Balancing the Tradeoff Between Clustering Value and Interpretability

Sandhya Saisubramanian*
University of Massachusetts Amherst
saisubramanian@cs.umass.edu

Sainyam Galhotra*
University of Massachusetts Amherst
sainyam@cs.umass.edu

Shlomo Zilberstein
University of Massachusetts Amherst
shlomo@cs.umass.edu

ABSTRACT

Graph clustering groups entities – the vertices of a graph – based on their similarity, typically using a complex distance function over a large number of features. Successful integration of clustering approaches in automated decision-support systems hinges on the interpretability of the resulting clusters. This paper addresses the problem of generating interpretable clusters, given features of interest that signify interpretability to an end-user, by optimizing interpretability in addition to common clustering objectives. We propose a β-interpretable clustering algorithm that ensures that at least β fraction of nodes in each cluster share the same feature value. The tunable parameter β is user-specified. We also present a more efficient algorithm for scenarios with $\beta = 1$ and analyze the theoretical guarantees of the two algorithms. Finally, we empirically demonstrate the benefits of our approaches in generating interpretable clusters using four real-world datasets. The interpretability of the clusters is complemented by generating simple explanations denoting the feature values of the nodes in the clusters, using frequent pattern mining.

KEYWORDS

Centroid-based clustering, Interpretability

ACM Reference Format:
Sandhya Saisubramanian, Sainyam Galhotra, and Shlomo Zilberstein. 2020. Balancing the Tradeoff Between Clustering Value and Interpretability. In *2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES'20), February 7–8, 2020, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3375627.3375843

1 INTRODUCTION

Graph clustering is increasingly used as an integral part of automated decision support systems for high-stake applications such as infrastructure development [18], criminal justice [3], and health care [16]. Such domains are characterized by high-dimensional data and the goal of clustering is to group these nodes, typically based on similarity over all the features [20]. The solution quality of the resulting clusters is measured by the objective value. As the number of features increases, it is increasingly difficult for an end-user to interpret the resulting clusters.

For example, consider the problem of clustering districts in Kenya to aid decision-making for infrastructure development (Figure 1), sanitation in particular [4, 19]. Each district is described by features denoting the population, access to basic sanitation, gender and age demographics, and location. The districts in a cluster are typically considered to be indistinguishable and hence may be assigned the same development policies. The similarity of districts for clustering is measured based on all the features. As a result, it is likely that the cluster composition is heterogeneous with respect to the sanitation feature (Figure 1(a)). This may significantly affect the decision-maker's ability to infer meaningful patterns, especially due to lack of ground truth, thereby affecting their policy decisions.

Recently, there has been growing interest in interpretable machine learning models [10, 22, 24], mostly focusing on explainable predictive models or interpretable neural networks. There is limited prior research, if any, on improving the interpretability of clusters [6, 8]. Clustering results are *expected* to be inherently interpretable as the aim of clustering is to group similar nodes together. However, when clustering with a large number of features, interpretability may be diminished since no clear patterns may be easy to recognize for an end-user, as in Figure 1(a).

Interpretability of the clusters is critical in high-impact domains since decision makers need to understand the solution beyond how the data is grouped into clusters: what characterizes a cluster and how it is different from other clusters. Additionally, the ability of a decision maker to evaluate the system for fairness and identify when to trust the system hinges on the interpretability of the results. In this work, the interpretability of clusters is measured based on the *homogeneity* of nodes in each cluster, with respect to certain predefined feature values of interest (FoI) in the data to the end-user.

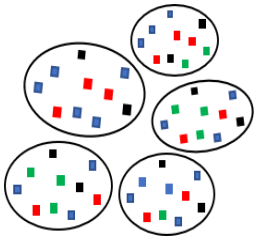
Solution quality of the clusters, denoted by the objective value, and interpretability are often *competing* objectives. For example in Figure 1(b), interpretability is optimized in isolation by partitioning the nodes only based on FoI, which significantly affects the solution quality and optimizing for solution quality affects interpretability (Figure 1(a)). Reliable decision support requires interpretable clusters, without significantly compromising the solution quality.

In this paper, we study the problem of optimizing for interpretability of clusters, in addition to optimizing the solution quality of centroid-based clustering algorithms such as k-center. We propose a β-interpretable clustering algorithm that generates clusters such that at least β fraction of nodes in each cluster share the same feature value, with respect to FoI. The β value is a user-specified input. By adjusting the value of β, the homogeneity of the nodes in the cluster with respect to FoI can be altered, thus facilitating balancing the trade-off between solution quality and interpretability

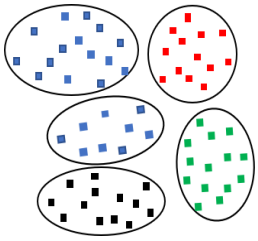
*Both authors contributed equally to this research.

