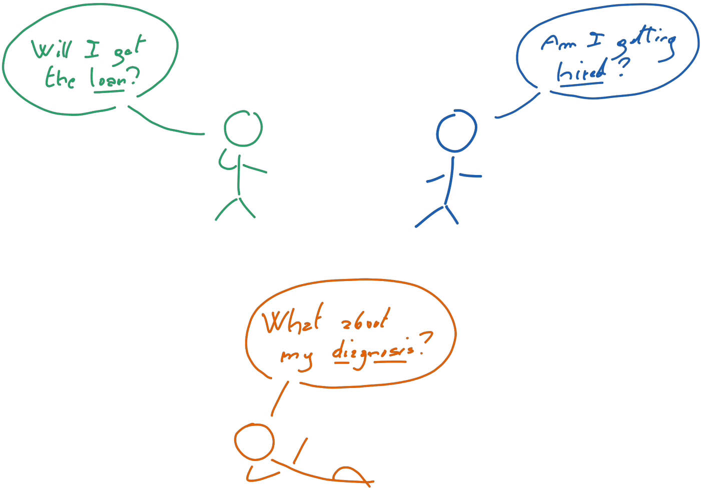
Fast and Accurate Fair k-Center Clustering in Doubling Metrics

Matteo Ceccarello

U. of Padova

Joint work with Andrea Pietracaprina and Geppino Pucci


Will I get the loan?
Am I getting hired?
What about my diagnosis?

Problem definition

Disparate impact

People in different protected classes should not experience disproportionally different outcomes.

Problem definition

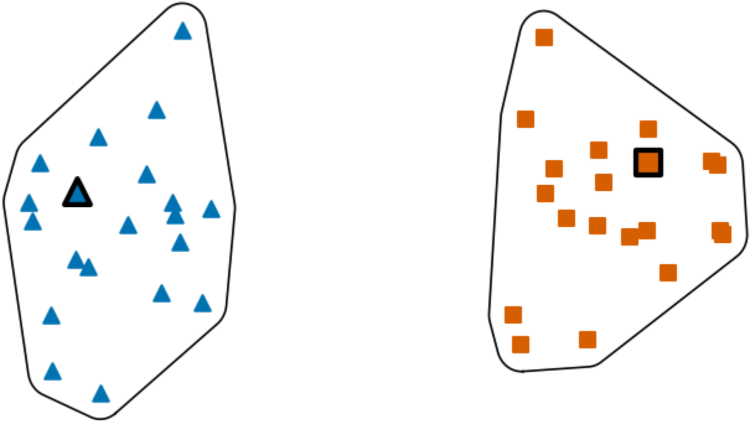
Disparate impact

People in different protected classes should not experience disproportionally different outcomes.

Unawareness does not help

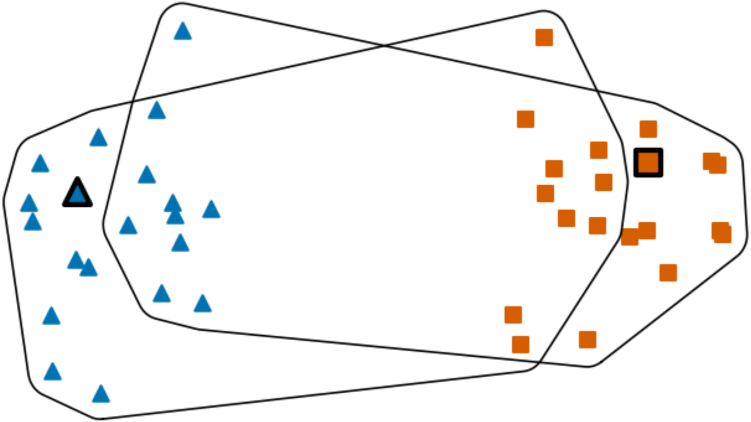
Blindly ignoring protected attributes is no solution: correlated features (e.g. height which correlates with biological sex) can leak information about the protected attributes and thus lead to *unfair* solutions.

