Fair Clustering with Over-Representation Constraint

Camilo Cedeno-Tobon, Kevin Ko, Simran Shinh, Irina Wang

Abstract

The foundation of this algorithm is the k-clustering problem, in which there is a set of points D to be separated into k clusters. The clustering without over-representation algorithm introduces a fairness parameter, such that each point has a "color", and the representation of each color in any cluster is constrained. In other words, it prevents any one color from dominating a cluster. In the real world, color can be any feature we want to protect from over-representation, such as gender, race, party-affiliation, etc.

In this paper, we improve upon the $3-$approximation fair k-center clustering algorithm developed by Ahmadian et al [1]. The algorithm finds a fractional solution using a linear program (LP), which is then rounded to an integral one by selecting a subset of the fractional facilities to open and rerouting all assignments to these facilities, violating the fairness constraint by an additive factor of at most 2. Thereafter, we extend the algorithm to the fair k-medoid clustering objective and define a $(b+1)-$approximation, where b is determined from the optimal LP solution.

1 Introduction

In recent years, machine learning has become an increasingly relevant decision-making tool, with automatic prediction and forecasting methods utilized for resource allocation, mortgage and credit applications, etc. It is thus crucial to design machine learning algorithms without bias against or towards particular groups. In fact, color-blind clustering, which does not take a protected attribute into its decision making, may still result in very unfair clusterings, which gives rise to the need for explicit fair clustering algorithms where the representation fraction is bound.

Fair clustering, introduced by Chierichetti et al. [2], imposes an additional constraint on the standard notion of clustering, such that the fraction of each color (ie, male or female when the protected attribute is gender) in any cluster matches its distribution in the overall population. The protected attribute is constrained to be binary, however, and the strict representation requirement may be inflexible for practical purposes. Thereafter, the notion of fairness has been extended to the more general case, where multiple colors (> 2) are allowed, and the representation of each color in any cluster is upper bounded by a given fraction, constant for all colors. This is the notion we adopt.

1.1 Related Works

Chierichetti et al [2] developed approximation algorithms for fair k-center and k-median with two colors. They introduced the notion of fairlet decomposition: a partitioning of the input D into small subsets, called fairlets, such that a balanced clustering can be obtained by merging fairlets into clusters. Building on that work, Backurs et al. [5] defined a more scalable algorithm for the k-means objective. Rösner and Schmidt [3] generalized the notion to multiple colors, and developed a $14-$approximation algorithm for k-center. Bercea et al. [4] generalized the notion further to the fair representation with a flat fractional upper bound for all colors. They defined approximation algorithms for k-center, k-median and also k-means, through the combination of a fair fractional LP solution and any integer solution of the general (unfair) clustering. This approach is entirely different from the original fairlet decomposition method, and is extended by Ahmadian et al [1] for a closer look at the k-center objective and a 3-approximation bound.

1.2 Our Contributions

We first outline and adjust the algorithm described by Ahmadian et al [1], giving an alternative approach to certain steps, so as to, with certainty, maintain or improve the computational upper bound of the k-center objective.

We then further extend the algorithm by implementing it for the k-medoid problem, in which

the objective is to minimize the total distance between points and their cluster centers. We derive an upper bound on the objective to the fair k-medoid problem by leveraging the optimal LP solution. Our algorithm differs from the one provided by Bercea et al. [4], which requires computing both a fair LP solution and an integer solution to the general k-median clustering problem.

We applied the algorithms to a variety of datasets in order to assess their performance. In particular, we tested the algorithm on an artificial dataset, a New York City census[1] dataset, and the `reuters`[2] dataset, which contains 2500 English language texts and was used by the original authors to empirically evaluate their results.

1.3 Outline

In section 2 we formalize the α-capped k-center algorithm and present our theoretical results. In section 3 we extend the algorithm to the k-mediod problem, and again present our theoretical findings. In section 4 we present empirical evaluations of our algorithms, with comparisons to the performance of Ahmadian et al [1] for k-center. In section 5 we present our conclusions, and in section 6, we outline further extensions.