AIES '20, February 7–8, 2020, New York, NY, USA

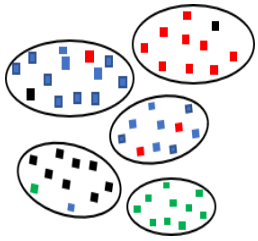
ACM ISBN 978-1-4503-7110-0/20/02…$15.00
https://doi.org/10.1145/3375627.3375843



(a) KC: k-center over all features



(b) P_F: Partitions based on FOI



(c) β-KC: Optimized KC & interpretability ($\beta = 0.8$)

Approach	K-center Objective Value	Cluster Explanation
KC	66,468.72	C1: 50-75% access to pit latrines ∨ 75-100% access to pit latrines C2: 25-50% access to pit latrines C3: 50-75% access to pit latrines C4: 50-75% access to pit latrines ∨ 75-100% access to pit latrines C5: 0-25% access to pit latrines
P_F	442,531.68	C1: 0-25% access to pit latrines C2: 25-50% access to pit latrines C3: 50-75% access to pit latrines C4: 75-100% access to pit latrines C5: 75-100% access to pit latrines
β-KC	86,218.93	C1: 0-25% access to pit latrines C2: 25-50% access to pit latrines C3: 50-75% access to pit latrines C4: 50-75% access to pit latrines C5: 75-100% access to pit latrines

(d) Objective values and corresponding explanations

Figure 1: Illustrative example using the Kenya sanitation data and k=5. The vertices of the graph are the districts with features describing population, sanitation, gender and age demographics, and location. The population % in each district with pit latrine access is divided into: {0-25%, 25-50%, 50-75%, 75-100%}, indicated by different colors. Interpretability is measured based on access to this basic sanitation. Dominant features in a cluster are generated as labels using frequent pattern mining.

(Figure 1(c)). We then present a more efficient algorithm to specifically handle settings with $\beta = 1$ and bound the loss in solution quality of centroid-based clustering objectives, when optimizing for interpretability.

While interpretable clusters are a minimal requirement, it may not be sufficient to guarantee interpretability of the system, due to the cognitive overload for users in understanding the results. Hence, the resulting clusters are complemented by logical combinations of cluster labels as explanations. The feature values of the nodes in the cluster, with respect to FoI, are generated as cluster labels, using frequent pattern mining. In Figure 1(d), traditional clustering produces longer explanations, which are generally undesirable [10], and optimizing for interpretability produces concise explanations. Thus, generating interpretable clusters is crucial for generating concise and useful explanations.

Our primary contributions are: (i) formalizing the problem of interpretable clustering that optimizes for interpretability, in addition to solution quality (Section 2); (ii) presenting two algorithms to achieve interpretable clustering and analyzing their theoretical guarantees (Section 3); and (iii) empirical evaluation of our approaches using four real-world datasets and using frequent pattern mining to generate cluster explanations (Section 4). Our experiments demonstrate the efficiency of our approaches in balancing the trade-off between interpretability and solution quality. The results also show that clusters with different levels of interpretability can be generated by varying β.

2 PROBLEM FORMULATION

Let $V = \{v_1, v_2, \ldots, v_n\}$ denote a set of n nodes, along with a pairwise distance metric $d : V \times V \to \mathbb{R}$. Let $A^* = \{A_1^*, \ldots, A_m^*\}$ denote the set of values of m features where A_i^* refers to the set of values for the i-th feature, $F^* = \bigtimes_{1 \le i \le m} A_i^*$ and $\Psi : V \to F^*$ denote the mapping from nodes to the feature values. Let $H = G(V, d)$ be a graph where $d : V \times V \to [0, \infty)$ is a metric over V. Given a graph instance H and an integer k, the goal is to partition V into k disjoint subsets by optimizing an objective function, which results in clusters $C = \{C_1, C_2, \ldots, C_k\}$. The objective function (o), for a graph H and a set of clusters C, returns an objective value as a real number, $o(H, C) \to \mathbb{R}$, which helps compare different clustering techniques. The optimal objective value of an objective function o is denoted by OPT_o. $C(u)$ denotes the cluster to which the node $u \in V$ is assigned.

The clusters produced by the existing algorithms are often non-trivial and non-intuitive to understand for an end-user due to the complex feature space. Let $A \subseteq A^*$ denote the set of features in A^* that signify *interpretability* for the user, $F = \bigtimes_{a \in A} a$ denoting the feature values of interest (FoI). In Figure 1, A is the sanitation feature and F ={0-25%, 25-50%, 50-75%, 75-100%}, denoting the four feature values of access to basic sanitation.

