the unconditional co-clustering clauses, guarantee that pairs in ML are clustered together.

$$(\neg x_{i,c}, x_{i',c}) \qquad \forall(i,i') \in ML, c \in [1..k-1] \tag{25}$$

$$(x_{i,c}, \neg x_{i',c}) \qquad \forall(i,i') \in ML, c \in [1..k-1] \tag{26}$$

Eqs. (27)-(29), namely the conditional separating clauses, guarantee that a b^-_w variable being set to false forces the pairs in distance class w to be clustered separately.

$$(b^-_w, x_{i,1}, x_{i',1}) \qquad \forall D_w \in \mathcal{D}, (i,i') \in D_w \tag{27}$$

$$(b^-_w, \neg x_{i,k-1}, \neg x_{i',k-1}) \qquad \forall D_w \in \mathcal{D}, (i,i') \in D_w \tag{28}$$

$$(b^-_w, \neg x_{i,c}, \neg x_{i',c}, x_{i,c+1}, x_{i',c+1}) \quad \forall D_w \in \mathcal{D}, (i,i') \in D_w, c \in [1..k-2] \tag{29}$$

Eqs. (30)-(31), namely the conditional co-clustering clauses, guarantee that a b^+_w variable being set to true forces the pairs in distance class w to be clustered together.

$$(\neg b^+_w, \neg x_{i,c}, x_{i',c}) \quad \forall D_w \in \mathcal{D}, (i,i') \in D_w, c \in [1..k-1] \tag{30}$$

$$(\neg b^+_w, x_{i,c}, \neg x_{i',c}) \quad \forall D_w \in \mathcal{D}, (i,i') \in D_w, c \in [1..k-1] \tag{31}$$

Eqs. (32)-(34) guarantee that the distance classes are partitioned according to valid λ^+ and λ^- values, i.e., that $D_1..D_{\lambda^+}$ distance classes are clustered together, and $D_{\lambda^-+1}..D_\mu$ distance classes are clustered separately.

$$(\neg b^-_w, b^-_{w-1}) \qquad \forall D_w \in \mathcal{D}, w > 1 \tag{32}$$

$$(\neg b^+_w, b^+_{w-1}) \qquad \forall D_w \in \mathcal{D}, w > 1 \tag{33}$$

$$(\neg b^+_w, b^-_w) \qquad \forall D_w \in \mathcal{D} \tag{34}$$

Smart Pairs

Since each pair of points has corresponding clauses that handle being clustered together or separately, a naive encoding will be quadratic in the number of points $|X|$. This is significant since all the other parts of the encoding are linear in $|X|$. To reduce the number of clauses, we see the set of points as nodes in a graph and exploit connections between pairs. Pairs that are forced to be clustered together are represented by *positive* edges and pairs that are force to be clustered separately by *negative* edges. The positive edges imply a set of connected components of points that will be clustered together while a negative edge between nodes in different components indicates that each of the components are mutually exclusive, i.e., each component will be in a different cluster. The order $<^*$, which sorts the pairs based on distance and breaks ties arbitrary, is used to greedily build the set of positive edges (E^+) and the set of negative edges (E^-). As we incrementally build the sets, each new edge can be classified as *inner* or *crossing*, based on the previous members in E^+ and E^-, to help us detect infeasibility or redundancy of clauses.

Definition 5 (Inner and Crossing edges)**.** *Given the sets E^+ and E^-, a new edge is said to be an:*

- *Inner edge, if it connects two nodes within an existing connected component based on E^+.*
- *Crossing edge, if it connects two nodes in two connected components based on E^+ that are mutually exclusive based on E^-.*

We will use edges and the pairs of points that they represent interchangeably onward.