2 k-Center Model

2.1 Important Terminology

In the k-clustering problem, the input is a set D of points in a metric space with the distance function $d(\cdot, \cdot)$, for which we use euclidean distance, and an integer bound k. The goal is to cluster the points into at most k clusters $C_1, C_2, \cdots, C_k$. For clarity, we refer to points as *clients* and chosen centers as *facilities*. A feasible k-center solution is denoted (F, σ), where $F \in D$ is the chosen subset of facilities, and σ denotes the mapping $\sigma : D \to F$ from clients to their assigned facilities. The objective value $\lambda(F, \sigma)$ for k-center is the maximum assignment cost, $\max_{j \in D} d(j, \sigma(j))$.

For the fairness constraint, there is an additional parameter $\alpha \in (0, 1]$, where the fraction of clients of any color in any cluster must be $\leq \frac{1}{\alpha}$ of the size of the cluster. This is called the α-capped k-center problem. For this purpose, we give each client a color $c(j)$, and define D_c to be the set of clients with color c.

[1] https://www.kaggle.com/muonneutrino/new-york-city-census-data/datanyc_census_tracts.csv

[2] archive.ics.uci.edu/ml/datasets/Reuter_50_50

2.2 High Level Description

Given a true optimal solution $\lambda(F^*, \sigma^*)$, a ρ-approximation algorithm is one such that $\lambda(F, \sigma) \leq \rho\lambda(F^*, \sigma^*)$. Ahmadian et al. [1] gives a 3-approximation algorithm with an additive violation of the fairness constraint of at most 2 for each color in each cluster. This is a *bicriteria* algorithm, where the second criteria is the upper bound on the violations. We define an improvement for this algorithm that, while not improving the constant-factor approximation bound of 3, gives a better upper bound specific to the inputs to the problem.

We first solve an LP formulation of the problem, which would give a fractional assignment of clients to facilities. Thereafter, we choose a subset of at most size k of the fractional facilities, and reroute all assignments to these chosen facilities. Lastly, we formulate a max flow network using the rerouted client assignments, which may still be fractional, such that by the integrality property of maximum flows with integer capacities, we obtain an integer solution.

2.3 LP Formulation [3]

The LP formulation of the problem is shown below, where the clustering cost is at most λ. As seen below, we restrict this through constraint (9), where assignments to a facility greater than λ away is not allowed.

$$\sum_{i \in F} x_{ij} \geq 1 \qquad \forall j \in D \tag{1}$$

$$x_{ij} \leq y_i \qquad \forall i \in F, j \in D \tag{2}$$

$$y_i \leq x_{ii} \qquad \forall i \in F \tag{3}$$

$$\sum_{j \in D_c} x_{ij} \leq \alpha \sum_{j \in D} x_{ij} \qquad \forall c \in [t], i \in F \tag{4}$$

$$\sum_{j \in D} x_{ij} \geq \left\lceil \frac{1}{\alpha} \right\rceil y_i \qquad \forall i \in F \tag{5}$$

$$\sum_{i \in F} y_i \leq k \tag{6}$$

$$0 \leq y_i \leq 1 \qquad \forall i \in F \tag{7}$$

$$0 \leq x_{ij} \leq 1 \qquad \forall i \in F, j \in D \tag{8}$$

$$x_{ij} = 0 \qquad \forall i \in F, j \in D, d(i, j) > \lambda \tag{9}$$

[3] The algorithms described in this section are adjusted versions of the ones given by Ahmadian et al. [1]

Constraint (4) is the representation constraint, where the total assignment of clients of a color c to a center i must be below α times the total assignment of clients to the center. Constraint (5) requires that a center must serve at least $\lceil \frac{1}{\alpha} \rceil$ clients in order for the α bound to hold. The LP will return a set of solutions denoted (x, y), the fractional facility and client assignments such that the maximum distance is below λ.

Instead of determining the λ through running the LP, we pre-determine a λ value and attempt to find a feasible assignment. To determine an initial λ value, we first implement the greedy algorithm for choosing cluster centers, and assign all clients to the closest center, regardless of color.

