Fair Clustering Through Fairlets

(EE531 Final Project - Fairness)

F. Chierichetti[1] R. Kumar[2] S. Lattanzi[2] S. Vassilvitskii[2]

[1]Dipartimento di Informatica, Sapienza University
[2]Google Research

Appeared at NIPS 2017

Table of Contents

Outline

Introduction

- In this work, the notion of *disparate impact* will be followed!

Introduction

- In this work, the notion of *disparate impact* will be followed!
- Disparate impact: protected attributes should not be explicitly used in making decisions, *and the decisions made* should not be disproportionately different for applicants in different protected classes.

Introduction

- In this work, the notion of *disparate impact* will be followed!
- Disparate impact: protected attributes should not be explicitly used in making decisions, *and the decisions made* should not be disproportionately different for applicants in different protected classes.
- In case of clustering problem, disparate impact translates to that of color balance in each cluster, as we will see from now on.

Outline

k-clustering

Definition

Let (M, d) be a metric space, equipped with the metric function d. Given a set of points $X \subset M$, a *k-clustering* of X is a partition of X into k disjoint subsets, $C_1, \ldots, C_k$, called *clusters*.

k-clustering

Definition

Let (M, d) be a metric space, equipped with the metric function d. Given a set of points $X \subset M$, a *k-clustering* of X is a partition of X into k disjoint subsets, $C_1, \ldots, C_k$, called *clusters*.

Alternate Formulation

A *k-clustering* of X is an *assignment function*, $\alpha : X \to [k]$.
Each cluster C_i is the preimage of i under α i.e. $C_i = \alpha^{-1}(i)$

k-clustering

Definition

Let (M, d) be a metric space, equipped with the metric function d.
Given a set of points $X \subset M$, a *k-clustering* of X is a partition of X into k disjoint subsets, $C_1, \ldots, C_k$, called *clusters*.

Alternate Formulation

A *k-clustering* of X is an *assignment function*, $\alpha : X \to [k]$.
Each cluster C_i is the preimage of i under α i.e. $C_i = \alpha^{-1}(i)$

- There are many ways to quantify "how good a given clustering is"
- Depending on the objective, different variants of clustering problems are possible.
- Here, we consider two specific types of k-clustering.

k-center problem

Problem

Given a set of points $X \subset M$, find a k-clustering of X, denoted as $\mathcal{C}$, that minimizes

$$\phi(X, \mathcal{C}) = \max_{C \in \mathcal{C}} \left[\min_{c \in C} \max_{x \in C} d(x, c) \right]$$

k-median problem

Problem

Given a set of points $X \subset M$, find a k-clustering of X, denoted as $\mathcal{C}$, that minimizes

$$\psi(X, \mathcal{C}) = \sum_{C \in \mathcal{C}} \left[\min_{c \in C} \sum_{x \in C} d(x, c) \right]$$

Fair clustering?

- In order to consider a "fair" version of clustering, we first have to identify the *unprotected attribute* and *protected attribute*

Fair clustering?

- In order to consider a "fair" version of clustering, we first have to identify the *unprotected attribute* and *protected attribute*
- We shall consider the *coordinate* as the unprotected attribute.

Fair clustering?

- In order to consider a "fair" version of clustering, we first have to identify the *unprotected attribute* and *protected attribute*
- We shall consider the *coordinate* as the unprotected attribute.
- For simplicity, let us represent the protected attribute as the *coloring* of the points.

Fair clustering?

- In order to consider a "fair" version of clustering, we first have to identify the *unprotected attribute* and *protected attribute*
- We shall consider the *coordinate* as the unprotected attribute.
- For simplicity, let us represent the protected attribute as the *coloring* of the points.
- To simplify things further (as in the paper), let us only consider the case of binary coloring.

Fair clustering?

For $Y \subset X$, let us denote:

- $\chi : X \to \{\text{RED}, \text{BLUE}\}$ is the given binary coloring.
- $R(Y) = \{x \in X : \chi(x) = \text{RED}\}$, $r(Y) = |R(Y)|$
- $B(Y) = \{x \in X : \chi(x) = \text{BLUE}\}$, $b(Y) = |B(Y)|$

Fair clustering?

For $Y \subset X$, let us denote:

- $\chi : X \to \{\mathrm{RED}, \mathrm{BLUE}\}$ is the given binary coloring.
- $R(Y) = \{x \in X : \chi(x) = \mathrm{RED}\}$, $r(Y) = |R(Y)|$
- $B(Y) = \{x \in X : \chi(x) = \mathrm{BLUE}\}$, $b(Y) = |B(Y)|$

Definition

For $\emptyset \neq Y \subset X$, the *balance* of Y is defined as:

$$\mathrm{balance}(Y) = \min\left(\frac{r(Y)}{b(Y)}, \frac{b(Y)}{r(Y)}\right) \in [0, 1]$$

The *balance* of a clustering $\mathcal{C}$ is defined as:

$$\mathrm{balance}(\mathcal{C}) = \min_{C \in \mathcal{C}} \mathrm{balance}(C)$$

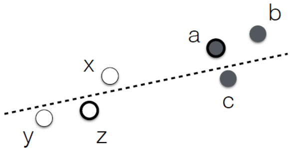
- If balance(Y) is 0(resp. 1), Y is fully unbalanced(resp. perfectly balanced)

Fair clustering?