Since E^+ (E^-) represents the set of pairs that are forced to be clustered together (separately), an inner pair is forced to have the same label and a crossing pair to have different labels. We make use of this fact to avoid adding co-clustering clauses Eqs. (25, 26, 30, 31) or separating clauses Eqs. (22, 23, 24, 27, 28, 29) when it is redundant to do so. Furthermore, we can conclude infeasibility when an inner (crossing) pair is forced to be clustered separately (together).

Our treatment of the unconditional clauses Eqs. (22-26) and of the conditional ones Eqs. (27-31) differ in two ways.

1. For the unconditional clauses the E^+ and E^- sets are respectively constructed from ML and CL links. For conditional ones however, we also include the pairs that are implied to be co-clustered (separated), due to the order of distance imposed by λ^+ and λ^- values. Note that implied pairs for the processing of conditional co-clustering clauses are not valid for the processing of conditional separating clauses.
2. If the unconditional separation or co-clustering of a pair is infeasible, the problem is infeasible. However, for conditional clauses, we only fix the corresponding b^+_w or b^-_w variable to satisfy the conditional clause.

The detailed procedure is described in the extended version [Shati *et al.*, 2023].

Objective

In Section 3.1 we established that in order to find an ϵ-approximation of an MS-MD Pareto optimal solution, we need to find a solution that is Pareto optimal with regards to maximizing λ^+ and minimizing λ^-. Similarly, for an ϵ-approximation of an MD optimal solution we need to find a solution that minimizes λ^-. Note that λ^+ and λ^- are the number of b^+_w and b^-_w variables set to true, respectively.

$$\lambda^+ = \sum_w \mathbb{1}(b^+_w = true) \tag{35}$$

$$\lambda^- = \sum_w \mathbb{1}(b^-_w = true) \tag{36}$$

To minimize λ^-, we simply introduce a soft clause with unit weight for each $\neg b^-_w$ (Eq. (37)). To obtain a Pareto-optimal solution, we can optimize any function that is increasing w.r.t λ^- and decreasing w.r.t λ^+ as objective. We opt for minimizing the simple combination of $\lambda^- - \lambda^+$ and model it using the soft clauses in Eq. (37) and Eq. (38).

$$(\neg b^-_w) \qquad w \in W \qquad (37)$$

$$(b^+_w) \qquad w \in W \qquad (38)$$

For the MD objective, the b^+_w variables and the clauses in Eqs. (30)-(31) become irrelevant and can be removed.

4 Experiments

4.1 Experiment Setup

We use the Loandra solver [Berg *et al.*, 2019] to solve our tree clustering encoding. Loandra is an any-time solver that guarantees optimality if run to completion, but can also produce intermediate solutions. We run experiments on a server with two 12-core Intel E5-2697v2 CPUs and 128G of RAM.

Datasets. We run experiments on seven real datasets from the UCI repository [Dua and Graff, 2017] and four synthetic datasets from FCPS [Ultsch and Lötsch, 2020]. The datasets vary in size, number of features, and number of clusters, as presented in Table 2. For all datasets, we normalize the values of each feature in the range [0, 100] so that features with larger values do not dominate the pairwise distances.

Constraint Generation. We evaluate the performance of our approach for different number of constraints, relative to the dataset size. Specifically, for a given κ value with $0 \leq \kappa \leq \frac{|X|-1}{2}$, we generate a set of $\kappa \cdot |X|$ random clustering constraints following Wagstaff and Cardie [2000]: (1) We generate $\kappa \cdot |X|$ pairs of data points without repetition; (2) For each pair, we generate a ML constraint if both data point share the same ground-truth label and a CL constraint otherwise.

Evaluation. We evaluate the quality of the obtained clusterings based on ground-truth labels using two well-known external clustering evaluation metrics: the Adjusted Rand Index (ARI) [Hubert and Arabie, 1985] and the Normalized Mutual Information (NMI) [Strehl and Ghosh, 2002].