Algorithm 1 Greedy k-center

$i_0 \leftarrow$ an arbitrary client in D.
$F' = \{i_0\}$
for $l \in \{1, 2, ..., k-1\}$ **do**
    $i_l \leftarrow \arg\, max_{j \in D} min_{i \in F'} d(j, i)$, the furthest client from F'
    $\forall j \in D : \sigma(j) \leftarrow i = \arg\, min_{i \in F'} d(i, j)$
    $\lambda' \leftarrow \lambda(F', \sigma)$
    return $((F', \sigma), \lambda')$

We lower bound the λ value for the LP by $\frac{\lambda'}{2}$, where λ' is given by the greedy algorithm. While no feasible solution is found, we slightly increment this λ and rerun the LP.

2.4 Opening k integer facilities

Having solved the LP relaxation and obtaining a fractional assignment (x, y), we must find the subset of facilities that will be open in the optimal solution where all facilities are integral. Let the set of facilities returned by the LP be $F' = \{i \in F | y_i > 0\}$. We then obtain a subset $F'' \subseteq F'$, by first opening the maximal subset of facilities where all facilities are at least 2λ from each other. More specifically, this applies for $i, i' \in F''$, $d(i, i') \geq 2\lambda$. Recall that λ is the upper bound on the objective value of the LP. This will choose $\leq k$ facilities, since they are all more than 2λ away from each other, and thus cannot be serviced by the same facility given that all assignments are within λ. There was a feasible LP solution with the given λ, so there must be less than k facilities found through this λ.

Next, since opening additional cluster centers should only help the solution, if the number of cluster centers is less than k in the previous step, we decrease the λ value used. This will increase the possible number of facilities chosen. As such, we decrease λ to λ'', where λ'' is the lowest value such that the total number of facilities in F'' is still less than or equal to k. We open all facilities in F'', which sets all $y'_i = 1$ for $i \in F''$ and $y'_i = 0$ for $i \notin F''$.

We then transfer all client assignments to facilities $i' \in F''$. First, for each $i \in F'$, we define a mapping θ to a $i' \in F''$. In essence, for each of the fractional facilities we decide not to choose, we find its closest chosen facility. Since F'' is defined as the maximum subset of facilities where all facilities are at least $2\lambda''$ from one another, we know that the distance between i' and its closest chosen facility must be $d(i', \theta(i')) < 2\lambda''$, or else i' would also be a chosen facility.

Algorithm 2 Pairing closest facilities

for $i \in F'$ **do**
    if $i \in F''$ **then**
        $\theta(i) = i$
    else
        $\theta(i) = i'$ where $i' \in F''$ with $d(i, i') < 2\lambda''$

Now for each $i' \in F'$, we then direct all assignments $x_{i'j}$, for each client $j \in D$, to be assigned to $\theta(i')$ instead.

$$x'_{ij} = \begin{cases} \sum_{i' \in \theta^{-1}(i)} x_{i'j} & \text{if } i \in F'' \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

This mapping preserves the α-cap representation constraint, since for each facility $i' \in F'$ in the LP solution, the α-cap representation constraint is satisfied. Additionally, when rerouting each i', we reroute its entire fair set of assignments assignments to $\theta(i') \in F''$, so the final assignment for each $i \in F''$ is a combination of fair assignments. Thus, the final assignment remains a fair assignment.

We can define an upper bound for the k-center objective given the new assignments (x', y'). The upper bound would be the maximum possible rerouted distance, $d(j, \theta(i'))$, where j was originally assigned to $i' \in F'$. We know, by construction, that $d(i', \theta(i')) \leq 2\lambda''$ for $i' \in F'$, and $d(i', j) \leq \lambda$, given that it is an assignment in the LP, which has maximum distance λ. Thus, by triangle inequality, $d(j, \theta(i')) < d(i', j) + d(i', \theta(i')) \leq \lambda + 2\lambda''$. This is the new upper bound for the fair k-center fractional solution.

Our upper bound differs from the one given

by Ahmadian et al [1], which is 3λ. While this gives a constant factor approximation, it is higher than $\lambda + 2\lambda''$, since $\lambda'' \leq \lambda$. We show in later sections the computational and visualized difference.