Problem definition



Classic k-center assigns each point to the closest center.

Problem definition



If we want to balance the colors in each cluster, we possibly have to assign points to farther away centers.

Problem definition

- Metric space $(\mathcal{X}, d)$
- Set of points S
- Each point has one (or more) colors out of a set Γ
- Parameter k

Goal

Build a clustering such that the *proportion* of points from each protected group is the same as the proportion in the entire dataset.

Problem definition

distance function

center of x

find set of centers $\subseteq S$

$$r_{\boxed{C},\boxed{\phi}}(S) = \max_{x \in \boxed{S}} \boxed{d}\,(x, \boxed{\phi(x)}\,)$$

find assignment $\phi : S \to C$

input set

Problem definition

distance function

center of x

find set of centers $\subseteq S$

$$r_{C,\phi}(S) = \max_{x \in S} d(x, \phi(x))$$

find assignment $\phi : S \to C$

input set

Minimize the above, where the assignment is subject to

points of color ℓ assigned to c

cluster centers

$$\beta_\ell \le \frac{|S_{c,\ell}|}{|S_c|} \le \alpha_\ell \qquad \forall\, c \in C,\ \ell \in \Gamma$$

all points assigned to c

set of colors

fairness constraints

State of the art

- Find a set of k centers, ignoring fairness
- Build the assignment function by means of linear programming, imposing fairness

State of the art

- Find a set of k centers, ignoring fairness
- Build the assignment function by means of linear programming, imposing fairness

- 3 approximation [Ber+19; HL20]

State of the art

- Find a set of k centers, ignoring fairness
- Build the assignment function by means of linear programming, imposing fairness

- 3 approximation [Ber+19; HL20]
- 9 approximation in MapReduce, $7 + \varepsilon$ in Streaming [Ber+22]

State of the art

- Find a set of k centers, ignoring fairness
- Build the assignment function by means of linear programming, imposing fairness

- 3 approximation [Ber+19; HL20]
- 9 approximation in MapReduce,
  $7 + \varepsilon$ in Streaming [Ber+22]
- large linear program of size $O(k \cdot n)$

Our contribution

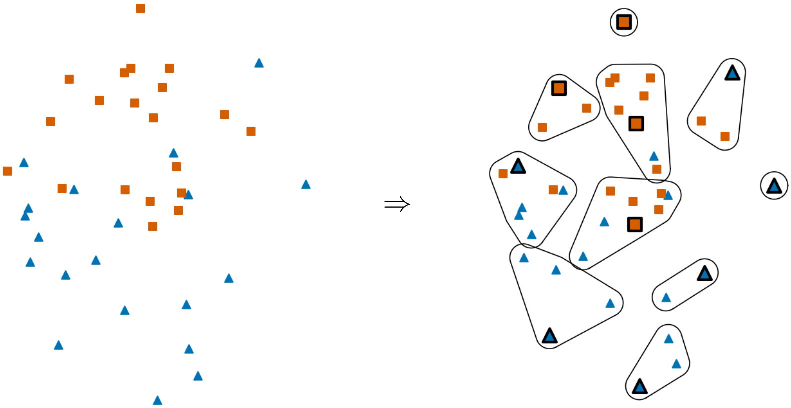
$3 + \varepsilon$ approximation algorithms

- **Sequential**: Linear time in the input size
- **Streaming**: 2 passes and memory $O\left(\log \frac{d_{max}}{d_{min}}\right)$
- **MapReduce**: 5 rounds and memory $O\left(\max\{|S|/p, p\}\right)$, where p is the number of processors