- A clustering algorithm is *colorblind* if it doesn't take the protected attribute (coloring) into its decision making.

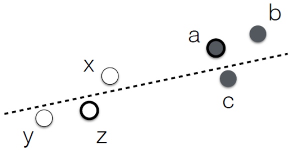
Fair clustering?

- A clustering algorithm is *colorblind* if it doesn't take the protected attribute (coloring) into its decision making.
- Colorblind algorithm may result in a very unfair clustering. (Unfair in the sense that the resulting clustering is very unbalanced)



Fair clustering?

- A clustering algorithm is *colorblind* if it doesn't take the protected attribute (coloring) into its decision making.
- Colorblind algorithm may result in a very unfair clustering. (Unfair in the sense that the resulting clustering is very unbalanced)



- Therefore a "fair" clustering must take into account not just the position of the centers, but also the assignment function!

Balance

Lemma(Combination)

Let $Y, Y' \subset X$ be disjoint.
If $\mathcal{C}$ and $\mathcal{C}'$ are clusterings of Y and Y', respectively, then

$$\text{balance}(\mathcal{C} \cup \mathcal{C}') = \min(\text{balance}(\mathcal{C}), \text{balance}(\mathcal{C}'))$$

- For any clustering $\mathcal{C}$ of X, we have $\text{balance}(\mathcal{C}) \leq \text{balance}(X)$.
- If X is not perfectly balanced, then no clustering of X can be perfectly balanced.

Toward fairlets

Definition

Let b, r be some integers such that $1 \leq b \leq r$ and $\gcd(b, r) = 1$.

- A clustering $\mathcal{Y}$ of X is called a $(b, r)-$*fairlet decomposition of* X if (i) $\forall Y \in \mathcal{Y}$ $|Y| \leq b + r$ and (ii) $\text{balance}(\mathcal{Y}) = b/r = \text{balance}(X)$
- Each $Y \in \mathcal{Y}$ is called a $(b, r)-$*fairlet*, or simply *fairlet*.

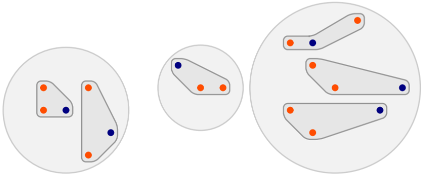
Toward fairlets

Definition

Let b, r be some integers such that $1 \le b \le r$ and $\gcd(b, r) = 1$.

- A clustering $\mathcal{Y}$ of X is called a $(b, r)-$*fairlet decomposition of* X if (i) $\forall Y \in \mathcal{Y}\ |Y| \le b + r$ and (ii) $\text{balance}(\mathcal{Y}) = b/r = \text{balance}(X)$
- Each $Y \in \mathcal{Y}$ is called a $(b, r)-$*fairlet*, or simply *fairlet*.

- Fairlet can be thought of as a group of points that are fair and cannot be split further into true subsets that are also fair.



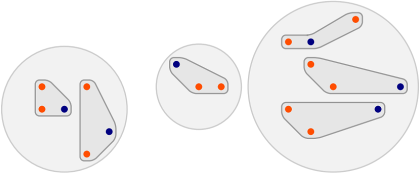
Toward fairlets

Definition

Let b, r be some integers such that $1 \leq b \leq r$ and $\gcd(b, r) = 1$.

- A clustering $\mathcal{Y}$ of X is called a $(b, r)-$*fairlet decomposition of* X if (i) $\forall Y \in \mathcal{Y}\ |Y| \leq b + r$ and (ii) $\text{balance}(\mathcal{Y}) = b/r = \text{balance}(X)$
- Each $Y \in \mathcal{Y}$ is called a $(b, r)-$*fairlet*, or simply *fairlet*.

- Fairlet can be thought of as a group of points that are fair and cannot be split further into true subsets that are also fair.



- Intuitively, the balance of the original set of points is preserved while keeping each cluster "small".

Toward fairlets

Lemma

Let $\text{balance}(X) = b/r$ for some integers $1 \leq b \leq r$ such that $\gcd(b, r) = 1$. Then there exists a $(b, r)-$fairlet decomposition of X.

Toward fairlets

Lemma

Let $\text{balance}(X) = b/r$ for some integers $1 \leq b \leq r$ such that $\gcd(b, r) = 1$. Then there exists a $(b, r)-$fairlet decomposition of X.