2.5 Rounding to integer assignments

While we have created an upper bound for the solution (x', y'), where the x' values are still fractional, to finish the algorithm, we still need to construct a completely integral solution (x'', y'') with the same upper bound. This is done through constructing a maximum flow network. We can utilize the fact that if a network has integral bounds on edges and integral demands, then it always has a feasible solution. The flow network (V, A) is constructed as follows:

- $V = \{s, t\} \cup D \cup \{(i, c) | i \in F'', c \in [t]\}$
- $A = A_1 \cup A_2 \cup A_3 \cup A_4$ where
  - $A_1 = \{(s, j) | j \in D\}$ with capacity 1
  - $A_2 = \{(j, (i, c)) | j \in D_c, x'_{ij} > 0\}$ with capacity 1
  - $A_3 = \{((i, c), i)\}$ with lower bound $\left\lfloor \sum_{j \in D_c} x'_{ij} \right\rfloor$ and capacity $\left\lceil \sum_{j \in D_c} x'_{ij} \right\rceil$
  - $A_4 = \{(i, t)\}$ with lower bound $\left\lfloor \sum_{j \in D} x'_{ij} \right\rfloor$ and capacity $\left\lceil \sum_{j \in D} x'_{ij} \right\rceil$

Since (x', y') is feasible for a flow of value $|D|$, an integral flow (x'', y'') of value $|D|$ must exist, where x'' is the final flow on the A_2 arcs. We now show that for any facility $i \in F''$ and color $c \in [t]$, $\sum_{j \in D_c} x''_{ij} \leq \alpha \sum_{j \in D} x''_{ij} + 2$. This implies a maximum of 2 violations of the α-cap constraint for each color and facility.

From the A_2 arcs, we see that each client j could possibly be assigned to any facility i where $x'_{ij} > 0$ in the fractional solution. For the integral flow, one such facility i is chosen. From the A_3 arcs, we see that for each color and facility, the allowed assignment is bound between the floor and ceiling of $\sum_{j \in D_c} x'_{ij}$, which is the value of the fractional assignment. Thus, the integer assignment will differ by less than 1 from the fractional assignment for each color for each cluster. From the A_4 arcs, we see that for each facility, the allowed assignment is bound between the floor and ceiling of $\sum_{j \in D} x'_{ij}$, so the integer assignment will again differ by less than 1 from the fractional assignment.

Combining this, we know that

$$\sum_{j \in D_c} x''_{ij} \leq \sum_{j \in D_c} x'_{ij} + 1 \leq \alpha \sum_{j \in D} x'_{ij} + 1 \tag{11}$$

$$\alpha \sum_{j \in D} x'_{ij} + 1 \leq \alpha (\sum_{j \in D} x''_{ij} + 1) + 1 \leq \alpha \sum_{j \in D} x''_{ij} + 2 \tag{12}$$

$$\sum_{j \in D_c} x''_{ij} \leq \alpha \sum_{j \in D} x''_{ij} + 2 \tag{13}$$

This completes our algorithm.

3 Extension to k-Medoid

3.1 Overview

We now create an extended model that solves the k-medoid problem instead of the k-center problem. While the LP for the k-center problem attempts to minimize λ, our new LP associated with the k-medoid problem aims to directly minimize the total cost of distances between points and their assigned facilities. This requires a change to the previously given LP, where constraint (9) is deleted, and an objective function $\min \sum_{j \in D} \sum_{i \in F} x_{ij} d(i, j)$ is added.

While the new objective is no longer concerned with λ, we use a similar method as in the previous section to find the optimal subset of facilities after solving the LP relaxation, such that we can readily define an upper bound on the objective value. Upon determining the set of facilities to open, we reroute assignments as before, and again utilize maximum flow to arrive at an integer solution that only minimally violates the α constraint.