Quantifying Interpretability: Interpretability score of a cluster C with respect to a feature value $f \in F$ is denoted by $I_f(C)$ and estimated based on the fraction of the nodes in the cluster that share the feature value, $\forall f \in F$:

$$I_f(C) = \frac{\sum_{v \in C} [v_f]}{|C|}$$

with $[v_f]$ denoting whether the node v satisfies feature value f and $|C| = \sum_{v \in V} [C(v) = C]$ denoting the total number of nodes in the cluster. Hence, $I_f(C) \in [0, 1]$. Given F, the interpretability score of a cluster, $I_F(C) \in (0, 1]$, is calculated as

$$I_F(C) = \max_{f \in F} I_f(C).$$

Definition 1. *The interpretability score of a clustering C, given F, is denoted by $\mathcal{I}_F(C)$ and is calculated as:*

$$\mathcal{I}_F(C) = \min_{C \in C} I_F(C).$$

Problem Statement: Given F, we aim to create clusters that maximize the interpretability score, $\mathcal{I}_F(C)$, while simultaneously optimizing for solution quality using centroid-based clustering objectives such as k-center. k-center clustering aims to identify k nodes as centers (say S, $|S| = k$) and assign each node to the closest cluster center ensuring that the maximum distance of any node from its cluster center is minimized. The objective value is calculated as:

$$o_{kC}(H, C) = \max_{v \in V} \min_{s \in S} d(u, s)$$

Definition 2. *A clustering C is **β-interpretable**, given F, if $\mathcal{I}_F(C) \ge \beta$. That is, each cluster is composed of at least β fraction of nodes that share the same feature value.*

Definition 3. *A clustering C is **strongly** interpretable, given F, if $\mathcal{I}_F(C) = 1$.*

We now analyze the maximum achievable interpretability for a given dataset and identify the upper bound on β.

2.1 Optimal upper bound of β

Let β_{max} denote the optimal upper bound of β. Without loss of generality, given a feature value $f \in F$, we assume that there exists at least one node $u \in V$ that satisfies f. When $|F| \leq k$, with k denoting the number of clusters, a clustering C can be generated such that $\mathcal{I}_F(C) = 1$. This is achieved by constructing each cluster with the nodes that satisfy the same feature value f, and hence $\beta_{max} = 1$.

However, when $|F| > k$, there exists no clustering C with $\mathcal{I}_F(C) = 1$, since the optimal solution cannot form clusters with nodes satisfying only one feature value of interest. Hence, in such cases, $\beta_{max} < 1$. The optimal value for this case can be estimated as follows: consider the top-k features based on frequency of occurrence in the data and assign the nodes that refer to each of these features to a different clusters. All the remaining unassigned nodes are then iteratively assigned to the cluster with maximum interpretability score. If multiple clusters have the same interpretability score, the new node is added to the cluster with larger size, since it is less likely to negatively affect the interpretability score.

In general, the interpretability score of a cluster C is dominated by the feature value satisfied by maximum number of nodes within C. For a given cluster C, interpretability can be boosted by either adding more nodes of the majority feature or removing the nodes that are different from the majority feature. If all the nodes that do not represent the majority feature are removed, the interpretability score of C is 1. Using this intuition, we propose algorithms to generate β-interpretable cluster, when $\beta \leq \beta_{max}$.

3 SOLUTION APPROACH

In many applications, the clusters that are considerably homogeneous but not strongly interpretable may still be acceptable since a few outliers do not affect the decision maker's abilities to infer a pattern. For example, if the nodes in a cluster are 90% homogeneous, the interpretability may not be significantly affected. However, this may help with improving the solution quality of the clusters formed using centroid-based objectives. To that end, we propose an algorithm (Algorithm 1) in which the homogeneity of the nodes in a cluster can be adjusted using a tunable parameter β. The algorithm identifies β interpretable clusters for all values of $\beta \leq \beta_{max}$. We present the algorithms using k-center as the clustering objective. However, it is straightforward to extend the algorithms to any other centroid-based clustering.

The input to Algorithm 1 is a graph $G(V, E)$, the parameters k and β, referring to the number of clusters needed and the interpretability score requirement. First, it initializes a collection of k clusters, C with the greedy k-center algorithm and optimizes the quality of clusters generated. In order to improve interpretability score, our algorithm iteratively identifies a cluster $C \in C$ with the least interpretability score and then post-processes it to improve its interpretability scores without considerable loss in the k-center objective. While processing C, a feature value $f \in F$ associated with maximum number of nodes in C is identified as the 'majority' feature value along with a set S corresponding to the collection of nodes that share the majority value. To boost the score of C, the fraction of nodes that share the majority feature needs to be increased. We employ the following two operations for this purpose:

- The total number of nodes with majority feature are increased (`boost_majority`); and
- The nodes that do not correspond to the majority feature value in C are removed from C and re-assigned to other clusters (`reduce_minority`).