4.2 Baselines

Constrained Clustering (CC). We compare our approach with optimal constrained clustering formulation that is not restricted to conform to a decision tree. To do so, we remove the tree-related components of the encoding, namely the variables $[a_{t,j}]$, $[s_{i,t}]$, $[z_{i,t}]$, $[g_{t,c}]$ and the clauses in Eqs. (7)-(18). Instead, we introduce the clauses in Eq. (39) that guarantee a well-formed unary encoding of labels.[6] When used with the MD objective and $\epsilon = 0.0$, this baseline is equivalent to the maximum diameter constrained clustering formulations in previous works [Dao *et al.*, 2016; Dao *et al.*, 2017], i.e., it has the same set of (feasible and) optimal solutions. However, this baseline also supports ϵ-approximation and the [MD,MS] objective for a fair comparison with our approach.

$$(x_{i,c}, \neg x_{i,c+1}) \qquad \forall x_i \in X, c \in [1..k-2] \qquad (39)$$

Mixed Integer Optimization (MIO). To our knowledge, the only approach in the literature for constructing tree-based clustering using discrete optimization is the MIO model in Bertsimas *et al.* [2021]. While their model does not support clustering constraints, we can extend it to incorporate such constraints. However, Bertsimas *et al.* note that their model does not scale well and therefore do not present any experimental results and instead focus on a heuristic procedure that does not have any solution quality guarantees and cannot naturally support clustering constraints. Due to the relevance of the MIO model to our work, we have implemented the model using the Gurobi v10 solver and extended it to support clustering constraints (description of the extended MIO model appears in the extended version [Shati *et al.*, 2023]).

4.3 Results

For our first set of experimental results presented in Table 2, we solve the tree clustering problem for our datasets with different values of κ. We fix the value of approximation at $\epsilon = 0.1$. Consistent with previous work [Dao *et al.*, 2016; Babaki *et al.*, 2014], we set the solver time limit to 30 minutes. To avoid bias in results due to a specific set of constraints, we generate 20 random sets of constraints and report average values for the evaluation metrics and the runtime. Note that it might not be possible for a tree clustering with a fixed depth to satisfy all of the constraints, in particular in high-dimensional datasets with complex patterns where a large number of univariate splits may be needed to fit the ground-truth constraints. Thus, we also report the number of feasible runs (out of the 20 random constraint sets) while excluding infeasible and unknown[7] cases from the computed average ARI.

The results show that our approach can produce high quality interpretable solutions with non-zero ϵ values in the time limit. We observe that the [MD,MS] Pareto optimality objective leads to higher quality solutions compared to MD in all but a few cases without significant overhead in runtime.

Interestingly, we observe that tree clustering almost always leads to higher quality solutions compared to CC. This seems unintuitive since CC is strictly more expressive than tree clustering. We conjecture that both clustering objectives tend to perform better with tree clustering because of its inherently restricted solution space, resulting in potentially worse objective values but higher ARI scores.

Figure 1 highlights the benefit of both tree clustering (∘) and the Pareto objective (dashed). While the difference in ARI in unconstrained clustering ($\kappa = 0$) is negligible, as we increase the number of constraints we observe that tree-based clustering significantly outperforms CC and the Pareto objective tends to perform better than the MD objective when constraints are limited. Figure 1 also shows the percentage of feasible instances vs. κ (bars) and demonstrates that as problems become more constrained, it may be infeasible to find a tree

[6]Note that these clauses are redundant in the encoding of tree clustering as the condition is already enforced for the leaves.

[7]Instances where the solver timed out without finding a feasible solution are considered “unknown”.