3.2 The k-Medoid Algorithm

Let (x, y) be the fractional assignment obtained from the adjusted k-medoid LP. We define λ to be the maximum assignment distance of any client j to a facility $i \in F'$ chosen by the LP. Using 2λ as a threshold, we again choose the maximal subset $F'' \subseteq F'$ of facilities where all $i, i' \in F''$ has $d(i, i') \geq 2\lambda$. As in the analogous step for k-center, this will select $\leq k$ facilities. We then increment λ to λ'' until the number of facilities chosen is maximized while still upper bounded by k.

Next, we run Algorithm 2 on $i \in F'$, and reroute all assignments based on (10). We have then obtained a new set of assignments (x', y'), where x' are fractional and y' is integral. This mapping

preserves the α-cap representation constraint, since the rerouting is identical to the one for k-center - the combination of multiple fair assignments is sill a fair assignment.

Lastly, we construct the maximum flow network as before, then reduce it to the equivalent minimum cost-max flow network, with costs $d(i,j)$ for the arcs in A_2, and cost of 0 for all other arcs. With this minimum cost network, we will be able to obtain an integral assignment where equation (13) still holds, and the overall cost is minimized, such that the objective value of the integer solution $(x'', y'') \leq$ the objective of (x', y'). This holds because the fractional flow is a feasible solution of the minimum cost network, thus its objective value must be $\geq$ than the optimal solution, for which there is an integer solution by the integrality property of the max flow.

3.3 Defining an upper bound

While the algorithm itself is very similar to that of the k-center objective, defining an upper bound for the k-medoid objective is a more involved process. The LP value, denoted C^{LP}, is

$$C^{LP} = \sum_{j \in D} \sum_{i \in F'} x_{ij} d(i,j) \tag{14}$$

The objective value after rerouting is

$$\hat{C} = \sum_{j \in D} \sum_{i \in F'} \sum_{i' \in \theta(i)} x_{i'j} d(i,j) \tag{15}$$

We know that for each client j rerouted from $i' \in F'$ to $i \in F''$, by the triangle inequality, the new distance $d(i,j) \leq d(i',j) + d(i,i')$. If the client is not rerouted, this also holds, since then $i' = i$ and $d(i,i') = 0$.

$$\hat{C} \leq \sum_{j \in D} \sum_{i \in F'} \sum_{i' \in \theta(i)} x_{i'j} (d(i',j) + d(i,i')) \tag{16}$$

We notice that the summation of $x_{i'j} d(i',j)$ is equal to C^{LP}. We can upper bound the objective by

$$\hat{C} \leq C^{LP} + \sum_{i \in R} \sum_{j \in D} \sum_{i' \in \theta(i)} x_{i'j} d(i,i') \tag{17}$$

$$\leq C^{LP} + \sum_{i \in F'} \sum_{i' \in \theta(i)} d(i,i') \sum_{j \in D} x_{i'j} \tag{18}$$

It then follows that we need to upper bound $\sum_{i \in F'} \sum_{i' \in \theta(i)} d(i,i') \sum_{j \in D} x_{i'j}$. For each rerouted facility $i' \in F' \setminus F''$, let $W_{i'} = \sum_{j \in D} x_{i'j}$, the total "weight" of assignments to that facility, and $C_{i'} = \sum_{j \in D} x_{i'j} d(i',j)$, the total contribution to the objective value from that facility. We want to find an upper bound of $d(i,i') \cdot W_{i'}$ in terms of $C_{i'}$, such that we can upper bound it by C^{LP}. Since by our construction, $d(i,i') \leq 2\lambda''$, we thus find the lowest number b such that

$$b \cdot C_{i'} \geq d(i,i') W_{i'} \tag{19}$$

$$b \geq 2\lambda'' \max_{i' \in F' \setminus F''} \frac{W_{i'}}{C_{i'}} \tag{20}$$

Summing over all facilities, we then have:

$$\sum_{i \in F'} \sum_{i' \in \theta(i)} d(i,i') \sum_{j \in D} x_{i'j} \leq \sum_{i \in F'} \sum_{i' \in \theta(i)} b \cdot C_{i'} \tag{21}$$

$$\leq b \sum_{i \in F'} \sum_{i' \in \theta(i)} \sum_{j \in D} x_{i'j} d(i',j) \tag{22}$$

$$\leq b \cdot C^{LP} \tag{23}$$

Combining with equation (18), an upper bound for the rerouted objective is then $\hat{C} \leq (b+1) C^{LP}$. This upper bound is specific to the LP solution returned, and is not a constant-factor approximation for all instances of the k-medoid problem.