Coreset

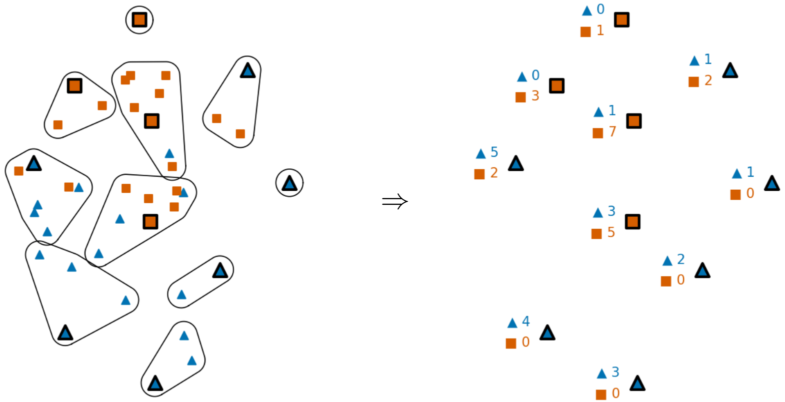
- Set $T \subseteq S$
- $|T| \ll |S|$
- Proxy function $\pi : S \to T$
- Weight function $w : T \to \mathbb{N}$

Locate a set of *proxy points*



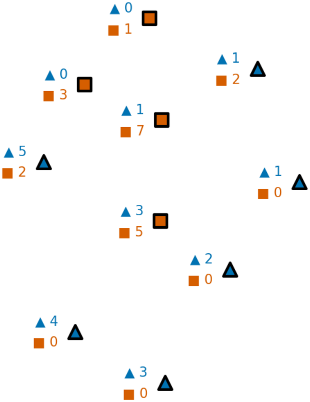
Goal: find a good compact representation of the input