Algorithm 1 β-Interpretability

Input: $G(V, E, d)$, # clusters k, β, Feature set $F \equiv \{f_1, \ldots, f_{|F|}\}$
Output: Clusters $C \equiv \{C_1, \ldots, C_k\}$
1: $C \leftarrow$ k-center(V, k)
2: **while** $\mathcal{I}_F(C) < \beta$ **do**
3: $\quad C \leftarrow \arg\min \mathcal{I}_F(C)$
4: $\quad$ majority$\leftarrow \arg\max_{f_i \in F} V_{F_i} \cap C$
5: $\quad S \leftarrow V_{majority} \cap C$
6: $\quad$ **if** $\exists v \in V_{majority} \setminus S$ **then**
7: $\quad\quad$ boost_majority$(C, C, majority)$
8: $\quad$ **else**
9: $\quad\quad$ reduce_minority(C, C, S)
10: **return** C

Algorithm 2 `boost_majority`

Input: C, C, majority
Output: C_1, C_2
1: $C' \leftarrow$ Identify closest cluster based on majority nodes
2: $C_1, C_2 \leftarrow identify\_toptwo(C \cup C')$
3: $R \leftarrow C \setminus (C_1 \cup C_2)$
4: $R_1, R_2 \leftarrow partition(R, |C_1||R|/\theta, |C_2||R|/\theta)$, where $\theta = |C_1| + |C_2|$
5: $C_1 \leftarrow C_1 \cup R_1, C_2 \leftarrow C_2 \cup R_2$
6: **return** C_1, C_2

Algorithm 3 `reduce_minority`

Input: C, C, S
Output: C
1: $\gamma \leftarrow |C| - \frac{|S|}{\beta}$
2: $u \leftarrow find\_center(S)$
3: $X \leftarrow identify\_farthest(C, \gamma, u, S)$
4: Greedily assign all $v \in X$ to the closest cluster $C' \in C$, ensuring $I_F(C') \geq \beta$
5: **return** C

`boost_majority`. Outlined in Algorithm 2, this subroutine iterates over the clusters $C \in C$ to identify the closest cluster C' that contains the nodes with the 'majority feature' and merges C with C' (Line 1,2). It then identifies two different features that have the maximum frequency within the merged cluster and assigns these features to two different clusters C_1 and C_2 (Line 2). The remaining nodes in the merged cluster are assigned to either of the two clusters such that C_1 and C_2 have comparable interpretability scores (Line 4,5).

`reduce_minority`. This subroutine, outlined in Algorithm 3, identifies the collection of nodes within C that do not have the 'majority' feature, which when removed help boost the interpretability score of C (Line 1). Nodes which do not belong to the majority feature and are farthest from the center u are considered for re-assignment (Lines 2,3). Each of farthest node $v \in X$ is then assigned to clusters $C' \in C$, considered in increasing order of distance from v such that the interpretability score of C' does not reduce below β (Line 4). This process of removing nodes from C is performed only when C has the maximum number of nodes present in the data set that share the majority feature.

Algorithm 4 `strong-interpretability`

Input: $G(V, E, d)$, # clusters k, Feature set $F \equiv \{f_1, \ldots, f_{|F|}\}$
Output: Clusters $C \equiv \{C_1, \ldots, C_k\}$
1: $s_1 \ldots s_t \leftarrow 0$
2: $\mathcal{S} = \{(s_1, \ldots, s_{|F|}) : s_i > 0, \sum s_i = k\}$
3: **for** $(s_1 \ldots s_t) \in \mathcal{S}$ **do**
4: $\quad C_{(s_1 \ldots s_{|F|})} \leftarrow$ Uk-center(V_{f_i}, s_i)
5: $C \leftarrow identify\_min\_obj(C_{(s_1 \ldots s_{|F|})}, \forall (s_1 \ldots s_{|F|}) \in \mathcal{S})$
6: **return** C

Remark 4. *In some cases, Algorithm 1 may converge to a local maxima and may not reach β_{max}, when the input $\beta = \beta_{max}$. This happens when the feature value being boosted is not one of the feature values in the optimal solution. However, we observe that this is a rare scenario in practice. A detailed algorithm that works in these cases is described in the full version of the paper [25].*

For cases in which the minimum distance pair identified in Algorithm 2 belong to same optimal cluster, we bound the loss in k-center objective when using `boost_majority`.

Lemma 5. *In each iteration of* `boost_majority` *where the minimum distance pair identified in Algorithm 2 belong to same optimal cluster,, the k-center objective value worsens by $\alpha OPT_{kC,IC}$, where $\alpha \leq 10$ and $OPT_{kC,IC}$ denotes the optimal k-center objective value of the clusters that achieve maximum interpretability.*

Proof in full version [25].

When generating clusters with $\beta = 1$, Algorithm 1 may take long to converge, especially if the initial k-center based clusters have poor interpretability. We propose a more efficient algorithm for strong interpretability that solves the interpretable clustering problem on each individual features to construct the final solution.