4 Empirical Evaluation

4.1 Dataset Overview

We assessed the performance of our implementation of the k-center algorithm and k-medoid extension on a myriad of datasets, and we compared the results to the classical k-means clustering problem to better depict the effect of the over-representation constraint on data with protected features.

The synthetically generated dataset was created using Scikit-Learn's sample generator. We generated a 2-dimensional dataset with 4 facilities and 300 datapoints. Each point is assigned one of 4 colors.

The `reuters`[4] dataset was created by transforming 2500 English language texts by 50 authors into 10-dimensional vectors using Gensim's Doc2Vec package, and this is the same methodology described by the Ahmadian et al [1]. In this scenario, the text's author is the protected value or color.

[4] archive.ics.uci.edu/ml/datasets/Reuter_50_50

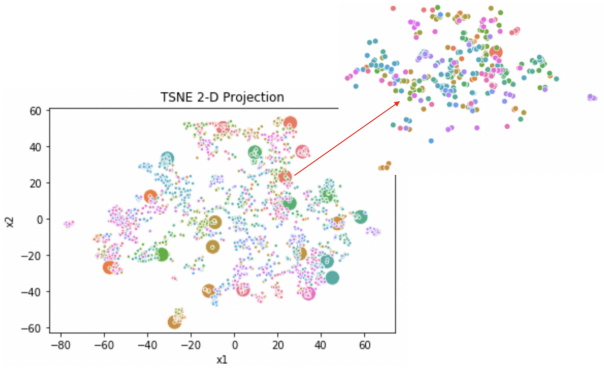
The New York City Census[5] dataset was created by joining two Kaggle datasets. The first dataset contained latitude and longitude for each city block code, and the second data contained city tract demographic data for each city. We used city blocks as data points and mapped each city block to its longitude and latitude. Colors were assigned by determining the majority race for each city block.

4.2 Reuters Dataset

For the Reuters data and k-center objective, we initially set $k = 25$ and $\alpha = 0.5$, to first examine the significant case where no cluster has one significantly dominating author. We found that $\lambda = 15$ is the minimum value that produces a feasible solution, given computational constraints. The LP runs in time quadratic to the number of data points, which is too high to fine tune the λ value.

After running the k-center algorithm, we used Scikit-Learn's T-SNE embedding to project the 10-dimensional document vectors to two dimensions. TSNE is a dimensionality reduction technique similar to PCA. The plot after the transformation is shown in Figure 1.

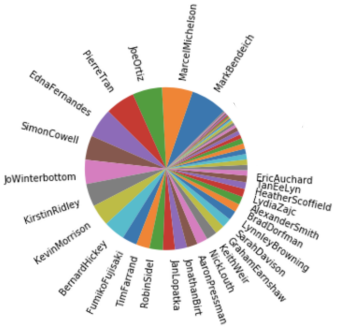
Figure 1: 2D projection of reuters data



The larger data points in the figure above correspond to the data points that are designated as facilities. If we examine an individual cluster, we see the distribution of authors that were assigned to that cluster. This is depicted in Figure 2.

[5]https://www.kaggle.com/muonneutrino/new-york-city-census-data/

Figure 2: Distribution of authors



We can discern that no author is over-represented in this cluster, as desired. This is a characteristic of each cluster obtained by the k-center fair clustering algorithm. Comparing the cost (objective value) of our results to the ones from Ahmadian et al [1] with $\lambda = 15$ and the following α values, we see that our new algorithm gives a consistently lower and tighter upper bound and lower actual objective value, as expected.