- This lemma tells us that every fair solution to the clustering problem induces a set of minimal fairlets
- (Proof is very simple! The proof in the paper seems too complex...)

$(t,k)-$fair clustering problems

$(t,k)-$fair center (resp. median) problem

Partition X into $\mathcal{C}$ such that

- $|\mathcal{C}| = k$
- balance$(\mathcal{C}) \geq t$
- $\phi(X,\mathcal{C})$ (resp. $\psi(X,\mathcal{C})$) is minimized.

$(t, k)-$fair clustering problems

$(t, k)-$fair center (resp. median) problem

Partition X into $\mathcal{C}$ such that

- $|\mathcal{C}| = k$
- balance$(\mathcal{C}) \geq t$
- $\phi(X, \mathcal{C})$ (resp. $\psi(X, \mathcal{C})$) is minimized.

- If fairness is not taken into account, the assignment function is implicit through a set $\{c_1, \dots, c_k\}$ of centers i.e.

$$\alpha(x) = \text{argmin}_{i \in [k]} d(x, c_i)$$

$(t, k)-$fair clustering problems

$(t, k)-$fair center (resp. median) problem

Partition X into $\mathcal{C}$ such that

- $|\mathcal{C}| = k$
- balance$(\mathcal{C}) \geq t$
- $\phi(X, \mathcal{C})$ (resp. $\psi(X, \mathcal{C})$) is minimized.

- If fairness is not taken into account, the assignment function is implicit through a set $\{c_1, \dots, c_k\}$ of centers i.e.

$$\alpha(x) = \text{argmin}_{i \in [k]} d(x, c_i)$$

- With fairness, an explicit assignment function is required.

Outline

Fairlet decomposition cost

- $\mathcal{Y} = \{Y_1, \ldots, Y_m\}$: a fairlet decomposition of X
- $y_j \in Y_j$ is the *center* of Y_j. (Its choice is arbitrary)
- $\beta : X \to [m]$ is the mapping from a point to the index of the fairlet to which it is mapped.

Definition

For a fairlet decomposition, define its costs:

- k-median cost $= \sum_{x \in X} d\left(x, y_{\beta(x)}\right)$
- k-center cost $= \max_{x \in X} d\left(x, y_{\beta(x)}\right)$

Also, we say that a (b, r)-fairlet decomposition is *optimal* if it has minimum cost among all possible (b, r)-fairlet decompositions.

Reduction to colorblind clustering

- Recall that a (t, k)-fair clustering of X requires that $t \leq \text{balance}(X)$

Reduction to colorblind clustering

- Recall that a (t, k)-fair clustering of X requires that $t \leq \text{balance}(X)$
- To achieve this, we consider the vanilla k-clustering of the centers of each fairlet i.e. k-clustering of $\{y_1, \dots, y_m\}$

Reduction to colorblind clustering

- Recall that a (t, k)-fair clustering of X requires that $t \leq \text{balance}(X)$
- To achieve this, we consider the vanilla k-clustering of the centers of each fairlet i.e. k-clustering of $\{y_1, \ldots, y_m\}$
- Then we obtain a set of centers $\{c_1, \ldots, c_k\}$ and an assignment function $\alpha_Y : Y \to [k]$.

Reduction to colorblind clustering

- Recall that a (t, k)-fair clustering of X requires that $t \leq \text{balance}(X)$
- To achieve this, we consider the vanilla k-clustering of the centers of each fairlet i.e. k-clustering of $\{y_1, \ldots, y_m\}$
- Then we obtain a set of centers $\{c_1, \ldots, c_k\}$ and an assignment function $\alpha_Y : Y \to [k]$.
- Define $\alpha(x) = \alpha_Y(y_{\beta(x)})$ as the overall assignment function and denote $\mathcal{C}_\alpha$ as the clustering induced by α.

Reduction to colorblind clustering

- Recall that a (t, k)-fair clustering of X requires that $t \leq \text{balance}(X)$
- To achieve this, we consider the vanilla k-clustering of the centers of each fairlet i.e. k-clustering of $\{y_1, \ldots, y_m\}$
- Then we obtain a set of centers $\{c_1, \ldots, c_k\}$ and an assignment function $\alpha_Y : Y \to [k]$.
- Define $\alpha(x) = \alpha_Y(y_{\beta(x)})$ as the overall assignment function and denote $\mathcal{C}_\alpha$ as the clustering induced by α.
- Then we have that balance $C_\alpha = t$

Reduction to colorblind clustering

- Recall that a (t, k)-fair clustering of X requires that $t \leq \text{balance}(X)$
- To achieve this, we consider the vanilla k-clustering of the centers of each fairlet i.e. k-clustering of $\{y_1, \ldots, y_m\}$
- Then we obtain a set of centers $\{c_1, \ldots, c_k\}$ and an assignment function $\alpha_Y : Y \to [k]$.
- Define $\alpha(x) = \alpha_Y(y_{\beta(x)})$ as the overall assignment function and denote $\mathcal{C}_\alpha$ as the clustering induced by α.
- Then we have that balance $C_\alpha = t$
- Also, its cost is bounded, as shown in the next lemma.