3.1 Strong interpretability, $\beta = 1$

Algorithm 4 is a more efficient approach to handle scenarios with $\beta = 1$. At a high-level, it identifies the distribution of feature values among k clusters and then quickly generates the clusters. It leverages the property that a clustering C with $\beta = 1$ is characterized by clusters such that all nodes in a cluster share the same feature value. As discussed earlier, $\beta = 1$ is achievable only when $|F| \leq k$ and this is an important assumption required for this algorithm.

The first step is to identify a set $\mathcal{S}$ which consists of a $|F|-$tuple of values that sum up to k (Line 2). This set identifies all possible distributions of the different feature values under consideration for interpretability (FoI) among the k clusters. For each value $(s_1, \ldots, s_{|F|}) \in \mathcal{S}$, it identifies s_i clusters for nodes with feature f. The collection of these k clusters refer to the solution corresponding $(s_1, \ldots, s_{|F|})$ (Lines 3-5). This step generates $|\mathcal{S}|$ collection of k-clusters and the one with minimum k-center objective value is chosen as the final set of clusters (Line 6).

Algorithm 4 is capable of generating clusters with high interpretability, without significant loss in the clustering objective value. We now show that the final solution returned by our algorithm is a 2-approximation of the optimal algorithm that generates interpretable clusters and optimizes for the k-center objective.

Lemma 6. *The* `strong-interpretability` *clustering algorithm generates C such that $\mathcal{I}_F(C) = 1$ and $o_{kc}(H, C) = 2OPT_{IC,kC}$, where o_{kC} refers to the k-center of objective of C and $OPT_{IC,kC}$ denotes the optimal k-center objective value of clusters that achieve maximum interpretability.*

Proof Sketch. Since each cluster $C \in C$ contains all nodes that share the same feature value, $\mathcal{I}_F(C) = 1$. Additionally, the optimal solution $OPT_{IC,kC}$ has a distribution of features $(s_1, \ldots, s_{|F|}) \in \mathcal{S}$. The solution $C_{(s_1,\ldots,s_{|F|})}$ is a 2-approximation of $OPT_{IC,kC}$ (following the proof of 2-approximation of greedy algorithm for k-center). Since, the final solution chooses C that minimizes the k-center objective over all possible clustering in $\mathcal{S}$, it is guaranteed that C is a 2-approximation of $OPT_{IC,kC}$. □

4 EXPERIMENTAL RESULTS

We evaluate the efficiency of our approaches based on two metrics: interpretability score of the clustering and the objective value of k-center algorithm. We refer to Algorithm 1 as β-IC and Algorithm 4 as *IKC*.

Baselines The results are compared with that of three baselines: 1. k-center clustering over all the features in the data (*KC*); 2. partitioning the dataset into k clusters based on the FoI (P_F); and 3. k-center clustering over only the features of interest for interpretability (KC_F). *KC* and P_F represent extremes of the spectrum, optimizing only for k-center objective or interpretability. KC_F aims to optimize for the distances, ensuring that the nodes with similar features are present close to each other.

Datasets The algorithms are evaluated using four datasets: 1. Kenya sanitation data in which the interpretability is defined over % population in a district with access to basic sanitation; 2. Kenya traffic accidents data[1], whose interpretability is measured based on the accident type; 3. Adult dataset[2] in which the interpretability of the clusters is defined based on the age and income of the population; and 4. Crime data[3], with FoI as the median family income of the communities.

Setup All algorithms are implemented in Python and tested on an Intel i7 computer with 8GB of RAM. In the interest of clarity, we experiment with $|F| = 4$ for all domains. Due to randomness in the k-center algorithm, the clustering objective behavior of our techniques may not be monotonic. For any given $k = \theta$, we run the algorithm for different values $k = \theta' \leq \theta$ and choose the best clustering returned.

4.1 Solution quality vs. cluster interpretability

We first study the trade-off between the k-center objective value and and the interpretability score of the clusters. We vary $\beta \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ for Algorithm 1 and compare the results with that of the baselines and that of Algorithm 4, with fixed $k = 5$. The results in Figure 2 show how the k-center objective value may be affected as we form increasingly interpretable clusters using our algorithms. We do not distinguish between the performances with various β values, denoted by the purple markers, since the goal is to understand how the algorithm balances the trade-off for any β value. We also do not consider $\beta < 0.5$ since that defeats the

[1] https://www.opendata.go.ke/datasets/2011-traffic-incidences-from-desinventar
[2] https://archive.ics.uci.edu/ml/machine-learning-databases/adult/
[3] http://archive.ics.uci.edu/ml/datasets/communities+and+crime