Dataset	κ	[MD,MS]				MD			
		ARI	ARI (CC)	Feas.	Time (s)	ARI	ARI (CC)	Feas.	Time (s)
Iris	0.0	0.6	0.6	20	0.6	0.62	0.7	20	0.4
$\|X\| = 150$	0.1	0.83	0.78	20	0.7	0.71	0.55	20	0.4
$\|F\| = 4$	0.25	0.86	0.8	20	0.7	0.81	0.6	20	0.4
$K = 3$	0.5	0.91	0.8	20	0.8	0.88	0.63	20	0.5
$d = 3$	1.0	0.95	0.91	16	0.7	0.94	0.71	16	0.4
Wine	0.0	0	0	20	0.7	0.38	0.21	20	0.6
$\|X\| = 178$	0.1	0.69	0.53	20	1.2	0.41	0.19	20	0.6
$\|F\| = 13$	0.25	0.79	0.57	20	1.8	0.6	0.2	20	0.9
$K = 3$	0.5	0.82	0.5	20	6.9	0.72	0.2	20	5.4
$d = 3$	1.0	0.93	0.72	20	8.8	0.89	0.23	20	8
Glass	0.0	0.22	0.22	20	2.1	0.18	0.26	20	1.2
$\|X\| = 214$	0.1	0.19	0.1	20	9.3	0.16	0.11	20	3.3
$\|F\| = 9$	0.25	0.24	0.06	20	61.1	0.16	0.05	20	35.1
$K = 7$	0.5	0.26	0.07	19	954.1	0.24	0.01	20	624.9
$d = 4$	1.0	-	0.1	0	193.8	-	0.01	0	200.1
Ionosphere	0.0	0.01	0.01	20	2.4	0.16	0.08	20	2
$\|X\| = 351$	0.1	0.28	0.11	20	33.7	0.15	0.09	20	10.2
$\|F\| = 34$	0.25	0.5	0.22	11	598.9	0.48	0.12	11	880.9
$K = 2$	0.5	-	0.38	0	29.5	-	0.18	0	31.4
$d = 3$	1.0	-	0.78	0	5.8	-	0.76	0	5.7
Seeds	0.0	0.68	0.66	20	1.8	0.57	0.68	20	0.5
$\|X\| = 210$	0.1	0.68	0.64	20	1	0.67	0.54	20	0.5
$\|F\| = 7$	0.25	0.73	0.63	20	1.3	0.72	0.52	20	0.7
$K = 3$	0.5	0.78	0.64	14	1.5	0.78	0.52	14	1.2
$d = 3$	1.0	0.9	0.79	1	0.7	0.93	0.72	1	0.7
Libras	0.0	0.21	0.22	20	1802.8	0.2	0.16	20	1802.9
$\|X\| = 360$	0.1	0.17	0.17	20	1159.1	0.15	0.12	20	941.6
$\|F\| = 90$	0.25	0.17	0.14	20	1080	0.14	0.1	20	681.9
$K = 15$	0.5	0.18	0.11	20	866.2	0.14	0.07	20	305.1
$d = 5$	1.0	0.18	0.11	20	1518.3	0.14	0.06	20	859.4
Spam	0.0	0	0	20	38	-0.02	-0.01	20	41.5
$\|X\| = 4601$	0.1	-	0.04	0	358.9	-	0.05	0	379.4
$\|F\| = 57$	0.25	-	0.07	0	286.1	-	0.13	0	312.6
$K = 2$	0.5	-	0.08	0	151.1	-	0.25	0	152.9
$d = 3$	1.0	-	0.51	0	77.1	-	0.51	0	77.9
Lsun	0.0	0.44	0.39	20	1.2	0.39	0.39	20	0.8
$\|X\| = 400$	0.1	0.95	0.65	20	1	0.74	0.25	20	0.7
$\|F\| = 2$	0.25	1	0.94	20	1	0.89	0.25	20	0.6
$K = 3$	0.5	1	1	20	0.9	0.96	0.26	20	0.6
$d = 3$	1.0	1	1	20	0.9	0.98	0.48	20	0.6
Chainlink	0	0.12	1	20	3	0.11	0.06	20	2
$\|X\| = 1000$	0.1	0.89	1	17	5	0.84	0.01	17	5.5
$\|F\| = 3$	0.25	0.89	1	1	1.8	0.91	0.03	1	1.7
$K = 2$	0.5	-	1	0	1.3	-	0.05	0	1.2
$d = 3$	1.0	-	1	0	1	-	0.56	0	0.9
Target	0.0	0.36	0.36	20	5.7	0.33	0.33	20	3.7
$\|X\| = 770$	0.1	1	1	20	2.8	0.64	0.57	20	18.8
$\|F\| = 2$	0.25	1	1	20	2.7	0.87	0.45	20	9.9
$K = 6$	0.5	1	1	20	2.4	0.95	0.25	20	4.4
$d = 4$	1.0	1	1	20	2.2	0.99	0.45	20	1.9
WingNut	0.0	1	1	20	2.5	1	0.15	20	2
$\|X\| = 1016$	0.1	1	1	20	1.7	0.99	0.2	20	1.3
$\|F\| = 2$	0.25	1	1	20	1.6	0.99	0.28	20	1.2
$K = 2$	0.5	1	1	20	1.6	1	0.49	20	1.2
$d = 3$	1.0	1	1	20	1.6	1	0.82	20	1.2