Lemma 6 (corrected)

Denote $\tilde{Y}$ as a multiset where each y_i appears $|Y_i|$ number of times. Then,

$$\psi(X, \mathcal{C}_\alpha) \leq \psi(X, \mathcal{Y}) + \psi(\tilde{Y}, \mathcal{C}_\alpha)$$

$$\phi(X, \mathcal{C}_\alpha) \leq \phi(X, \mathcal{Y}) + \phi(\tilde{Y}, \mathcal{C}_\alpha)$$

Reduction to colorblind clustering

Lemma 6 (corrected)

Denote $\tilde{Y}$ as a multiset where each y_i appears $|Y_i|$ number of times. Then,

$$\psi(X, \mathcal{C}_\alpha) \leq \psi(X, \mathcal{Y}) + \psi(\tilde{Y}, \mathcal{C}_\alpha)$$

$$\phi(X, \mathcal{C}_\alpha) \leq \phi(X, \mathcal{Y}) + \phi(\tilde{Y}, \mathcal{C}_\alpha)$$

This lemma, along with previous reasoning, shows that the fair clustering problem can be reduced to

- Find a good fairlet decomposition (α-approximation)
- Solve the vanilla clustering problem on the centers of the fairlets (β-approximation)

, which is actually a $(\alpha + \beta)$-approximation in total!

Proof of Lemma 6

- Let us only consider the k-median setting; k-center version is similar.
- Let $\mathcal{C}_\alpha = \{C_1, \ldots, C_k\}$ with corresponding centers $\{c_1, \ldots, c_k\}$. Then, $i = \operatorname{argmin}_{j\in[k]} d(x, c_j)$.
- Using the definition and triangle inequality,

$$\psi(X, \mathcal{C}_\alpha) = \sum_{C\in\mathcal{C}_\alpha}\left[\sum_{x\in C}\min_{c\in C} d(x,c)\right]$$

$$= \sum_{i=1}^{k}\sum_{x\in C_i} d(x, c_i)$$

$$\leq \sum_{i=1}^{k}\sum_{x\in C_i} \left(d(x, y_{\beta(x)}) + d(y_{\beta(x)}, c_i)\right)$$

$$= \psi(X, \mathcal{Y}) + \psi(\tilde{Y}, \mathcal{C}_\alpha)$$

Outline

Outline

Fair k-center: $(1, 1)$-fairlets

- Let us first consider the case when $\text{balance}(X) = 1$.

Fair k-center: $(1, 1)$-fairlets

- Let us first consider the case when $\text{balance}(X) = 1$.
- How can we find a perfectly balanced clustering?

Fair k-center: $(1, 1)$-fairlets

- Let us first consider the case when balance$(X) = 1$.
- How can we find a perfectly balanced clustering?
- We utilize a good $(1, 1)$-fairlet decomposition!

Fair k-center: $(1, 1)$-fairlets

- Let us first consider the case when $\text{balance}(X) = 1$.
- How can we find a perfectly balanced clustering?
- We utilize a good $(1, 1)$-fairlet decomposition!

Lemma 7

An optimal $(1, 1)$-fairlet decomposition for k-center can be found in polynomial time.

(The approach used in the proof will be used later!)

Proof of Lemma 7

- We shall prove this by relating it to a graph covering problem.
- Denote $B(X) = \{b_i\}_i$ and $R(X) = \{r_j\}_j$
- Create a weighted, *complete* bipartite graph $G = (B, R, E)$ with the weight function $w(b_i, r_j) = d(b_i, r_j)$
- Every $(1, 1)$-fairlet decomposition corresponds to some perfect matching in G where each edge represents a fairlet, Y_i.
- Letting $\mathcal{Y} = \{Y_i\}_i$, the k-center cost $\phi(X, \mathcal{Y})$ is exactly the cost of the maximum weight edge in the matching.

Proof of Lemma 7

- Now, our problem is to find a perfect matching that minimizes the weight of the maximum edge.
- Can be done in $O(n^2)$ time.
  (cf. "threshold graph")
- For each Y_i, arbitrarily set one of the two nodes of the corresponding edge as the center, y_i.

Fair k-center: $(1, 1)$-fairlets

- From Lemma 3, we know that any fair solution induces a set of minimal fairlets.

Fair k-center: $(1, 1)$-fairlets

- From Lemma 3, we know that any fair solution induces a set of minimal fairlets.
- Thus, the cost of the fairlet decomposition found is at most twice the cost of an optimal solution to the clustering.