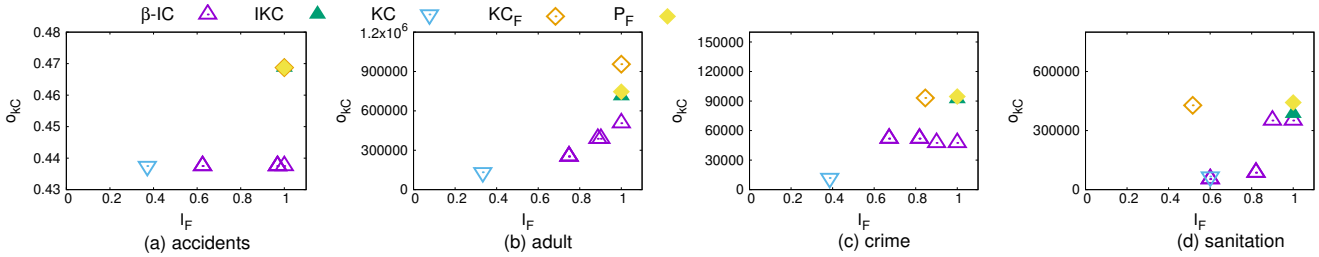


Figure 2: k-center objective value o_{kC} versus interpretability score I_F.

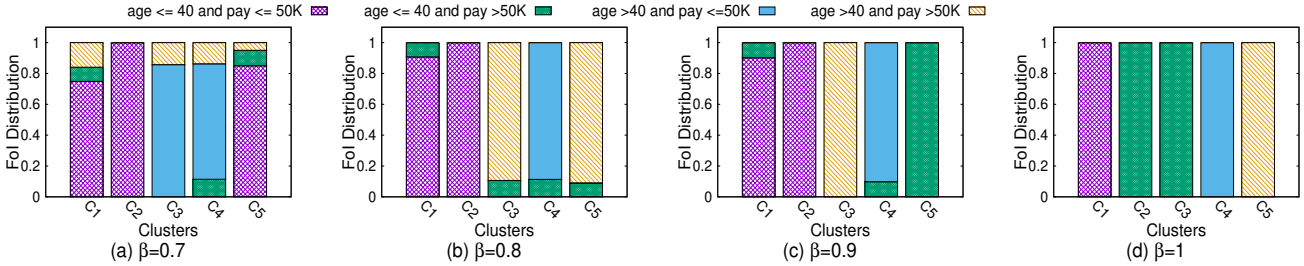


Figure 3: Distribution of FoI across clusters with varying β on Adult dataset.

purpose of optimizing for interpretability. Note that our algorithm supports any value of $\beta \in [0, 1]$ as input.

Approaches that minimize k-center objective and maximize the interpretability, **lower right** corner of the figure, are desirable. Overall, the baselines either achieve high interpretability with poor k-center objective or low k-center objective with a very low interpretability. Our approach has a better balance between them since the clusters generated by our algorithm have high interpretability, without significant loss in k-center objective. With the increase in β values, the k-center objective worsens but the loss in k-center objective is not high and is within a factor of 5 in most cases. The runtime of *KC* is at most 40 seconds across all datasets and the runtime of our approach is at most 65 seconds across all datasets and all values of β. This shows that there is no significant overhead in optimizing for both interpretability and solution quality.

4.2 Effect of varying β

As discussed above, it is evident that our approaches efficiently balance the trade-off even for higher values of β. We now study the effects of varying β on the cluster composition. Figure 3 shows the distribution of FoI in each cluster for different values of β for the Adult dataset. In the interest of readability, we do not include results for lower values of β. With the increase in β, the fraction of majority feature in each cluster grows. For example, the nodes represented by yellow color are merged as β is increased from 0.7 to 0.8. Similarly, when β is increased from 0.8 to 0.9, the green colored feature is a minority, which are merged to form a new cluster. Notice that all the green feature nodes are not merged and this process stops as soon as the clusters reach interpretability of 0.9. However, in the case of strong-interpretability with $\beta = 1$, the clusters are homogeneous. In our experiments, the runtime with $\beta = 1$ is at most twice as that of $\beta = 0.5$ and the runtime of *IKC* is consistently lower than β-IC for $\beta = 1$. Similar trends were observed for other domains and the results are available in the full version of the paper [25].

4.3 Effect of varying k

To ensure that the trends in the relative performances of the approaches in minimizing the k-center objective are consistent, we experiment with varying the number of clusters k, and with fixed $\beta = 1$. Figure 4 plots the results of the approaches for k varying from 10 to 50. As expected, the k-center objective value decreases with the increase in k and the relative behavior of all the techniques is consistent. Our techniques β-IC and IKC are close to KC across all datasets, while P_F works well for accidents and sanitation datasets only. All techniques run in less than three hours for all datasets and for all values of k in the experiments, demonstrating the scalability of our approach.

4.4 Interpretability and Explanation Generation

The interpretability of the resulting clusters can be further improved by generating explanations based on the feature values of the nodes in the clusters. Concise and correct explanations based on FoI are possible only when the clusters are homogeneous with respect to FoI. Hence, generating explanations also allows us to understand and compare the performance of different techniques beyond the interpretability scores.