Table 2: Interpretable constrained clustering with $\epsilon = 0.1$ averaged over 20 runs.

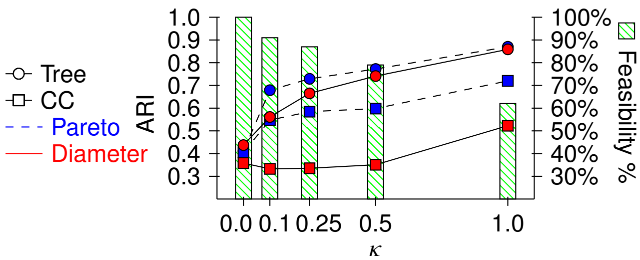

Figure 1: Results for $\epsilon = 0.1$ averaged over 20 runs and 11 datasets.

clustering of a given depth that satisfies all the constraints, exposing the trade-off between interpretability via decision trees and satisfaction of user-provided clustering constraints.

Next, we investigate how depth impacts the feasibility of finding a tree that satisfies the clustering constraints. Table 3 shows that deeper trees can resolve the infeasibility problem in some datasets, e.g., Glass and Ionosphere, however in Spam we find that even with a depth of four we are not able to find decision trees for any of the random constraint sets. Another interesting observation is that unnecessarily deep trees could lead to a lower score, demonstrating the potential benefit of restricted solution spaces induced by shallow trees. We observed similar trends for the NMI metric and provide the results in the extended version [Shati *et al.*, 2023].

Comparison with MIO. Our experiments with the MIO baseline found that it is unable to find a feasible solution for any of the datasets for the depths specified in Table 2 across multiple runs, both in constrained and unconstrained settings. This result is consistent with Bertsimas *et al.*'s [2021] observation on the limited scalability of their MIO formulation and emphasizes the strong performance of our approach.

Ablation Study. Finally, we study the impact of the smart pairs procedure (Section 3.2) and the ϵ-approximation (Section 3.1) on the performance of our approach. Unlike the ϵ-approximation, smart pairs is guaranteed to not change the set of feasible solutions and the set of optimal solutions. However, since there could be multiple optimal solutions, the different encoding may still lead to an optimal solution with a slightly different ARI score. The results presented in Table 4 show that the aforementioned methods can reduce the number of clauses and runtimes without meaningful decrease in score.