Lemma 8 (corrected)

Let $\mathcal{Y}$ be the partition found previously, and let ϕ_t^* be the cost of the optimal (t, k)-fair center clustering. Then, $\phi(X, \mathcal{Y}) \leq 2\phi_t^*$.

Fair k-center: $(1, 1)$-fairlets

- From Lemma 3, we know that any fair solution induces a set of minimal fairlets.
- Thus, the cost of the fairlet decomposition found is at most twice the cost of an optimal solution to the clustering.

Lemma 8 (corrected)

Let $\mathcal{Y}$ be the partition found previously, and let ϕ_t^* be the cost of the optimal (t, k)-fair center clustering. Then, $\phi(X, \mathcal{Y}) \leq 2\phi_t^*$.

- Let us utilize a result by Gonzalez for k-center problem:

Theorem (Gonzalez, 1985)

There is an algorithm which, given a k-center instance $\mathcal{I}$, produces a 2-approximation solution to $\mathcal{I}$ in running time $O(kn)$

Fair k-center: $(1, 1)$-fairlets

Theorem 9 (corrected)

The algorithm that first finds fairlets and then clusters them is a 4-approximation for the $(1, k)$-fair center problem.

Outline

Fair k-center: $(1, t')$-fairlets

- Now let us consider the case when balance$(X) = t < 1$.

Fair k-center: $(1, t')$-fairlets

- Now let us consider the case when $\mathrm{balance}(X) = t < 1$.
- For simplicity, assume that $t = 1/t'$ for some integer $t' > 1$ (as done in the paper)

Fair k-center: $(1, t')$-fairlets

- Now let us consider the case when $\text{balance}(X) = t < 1$.
- For simplicity, assume that $t = 1/t'$ for some integer $t' > 1$ (as done in the paper)
- As a generalization of previous argument, we shall transform this problem into a minimum cost flow problem (MCFP).

MCFP

Definition

A *flow network* is a directed graph $G = (V, E)$ with a source vertex $s \in V$ and a sink vertex $t \in V$, where each edge $(u, v) \in E$ has capacity $c(u, v) > 0$, flow $f(u, v) \geq 0$ and cost $a(u, v) \in \mathbb{R}$

Minimum Cost Flow Problem (MCFP)

Input: A flow network $(G = (V, E), s, t, c, a)$ (without the flow), d
Constraints:

- Capacity constraints: $f(u, v) \leq c(u, v)$
- Skew symmetry: $f(u, v) = -f(v, u)$
- Flow conservation: $\forall u \neq s, t \ \sum_{w \in V} f(u, w) = 0$
- Required flow from s to t: $\sum_{w \in V} f(s, w) = \sum_{w \in V} f(w, t) = d$

Output: Flow $f(u, v)$ such that $\sum_{(u,v) \in E} a(u, v) f(u, v)$ is minimized

Fair k-center: $(1, t')$-fairlets

- Let us construct an instance of MCFP, with a parameter $\tau > 0$.

Fair k-center: $(1, t')$-fairlets

- Let us construct an instance of MCFP, with a parameter $\tau > 0$.
- First, let us construct a directed graph $H_\tau = (V, E)$

Fair k-center: $(1, t')$-fairlets

- Let us construct an instance of MCFP, with a parameter $\tau > 0$.
- First, let us construct a directed graph $H_\tau = (V, E)$
- Vertex set:

$$V = \{\beta, \rho\} \cup B(X) \cup R(X) \cup \left\{ b_i^j | b_i \in B(X) \right\}_{j \in [t']} \cup \left\{ r_i^j | r_i \in R(X) \right\}_{j \in [t']}$$

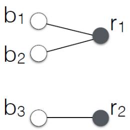
- Edge set:
  - (β, ρ) with cost 0 and capacity $\min(|B(X)|, |R(X)|)$
  - (β, b_i) and (r_i, ρ) for each $b_i \in B(X), r_i \in R(X)$, each with cost 0 and capacity $t' - 1$
  - (b_i, b_i^j) and (r_i, r_i^j) for each $b_i \in B(X), r_i \in R(X), j \in [t']$, each with cost 0 and capacity 1
  - (b_i^k, r_j^l) for each $b_i \in B(X), r_i \in R(X), 1 \leq k, l \leq t'$, each with cost 1 if $d(b_i, r_j) \leq \tau$ and ∞ otherwise.