We generate explanations as logical combinations of the feature values of FoI associated with the nodes in each cluster, using frequent pattern mining [14]. This is implemented using Python `pyfpgrowth` package with a minimum support value of 20% of the cluster size. That is, this approach lists all the feature values that are associated with at least 20% of the nodes in the cluster and 20%

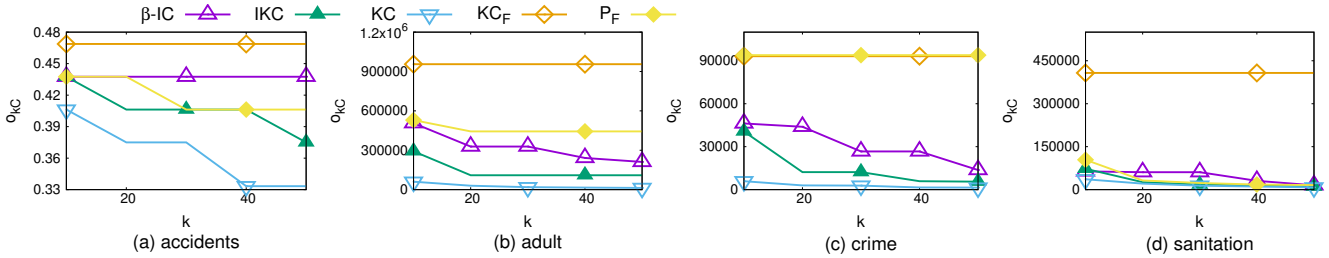


Figure 4: Comparing o_{kC} of various techniques with varying k.

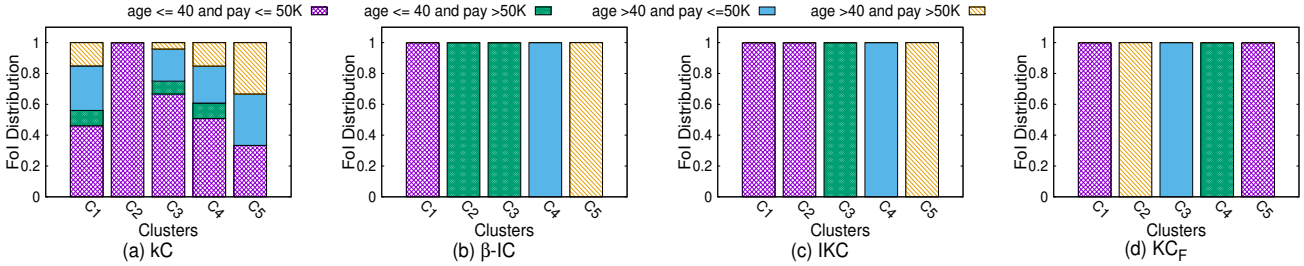


Figure 5: Distribution of features of interest (FoI) across clusters for Adult dataset.

is the tolerance for outliers in the cluster. This value can be adjusted depending on the application. Explanations are then generated by a logical OR over these feature values. Figure 5 shows the distribution of explanations across clusters for different techniques on the Adult data, with $\beta = 1$. Clusters generated by the *KC* approach contain a skewed distribution of features across all clusters and are hard to interpret, with respect to FoI. Approaches that focus on interpretability have generated homogeneous clusters with majority of the nodes in a cluster sharing the same feature value. As a result, the generated explanations for these approaches are concise and fairly different across the clusters, thereby improving the interpretability for the decision maker.

5 RELATED WORK

Interpretable machine learning The two main threads of research in interpretable machine learning are generating explanations for black-box models [1, 12, 13, 17, 22] and improving the transparency with interpretable models [7, 10, 24]. Most of these approaches have been developed for predictive models or for interpretable neural networks and have heavily relied on domain-dependent notions of interpretability [10]. We define a domain-independent notion of interpretability and aim to form interpretable clusters, which is critical for high-impact applications [24]. We argue that generating explanations for clustering requires homogeneous clusters and propose algorithms that improve the interpretability without compromising on the solution quality.

Clustering with multiple objectives Prior research on clustering focuses heavily on improving the performance metrics [2, 20, 27], such as accuracy, scalability and runtime, but neglect the interpretability aspect. Another thread of work employs soft clustering methods [8, 11] or mixed integer optimization [6] to improve interpretability but do not provide any solution guarantees. Constrained clustering [26], in which the pairs of nodes that must belong to the same cluster are enforced as constraints, cannot be used to generate interpretable clusters when $\beta < 1$. Another related body of work is the research on multi-objective clustering [5, 9, 15, 21, 23] that has been predominantly applied for specific applications and recently for improving fairness. Extending these approaches to our setting is not straightforward since the algorithms are problem-specific. There is limited research on interpretable clustering [8] since clusters are expected to be interpretable as they group similar nodes, which is not necessarily the case when dealing with high-dimensional data.