Dataset	d	ARI	Feas.	Time (s)
Iris	2	0.83	2	0.3
	3	0.91	20	0.8
	4	0.88	20	0.8
Wine	2	0.87	2	0.4
	3	0.82	20	6.9
	4	0.71	20	6.2
Glass	3	-	0	2.6
	4	0.26	19	953.6
	5	0.22	20	35.8
Ionosphere	2	-	0	0.8
	3	-	0	29.5
	4	0.69	18	731.7
Seeds	2	-	0	0.3
	3	0.78	14	1.5
	4	0.74	20	2.1
Libras	4	0.16	2	1801.4
	5	0.18	20	866.2
	6	0.17	20	315.4
Spam	2	-	0	35
	3	-	0	151.1
	4	-	0	975.6
Lsun	2	1	20	0.9
	3	1	20	0.9
	4	1	20	1.1
Chainlink	2	-	0	0.8
	3	-	0	1.3
	4	1	20	3.4
Target	3	-	0	1.8
	4	1	20	2.5
	5	1	20	4
WingNut	2	1	20	1.5
	3	1	20	1.7
	4	1	20	2.1

Table 3: Tree clustering with the [MD,MS] objective for different tree depths ($\kappa = 0.5$, $\epsilon = 0.1$) averaged over 20 runs.

Dataset	Setting	ARI	Time (s)	# Clauses
Libras	SP & ϵ=0.1	0.18	866.4	2,082,261.2
	ϵ=0.1	0.16	822.0	3,888,452.0
	ϵ=0.0	0.16	1197.1	4,140,872.0
Spam	SP & ϵ=0.1	Inf.	151.6	3,823,479.2
	ϵ=0.1	Inf.	332.7	24,980,546.4
	ϵ=0.0	Inf./Unk.	864.0	69,166,751.4
WingN	SP & ϵ=0.1	1.00	1.7	95,879.25
	ϵ=0.1	1.00	4.2	1,128,700.4
	ϵ=0.0	OOM†	98.3	3,449,740.4

†OOM indicates an out-of-memory error.

Table 4: Ablation study ([MD,MS]) averaged over 20 runs.

5 Conclusion

In this work, we present the first approach for *interpretable* and *constrained* clustering using decision trees. Specifically, we present a novel SAT-based encoding for constructing clustering trees that approximate two well-known clustering objectives. Our experiments on a range of real-world and synthetic datasets demonstrate the ability of our approach to produce high-quality and interpretable clustering solutions that incorporate user-provided clustering constraints.

Our work raises several interesting questions to investigate in future work. As there are potentially many Pareto-optimal solutions, investigating and empirically evaluating strategies or tools, e.g. multi-objective solvers [Jabs *et al.*, 2022], for exploring the Pareto front and selecting promising solutions is an interesting direction for future work. One of the challenges identified in this work is that highly-constrained problems in complex, high-dimensional datasets can become infeasible for a given tree depth. In future work, we would like to investigate strategies to overcome this challenge, e.g., by converting the hard clustering constraints into soft constraints that are encouraged rather than required to be satisfied.

Acknowledgments

We gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada CIFAR AI Chairs Program. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute for Artificial Intelligence (www.vectorinstitute.ai/partners). The final author would also like to thank the Schwartz Reisman Institute for Technology and Society for providing a rich multidisciplinary research environment.

References

[Babaki *et al.*, 2014] Behrouz Babaki, Tias Guns, and Siegfried Nijssen. Constrained clustering using column generation. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 438–454. Springer, 2014.

[Berg and Järvisalo, 2017] Jeremias Berg and Matti Järvisalo. Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability. *Artificial Intelligence*, 244:110–142, 2017.

[Berg *et al.*, 2019] Jeremias Berg, Emir Demirović, and Peter J Stuckey. Core-boosted linear search for incomplete MaxSAT. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, pages 39–56. Springer, 2019.

[Bertsimas *et al.*, 2021] Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. Interpretable clustering: an optimization approach. *Machine Learning*, 110(1):89–138, 2021.

[Biere *et al.*, 2009] Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.

[Bilenko *et al.*, 2004] Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11, 2004.

[Cohen *et al.*, 2020] Eldan Cohen, Arik Senderovich, and J Christopher Beck. An ising framework for constrained clustering on special purpose hardware. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 130–147. Springer, 2020.