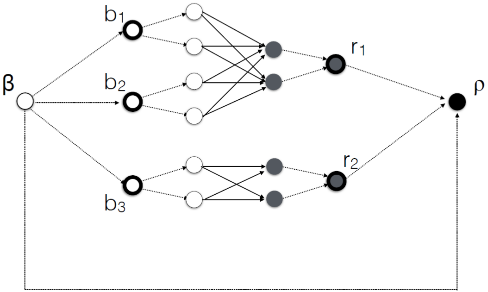
Fair k-center: $(1, t')$-fairlets

- To finish the description, we need specify the supply and demand at every node:
  - Every node in $B(X)$ has a supply of 1
  - Every node in $R(X)$ has a demand of 1
  - β has a supply of $|R(X)|$
  - ρ has a demand of $|B(X)|$
  - Every other node has zero supply and demand

Fair k-center: $(1, t')$-fairlets

- To finish the description, we need specify the supply and demand at every node:
  - Every node in $B(X)$ has a supply of 1
  - Every node in $R(X)$ has a demand of 1
  - β has a supply of $|R(X)|$
  - ρ has a demand of $|B(X)|$
  - Every other node has zero supply and demand





Fair k-center: $(1, t')$-fairlets

Lemma 10

Let $\mathcal{Y}$ be a $(1, t')$-fairlet decomposition of cost C for the $(1/t', k)$-fair center problem. Then it is possible to construct a feasible solution of cost $2C$ to the (constructed) MCF instance.

Above lemma tells us that a $(1, t')$-fairlet decomposition can be used to construct a feasible solution for the MCF instance of twice the cost.

Proof of Lemma 10

(Excluded due to time limit. If requested, I will make the slides and update the file!)

Fair k-center: $(1, t')$-fairlets

Lemma 11

Let $\mathcal{Y}$ be an optimal solution of cost C to the (constructed) MCF instance. Then it is possible to construct a $(1, t')$-fairlet decomposition for the $(1/t', k)$-fair center problem of cost at most C.

Above lemma tells us that an optimal solution for the MCF instance can be used to obtain a $(1, t')$-fairlet decomposition of bounded cost.

Proof of Lemma 11

(Excluded due to time limit. If requested, I will make the slides and update the file!)

Fair k-center: $(1, t')$-fairlets

Combining the previous two lemmas yield:

Lemma 12

By reducing the $(1, t')$-fairlet decomposition problem to an MCFP, it is possible to compute a 2-approximation for the optimal $(1, t')$-fairlet decomposition for the $(1/t', k)$-fair center problem.

Combining above with the result by Gonzalez gives... (next slide)

Fair k-center

Theorem 13

For any integer $t \in \mathbb{N}$, the algorithm that first finds fairlets and then clusters them is a 4-approximation for the $(1/t', k)$-fair center problem.

Outline

Fair k-median

- Results from previous section can be modified for the (t, k)-fair median problem

Fair k-median

- Results from previous section can be modified for the (t, k)-fair median problem
- For the perfectly balanced case, our goal is to look for a perfect matching of minimum total cost on the bichromatic graph.

Fair k-median

- Results from previous section can be modified for the (t, k)-fair median problem
- For the perfectly balanced case, our goal is to look for a perfect matching of minimum total cost on the bichromatic graph.
- To find $(1, t')$-fairlet decomposition for integer $t' > 1$, we again resort to MCF and create an instance as done before.

Fair k-median

- Results from previous section can be modified for the (t, k)-fair median problem
- For the perfectly balanced case, our goal is to look for a perfect matching of minimum total cost on the bichromatic graph.
- To find $(1, t')$-fairlet decomposition for integer $t' > 1$, we again resort to MCF and create an instance as done before.
- In this case, for each $b_i \in B, r_j \in R$ and $1 \leq k, l \leq t$, set the cost of the edge (b_i^k, r_j^k) to $d(b_i, r_j)$

Fair k-median

- Results from previous section can be modified for the (t, k)-fair median problem
- For the perfectly balanced case, our goal is to look for a perfect matching of minimum total cost on the bichromatic graph.
- To find $(1, t')$-fairlet decomposition for integer $t' > 1$, we again resort to MCF and create an instance as done before.
- In this case, for each $b_i \in B, r_j \in R$ and $1 \leq k, l \leq t$, set the cost of the edge (b_i^k, r_j^k) to $d(b_i, r_j)$
- Let us utilize a result by Li & Svensson for k-median problem:

Theorem (Li & Svensson, 2013)

There is an algorithm which, given a k-median instance $\mathcal{I}$ and $\varepsilon > 0$, produces a $(1 + \sqrt{3} + \varepsilon)$-approximation solution to $\mathcal{I}$ in running time $O\left(n^{O(1/\varepsilon^2)}\right)$

Fair k-median

Theorem 15

For any integer $t' \in \mathbb{N}$, the algorithm that first finds fairlets and then clusters them is a $(t' + 1 + \sqrt{3} + \varepsilon)$-approximation for the $(1/t', k)$-fair median problem.

Outline

Hardness

- We now have a theoretical framework and an actual algorithm for solving fair clustering problems.

Hardness

- We now have a theoretical framework and an actual algorithm for solving fair clustering problems.
- But by taking fairness into account, we have introduced some extra complexity to the classical clustering problems.

Hardness

- We now have a theoretical framework and an actual algorithm for solving fair clustering problems.
- But by taking fairness into account, we have introduced some extra complexity to the classical clustering problems.
- How bad can it be, right?