Assign weights to coreset points

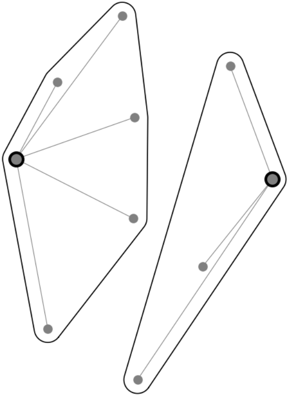


Goal: enable addressing fairness later

Find an unfair k-center clustering on the coreset

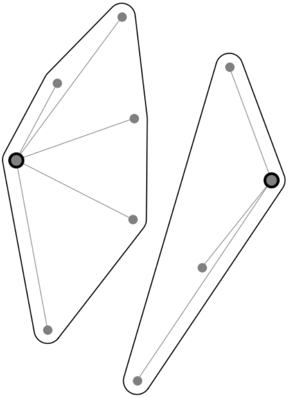


$\Rightarrow$

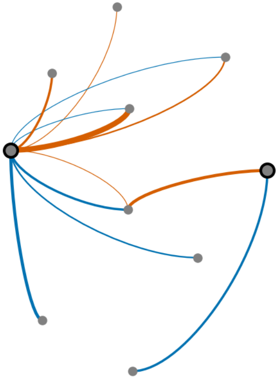


Goal: optimize the placement of centers

Distribute weight to centers

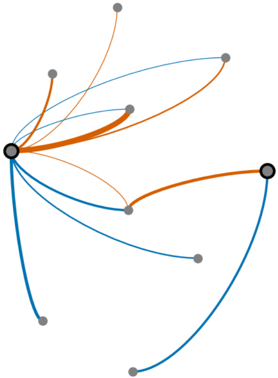


$\Rightarrow$

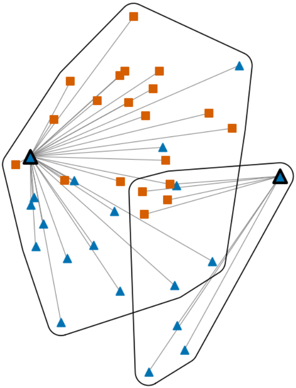


Goal: address fairness

Assign original points

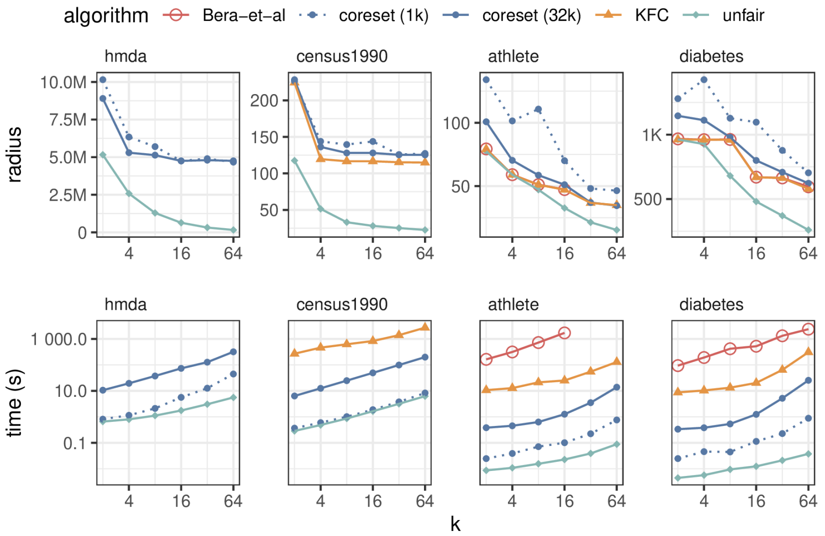


$\Rightarrow$

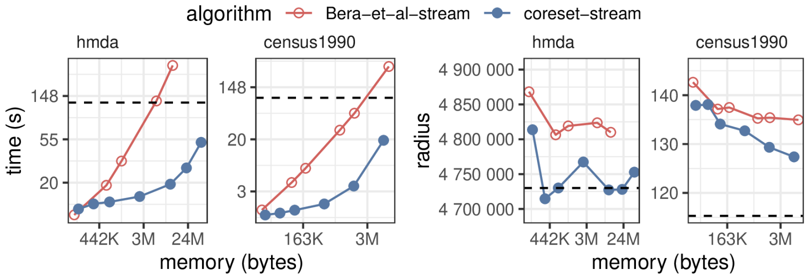


Goal: compute the solution

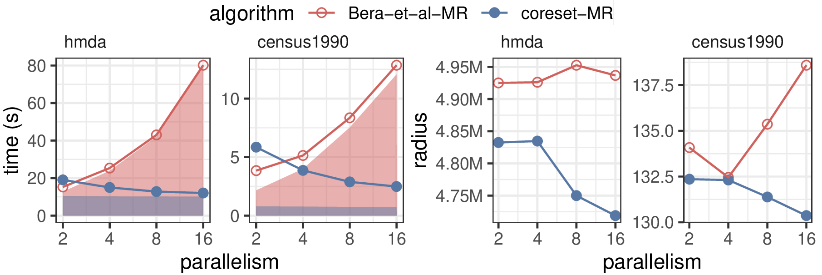
Experiments (sequential)



Experiments (Streaming)



Experiments (MapReduce)



Thank you!

Appendix

Problem definition

Additive violation

The additive violation of an assignment, w.r.t. the fairness constraints α_ℓ, β_ℓ, is the minimum $\mathcal{E}$ s.t. $\forall c \in C, \ell \in \Gamma$

constraints on the number of points with color ℓ in C

$$\beta_\ell |S_c| - \mathcal{E} \leq |S_{c,\ell}| \leq \alpha_\ell |S_c| + \mathcal{E}$$

violation

Our algorithm provides an additive violation of $4\Delta + 3$, where Δ is the maximum number of colors per point.

Preliminaries: GMM

Classic algorithm for *unfair* k-center.

Input: Set S, parameter k

$C \leftarrow \{\text{arbitrary point from } S\}$;
while $|C| < k$ **do**
 $c \leftarrow \arg\max_{x \in S} d(x, C)$;
 $C \leftarrow C \cup \{c\}$;
return C;

Provides a 2-approximation in time $O(k \cdot n)$

Starting point [Ber+19; HL20]

Algorithm

1. Find a set C of centers with the GMM algorithm
2. Do a binary search on guesses over the possible clustering radii
3. Instantiate a linear program and find a fractional solution (if any)
4. If the linear program has a feasible solution, then iteratively round it (will not see today).