[Dao *et al.*, 2016] Thi-Bich-Hanh Dao, Christel Vrain, Khanh-Chuong Duong, and Ian Davidson. A framework for actionable clustering using constraint programming. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, pages 453–461, 2016.

[Dao *et al.*, 2017] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain. Constrained clustering by constraint programming. *Artificial Intelligence*, 244:70–94, 2017.

[Davidson and Ravi, 2005] Ian Davidson and SS Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 138–149. SIAM, 2005.

[Davidson *et al.*, 2010] Ian Davidson, SS Ravi, and Leonid Shamis. A sat-based framework for efficient constrained clustering. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 94–105. SIAM, 2010.

[Dua and Graff, 2017] Dheeru Dua and Casey Graff. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2017. Accessed: 2022-12-01.

[Frost *et al.*, 2020] Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. Exkmc: Expanding explainable k-means clustering. *arXiv preprint arXiv:2006.02399*, 2020.

[Gamlath *et al.*, 2021] Buddhima Gamlath, Xinrui Jia, Adam Polak, and Ola Svensson. Nearly-tight and oblivious algorithms for explainable clustering. *Advances in Neural Information Processing Systems*, 34:28929–28939, 2021.

[Hubert and Arabie, 1985] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[Ignatiev *et al.*, 2021] Alexey Ignatiev, Joao Marques-Silva, Nina Narodytska, and Peter J Stuckey. Reasoning-based learning of interpretable ml models. In *IJCAI*, pages 4458–4465, 2021.

[Jabs *et al.*, 2022] Christoph Jabs, Jeremias Berg, Andreas Niskanen, and Matti Järvisalo. Maxsat-based bi-objective boolean optimization. In *25th International Conference on Theory and Applications of Satisfiability Testing (SAT 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[Liu and Fu, 2015] Hongfu Liu and Yun Fu. Clustering with partition level side information. In *2015 IEEE international conference on data mining*, pages 877–882. IEEE, 2015.

[Liu *et al.*, 2017] Hongfu Liu, Zhiqiang Tao, and Yun Fu. Partition level constrained clustering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2469–2483, 2017.

[Moshkovitz *et al.*, 2020] Michal Moshkovitz, Sanjoy Dasgupta, Cyrus Rashtchian, and Nave Frost. Explainable k-means and k-medians clustering. In *International Conference on Machine Learning*, pages 7055–7065. PMLR, 2020.

[Pelleg and Baras, 2007] Dan Pelleg and Dorit Baras. K-means with large and noisy constraint sets. In *European Conference on Machine Learning*, pages 674–682. Springer, 2007.

[Shati *et al.*, 2021] Pouya Shati, Eldan Cohen, and Sheila McIlraith. Sat-based approach for learning optimal decision trees with non-binary features. In *27th International Conference on Principles and Practice of Constraint Programming (CP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[Shati *et al.*, 2023] Pouya Shati, Eldan Cohen, and Sheila McIlraith. Optimal decision trees for interpretable clustering with constraints (extended version). *arXiv preprint arXiv:2301.12671*, 2023.

[Strehl and Ghosh, 2002] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[Thalamuthu *et al.*, 2006] Anbupalam Thalamuthu, Indranil Mukhopadhyay, Xiaojing Zheng, and George C Tseng. Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405–2412, 2006.

[Ultsch and Lötsch, 2020] Alfred Ultsch and Jörn Lötsch. The fundamental clustering and projection suite (fcps): A dataset collection to test the performance of clustering and data projection algorithms. *Data*, 5(1):13, 2020.

[Wagstaff and Cardie, 2000] Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110. Citeseer, 2000.

[Wagstaff *et al.*, 2001] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.

[Wu and Chou, 2011] Roung-Shiunn Wu and Po-Hsuan Chou. Customer segmentation of multiple category data in e-commerce using a soft-clustering approach. *Electronic Commerce Research and Applications*, 10(3):331–341, 2011.