Hardness

- We now have a theoretical framework and an actual algorithm for solving fair clustering problems.
- But by taking fairness into account, we have introduced some extra complexity to the classical clustering problems.
- How bad can it be, right?
- Well, as the next theorem shows, ensuring fairness actually introduces a computational bottleneck! (and a very narrow one, indeed.)

Hardness

- We now have a theoretical framework and an actual algorithm for solving fair clustering problems.
- But by taking fairness into account, we have introduced some extra complexity to the classical clustering problems.
- How bad can it be, right?
- Well, as the next theorem shows, ensuring fairness actually introduces a computational bottleneck! (and a very narrow one, indeed.)

Theorem 16

For each fixed $t' \geq 3$,

- Finding an optimal $(1, t')$-fairlet decomposition is NP-hard.
- Finding the minimum cost $(1/t', k)$-fair median clustering is NP-hard

Proof of Theorem 16

Definition

For an arbitrary graph H and given graph G,

- *H-packing* of G is a set $\{H_1, \ldots, H_d\}$ of disjoint subgraphs of G such that $\forall i \; H_i \cong H$.
- *H-factor* of G is a H-packing such that the set $\{V(H_1), \ldots, V(H_d)\}$ is a partition of $V(G)$.
- A *H-factor* of G is *strict* if each H_i belonging to the packing is an induced subgraph of G.

Proof of Theorem 16

Definition

For an arbitrary graph H and given graph G,

- *H-packing* of G is a set $\{H_1, \ldots, H_d\}$ of disjoint subgraphs of G such that $\forall i \; H_i \cong H$.
- *H-factor* of G is a H-packing such that the set $\{V(H_1), \ldots, V(H_d)\}$ is a partition of $V(G)$.
- A *H-factor* of G is *strict* if each H_i belonging to the packing is an induced subgraph of G.

S-FACT(H)

Input: An undirected, connected graph $G = (V, E)$
Question: Does G admit a strict H-factor?

Proof of Theorem 16

Definition

For an arbitrary graph H and given graph G,

- *H-packing* of G is a set $\{H_1, \ldots, H_d\}$ of disjoint subgraphs of G such that $\forall i \; H_i \cong H$.
- *H-factor* of G is a H-packing such that the set $\{V(H_1), \ldots, V(H_d)\}$ is a partition of $V(G)$.
- A *H-factor* of G is *strict* if each H_i belonging to the packing is an induced subgraph of G.

$\mathrm{S\text{-}FACT}(H)$

Input: An undirected, connected graph $G = (V, E)$
Question: Does G admit a strict H-factor?

Theorem (Kirkpatrick & Hell, 1978)

If H has at least 3 vertices, then $\mathrm{S\text{-}FACT}(H)$ is NP-complete.

Proof of Theorem 16

- We shall prove Theorem 16 by reduction from $\text{S-FACT}(K_{1,t'-1})$.
- To do this, WLOG assume that $|V|$ is divisible by t' and consider the following instance of set of red-blue points.

Proof of Theorem 16

- We shall prove Theorem 16 by reduction from $\mathrm{S\text{-}FACT}(K_{1,t'-1})$.
- To do this, WLOG assume that $|V|$ is divisible by t' and consider the following instance of set of red-blue points.
- X consists of $|V|$ red points (which are denoted as r_v's to represent the correspondence between red points and V), and $|V|/t'$ blue points.
- X is equipped with the metric function d, defined as

$$d(x,y) = \begin{cases} 1 & \text{if } x = r_{v_1}, y = r_{v_2}, v_1 v_2 \in E \\ 2 & \text{otherwise} \end{cases}$$

(Easy to see!)

Proof of Theorem 16

Note that:

- Fairlet decomposition problem:
  Does this instance admit a $(1, t')$-fairlet decomposition with total cost upper bounded by $\left(1 + \frac{1}{t'}\right)|V|$?
- Fair median problem:
  Does this instance admit a $(1/t', k')$-fair k-clustering, with $k = |V|/t'$, having median cost upper bounded by $\left(1 + \frac{1}{t'}\right)|V|$?

Proof of Theorem 16

- Suppose that G admits a strict $K_{1,t'-1}$-factor, whose set of vertex sets is denoted as $\{S_1, \ldots, S_{|V|/t'}\}$.
- For each $i \in [|V|/t']$, create a cluster C_i consisting of red elements corresponding to S_i and one ith blue element.
- Observe that the cost of each C_i is at most $t' + 1$ since S_i induces a $(t' - 1)$-star, another name for $K_{1,t'-1}$.
- Then the total cost is at most $\frac{|V|}{t'}(t' + 1) = \left(1 + \frac{1}{t'}\right)|V|$.