6 CONCLUSION

We address the challenge of generating interpretable clustering, while simultaneously optimizing for solution quality of the resulting clusters. We propose an algorithm to generate β-interpretable clusters, given β and the features of interest that signify interpretability to the user. A more efficient algorithm specifically to handle scenarios with $\beta = 1$ is also presented, along with the theoretical guarantees of the two approaches. Our approaches efficiently balance the trade-off between interpretability and solution quality, compared to the baselines. The proposed approach can be extended to handle continuous FoI by treating each interval of continuous values as a discrete value for β-interpretability.

We currently target settings in which clustering is performed using centroid-based algorithms. In the future, we aim to expand the range of clustering objectives considered, including hierarchical clustering, and analyze their theoretical guarantees. Using interpretable clustering to identify bias in decision-making is another interesting direction for future research.

REFERENCES

[1] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the CHI conference on human factors in computing systems*. ACM, 582.

[2] Charu C. Aggarwal and Haixun Wang. 2010. A survey of clustering algorithms for graph data. In *Managing and mining graph data*. Springer, 275–301.

[3] Turki Aljrees, Daming Shi, David Windridge, and William Wong. 2016. Criminal Pattern Identification Based on Modified K-means Clustering. In *IEEE International Conference on Machine Learning and Cybernetics (ICMLC)*, Vol. 2. 799–806.

[4] ICT Authority. 2017. Kenya Sanitation by District. https://www.opendata.go.ke/datasets/sanitation-by-district. (2017).

[5] Suman K Bera, Deeparnab Chakrabarty, and Maryam Negahbani. 2019. Fair algorithms for clustering. *arXiv preprint arXiv:1901.02393* (2019).

[6] Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. 2018. Interpretable clustering via optimal trees. *arXiv preprint arXiv:1812.00539* (2018).

[7] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. 2018. This looks like that: deep learning for interpretable image recognition. *arXiv preprint arXiv:1806.10574* (2018).

[8] Junxiang Chen, Yale Chang, Brian Hobbs, Peter Castaldi, Michael Cho, Edwin Silverman, and Jennifer Dy. 2016. Interpretable Clustering Via Discriminative Rectangle Mixture Model. In *IEEE 16th International Conference on Data Mining (ICDM)*. 823–828.

[9] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*.

[10] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

[11] Derek Greene and Pádraig Cunningham. 2005. Producing Accurate Interpretable Clusters from High-Dimensional Data. In *European Conference on Principles of Data Mining and Knowledge Discovery*.

[12] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2019. A survey of methods for explaining black box models. *ACM computing surveys* 51, 5 (2019), 93.

[13] David Gunning. 2017. Explainable Artificial Intelligence. *Defense Advanced Research Projects Agency (DARPA)* (2017).

[14] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. 2007. Frequent Pattern Mining: Current Status and Future Directions. *Data Mining and Knowledge Discovery* 15 (2007), 55–86.

[15] Julia Handl and Joshua Knowles. 2007. An Evolutionary Approach to Multiobjective Clustering. *IEEE Transactions on Evolutionary Computation* 11, 1 (2007), 56–76.

[16] Ramzi A. Haraty, Mohamad Dimishkieh, and Mehedi Masud. 2015. An Enhanced K-means Clustering Algorithm for Pattern Discovery in Healthcare Data. *International Journal of Distributed Sensor Networks* (2015).

[17] Andreas Holzinger. 2018. From machine learning to explainable AI. In *IEEE World Symposium on Digital Intelligence for Systems and Machines*. 55–66.

[18] Gert-Jan Hospers, Pierre Desrochers, and Frédéric Sautet. 2009. The Next Silicon Valley? On the Relationship Between Geographical Clustering and Public Policy. *International Entrepreneurship and Management Journal* 5, 3 (2009), 285–299.

[19] Kenya ICT. 2017. Kenya Vision 2030: A Globally Competitive and Prosperous Kenya . https://www.opendata.go.ke/datasets/vision-2030. (2017).

[20] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. 1999. Data Clustering: A Review. *Comput. Surveys* 31, 3 (1999), 264–323.

[21] Rachsuda Jiamthapthaksin, Christoph F. Eick, and Ricardo Vilalta. 2009. A framework for multi-objective clustering and its application to co-location mining. In *Proceedings of the International Conference on Advanced Data Mining and Applications*.

[22] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. Faithful and Customizable Explanations of Black Box Models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*.

[23] Martin H.C. Law, Alexander P. Topchy, and Anil K. Jain. 2004. Multiobjective data clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[24] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206.

[25] Sandhya Saisubramanian, Sainyam Galhotra, and Shlomo Zilberstein. 2019. Balancing the Tradeoff Between Clustering Value and Interpretability - full version. https://arxiv.org/abs/1912.07820. (2019).

[26] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*.

[27] Rui Xu and Donald Wunsch. 2005. Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks* 16 (2005), 645–678.