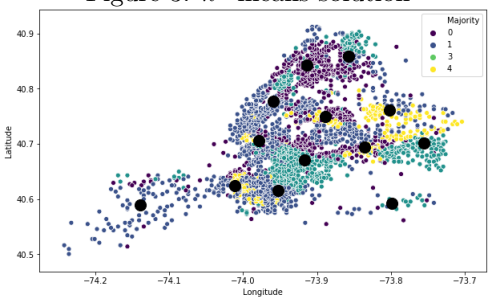
α	3λ	Original Cost	$\lambda + 2\lambda''$	Our Cost
0.5	45	36.21	30.4	27.54
0.25	45	34.8	30.5	26.46
0.1	45	35.44	31	28.89

4.3 New York City Census Dataset

We applied our algorithm to the New York City Census dataset in order to assess its performance on a dataset with real world implications and features that generally need to be protected, such as race, gender, age, and socioeconomic status. In this setting, we considered city blocks and assigned them the protected attribute of race majority. In applying the fair $k-$center clustering algorithm to the data, we expected to cluster New York City such that each cluster contains no more than a third of any city block of a certain racial majority.

As a baseline, we first used the classical $k-$means clustering to illustrate the distribution of points within clusters when we do not control for over-representation. This result is depicted in Figure 3.

Figure 3: $k-$means solution



As detailed before, each data point is a city block, plotted by its longitude and latitude. The points together comprise a map of New York City. Each number/color pair correspond to a racial category, and each point is assigned a color based on the racial category that comprises the majority of that city block. In Figure 4, 0 maps to the category of Hispanic, 1 maps to White, 2 maps to Native American, 3 maps to Black, and 4 maps to Asian. 2 is omitted from the legend as Native Americans do not represent the majority of any city block.

Running our implementation of $k-$center clustering on the same dataset with parameter $\alpha = 0.333$ yields many new facility assignments, as shown in Figure 4.

Figure 4: Fair $k-$center solution

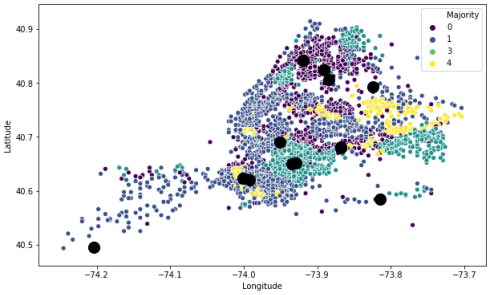
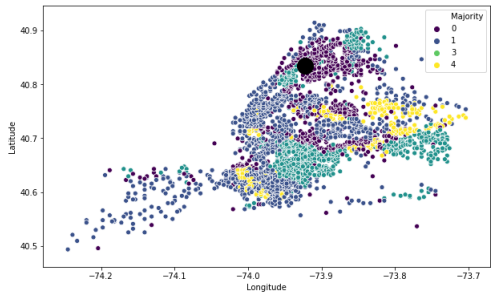


Figure 5: Non-Maximized $k-$center



The set of facilities depicted in Figure 4 are the facilities that our algorithm determines after maximizing the number of facilities chosen. Utilizing the original algorithm by Ahmadian et all [1], we arrive at a single cluster, as shown in Figure 5.

For other α values, we also compare ours and the original algorithm, and observe that our results, after maximizing the subset chosen, are much tighter.

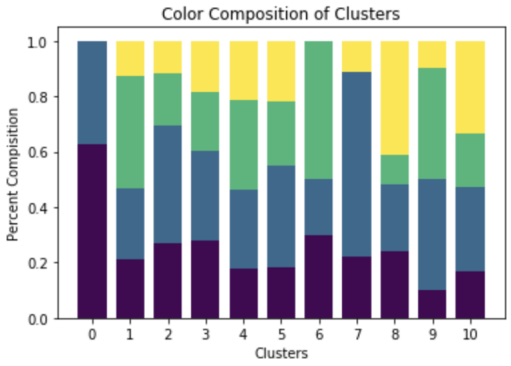
α	3λ	Original Cost	$\lambda + 2\lambda''$	Our Cost
0.5	0.27	0.205	0.127	0.120
0.33	0.27	0.222	0.134	0.122
0.25	0.27	0.202	0.12	0.1173
0.1	0.3	0.2472	0.123	0.1147

4.4 Synthetic Dataset

4.4.1 k-center

We generated 300 random points of 2-dimensions by using Scikit-Learn's sample generator. Since the size and dimensions of this dataset are small, it is useful in testing our theoretical results. We first conducted a test of the k-center fair clustering algorithms using $\alpha = 0.25$ and $k = 25$. Figure 3 depicts the color composition of each cluster that is output by the k-center fair clustering algorithm.