Proof of Theorem 16

- Now suppose that G does not admit a strict $K_{1,t'-1}$-factor.
- Note that any feasible solution has to create $k = |V|/t'$ clusters, each containing exactly 1 blue element and t' red elements. (fairlets!)
- Observe that the median cost of C_i is $t' + 1$ if S_i induces a $(t' - 1)$-star, and at least $t' + 2$ otherwise
- Then the total cost (of either problems) is at least $(t' + 1)\frac{|V|-t'}{t'} + (t' + 2) = \left(1 + \frac{1}{t'}\right)|V| + 1$.

Outline

Experiments

The goal of the experiments is two-fold:

Experiments

The goal of the experiments is two-fold:

- Traditional algorithms for k-center and k-median tend to produce unfair clusters

Experiments

The goal of the experiments is two-fold:

- Traditional algorithms for k-center and k-median tend to produce unfair clusters
- Proposed algorithm outputs clusters that respect the fairness guarantees

Experiment Design

- 3 datasets from the UCI repository Lichman (2013) were used:
  Diabetes, Bank, Sensus
  (Protected attributes: gender, married or not, gender, respectively)

Experiment Design

- 3 datasets from the UCI repository Lichman (2013) were used: *Diabetes, Bank, Sensus*
  (Protected attributes: gender, married or not, gender, respectively)
- Flow-based fairlet decomposition algorithm (as proposed) was implemented.

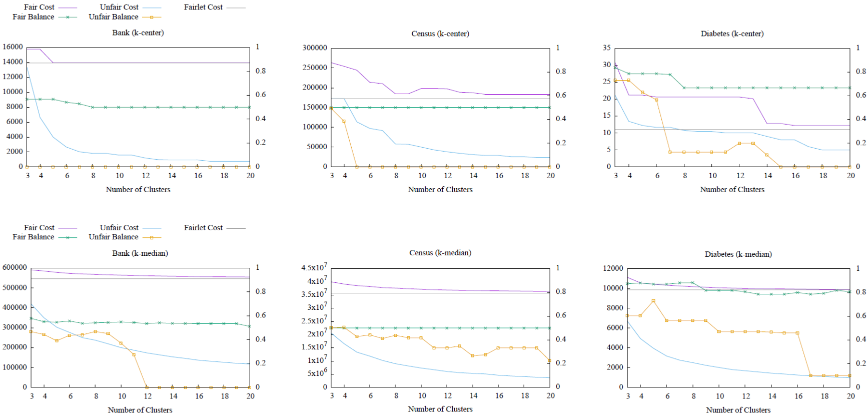
Experiment Design

- 3 datasets from the UCI repository Lichman (2013) were used: *Diabetes, Bank, Sensus*
  (Protected attributes: gender, married or not, gender, respectively)
- Flow-based fairlet decomposition algorithm (as proposed) was implemented.
- For the vanilla k-center clustering algorithm, the greedy furthest point algorithm due to Gonzalez (1985) was used.
  (known to obtain 2-approximation)

Experiment Design

- 3 datasets from the UCI repository Lichman (2013) were used: *Diabetes, Bank, Sensus*
  (Protected attributes: gender, married or not, gender, respectively)
- Flow-based fairlet decomposition algorithm (as proposed) was implemented.
- For the vanilla k-center clustering algorithm, the greedy furthest point algorithm due to Gonzalez (1985) was used.
  (known to obtain 2-approximation)
- For the vanilla k-median clustering algorithm, single swap algorithm due to Arya *et al.* (2004) was used.
  (known to obtain 5-approximation, but performs well in practice. Refer to Kanungo *et al.*, 2002)

Results



Fair Cost
Unfair Cost
Fairlet Cost
Fair Balance
Unfair Balance
Bank (k-center)
Census (k-center)
Diabetes (k-center)
Bank (k-median)
Census (k-median)
Diabetes (k-median)
Number of Clusters

Outline

Summary

In summary,

Summary

In summary,

- Reduction of fair clustering to classical clustering via fairlets

Summary

In summary,

- Reduction of fair clustering to classical clustering via fairlets
- Efficient approximation algorithms for finding fairlet decompositions

Summary

In summary,

- Reduction of fair clustering to classical clustering via fairlets
- Efficient approximation algorithms for finding fairlet decompositions
- Showed that fairness can introduce a computational bottleneck

Outline

Future research

- Improve the approximation ratio of the decomposition algorithms

Future research

- Improve the approximation ratio of the decomposition algorithms
- Give stronger hardness results

Future research

- Improve the approximation ratio of the decomposition algorithms
- Give stronger hardness results
- Extend to the case where the protected class is not binary, but can take on multiple values
  (Already done! *Scalable Fair Clustering (ICML 2019)*)

Outline

Roles and Responsibilities

- Junghyun Lee prepared all the materials.

Roles and Responsibilities

- Junghyun Lee prepared all the materials.
- And as requested, here is a picture of myself:

Thank you for your attention! Any questions?