Figure 6: Color composition of fair clusters



The α-cap constraint is slightly violated for some clusters and colors, but not beyond an additive factor of 1. For clusters that do not contain many data points, this additive factor has a large impact, but for larger clusters, the impact is minimal. We again compare the cost differences between our algorithm and the original algorithm, for multiple α and λ values, and the same pattern as before follows.

α	3λ	Original Cost	$\lambda + 2\lambda''$	Our Cost
0.5	2.4	1.89	1.57	1.34
0.33	2.4	1.96	1.57	1.40
0.25	2.4	1.96	1.57	1.40
0.1	2.7	2.31	1.42	1.34

4.4.2 k-medoid

As the LP for k-medoid does not scale well for large datasets, given the lack of the λ constraint, we test it only on the synthetic data set. We set $k = 25$, and compare our upper bound with the actual cost.

α	λ''	C^{LP}	b	$(b+1)C^{LP}$	$\hat{C}$
0.5	0.3	95.42	2.82	364.78	119.25
0.33	0.315	95.42	3.07	388.73	109.77
0.25	0.31	95.42	2.86	368.03	109.28
0.1	0.3045	98.21	3.57	448.72	114.88

We see that for this data-set, b is around 3, and below 4, so the $(b + 1)$-approximation algorithm is around a 5-approximation. It would be very useful to observe a pattern for larger data-sets, but was not computationally possible with our limitations.

5 Conclusion

Fair clustering and the elimination of bias in large datasets has become increasingly crucial for extracting meaningful information from problems of interest. With machine learning models becoming more ubiquitous, it is imperative to verify that your models are neither favoring nor debilitating any group or individuals. In particular, clustering is used in a myriad of settings, from detecting fake news to creating targeted ads, and its applications underscore the importance of ensuring that objectiveness is maintained. The purpose of our paper is to add to the body of work exploring this domain.

A particular application that we examined in our paper was clustering neighborhoods in New York City based on location while keeping race as a protected feature. Our experimental study on this dataset allowed us to create groups with fair representation, and this is something that is not intrinsic to the original $k-$center algorithm.

Our study revisited the application of the $k-$center clustering problem and improved upon this model by generating a new, tighter upper bound specific to the solution returned by the LP relaxation. We then extended the application to $k-$medoid, where we again generated an upper bound from the objective value of the LP relaxation. We tested our models on both synthetic datasets and real-world datasets, yielding extensive results that were compared based on objective function value and upper bound tightness.

6 Future Work

The k-center clustering problem is NP-Hard which means that solving for the optimal solution is extremely computationally expensive. We construct an approximation algorithm that runs in polynomial time which makes solving for a good solution possible. However, our algorithm will take a long time to terminate on large datasets because it does not scale linearly with the size of the dataset. As a result, there is future work to be done in improving the algorithm's efficiency so that it can be applied to larger, higher dimensional datasets without requiring an infeasible amount of time to solve the corresponding LP.

7 References

[1] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. "Clustering without Over-Representation." Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining - KDD 19, 2019, doi:10.1145/3292500.3330987.

[2] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair clustering through fairlets. In NIPS. 5029–5037.

[3] Clemens Rosner and Melanie Schmidt. 2018. Privacy preserving clustering with constraints. In ICALP. 96:1–96:14.

[4] Ioana O. Bercea, Martin Gross, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R. Schmidt, and Melanie Schmidt. 2018. On the cost of essentially fair clusterings. Technical Report 1811.10319. arXiv.

[5] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. 2019. Scalable Fair Clustering. In ICML.