Starting point [Ber+19; HL20]

Algorithm

1. Find a set C of centers with the GMM algorithm
2. Do a binary search on guesses over the possible clustering radii
3. Instantiate a linear program and find a fractional solution (if any)
4. If the linear program has a feasible solution, then iteratively round it (will not see today).

Guarantees

- The algorithm provides a 3 approximation to the radius
- The fairness constraints have an additive violation up to 7

Outline of our approach

1. Build a *weighted* coreset T out of the input set S
2. Compute a fair k-center clustering on of T, whose centers are C
3. Build a fair assignment of S to C by using information from the solution on the coreset

Outline of our approach

1. Build a *weighted* coreset T out of the input set S
2. Compute a fair k-center clustering on of T, whose centers are C
3. Build a fair assignment of S to C by using information from the solution on the coreset

---

In MapReduce, steps 1. and 3. are carried out in a single parallel round each

In Streaming, steps 1. and 3. require each a pass on the data

Sequential coreset construction

Input: Set S, parameter k, parameter ε

$T \leftarrow \{\text{arbitrary point from } S\}$;
while $|T| < k$ **do** $T \leftarrow T \cup \{\arg\max_{x \in S} d(x, T)\}$;

Sequential coreset construction

Input: Set S, parameter k, parameter ε

$T \leftarrow \{\text{arbitrary point from } S\}$;
while $|T| < k$ **do** $T \leftarrow T \cup \{\arg\max_{x \in S} d(x, T)\}$;
$r_k \leftarrow \max_{x \in S} d(x, T)$;

Sequential coreset construction

Input: Set S, parameter k, parameter ε

$T \leftarrow \{\text{arbitrary point from } S\}$;
while $|T| < k$ **do** $T \leftarrow T \cup \{\arg\max_{x \in S} d(x, T)\}$;
$r_k \leftarrow \max_{x \in S} d(x, T)$;
while $\max_{x \in S} d(x, T) > \frac{\varepsilon}{6} \cdot r_k$ **do**
  $T \leftarrow T \cup \{\arg\max_{x \in S} d(x, T)\}$

Sequential coreset construction

Input: Set S, parameter k, parameter ε

$T \leftarrow \{\text{arbitrary point from } S\}$;
while $|T| < k$ **do** $T \leftarrow T \cup \{\arg\max_{x \in S} d(x,T)\}$;
$r_k \leftarrow \max_{x \in S} d(x,T)$;
while $\max_{x \in S} d(x,T) > \frac{\varepsilon}{6} \cdot r_k$ **do**
  $T \leftarrow T \cup \{\arg\max_{x \in S} d(x,T)\}$

for $t \in T$ **do**
  copy t for each color combination in Γ, with weight 0;

Sequential coreset construction

Input: Set S, parameter k, parameter ε

$T \leftarrow \{\text{arbitrary point from } S\}$;

while $|T| < k$ **do** $T \leftarrow T \cup \{\arg\max_{x \in S} d(x, T)\}$;

$r_k \leftarrow \max_{x \in S} d(x, T)$;

while $\max_{x \in S} d(x, T) > \frac{\varepsilon}{6} \cdot r_k$ **do**

  $T \leftarrow T \cup \{\arg\max_{x \in S} d(x, T)\}$

for $t \in T$ **do**

  copy t for each color combination in Γ, with weight 0;

for $x \in S$ **do**

  $t' \leftarrow \arg\min_{t \in T : col(t) = col(x)} d(x, t)$;

  $w(t') \leftarrow w(t') + 1$;

  $\pi(x) \leftarrow t'$;

return T, w, π;

Properties of the coreset

Proxy radius

Let T be a coreset on S constructed as above, and let π be its proxy function. Then

$$d(x, \pi(x)) \leq \frac{\varepsilon}{3} OPT_{unf} \leq \frac{\varepsilon}{3} OPT_{fair}$$

Size

If S belongs to a metric space with doubling dimension D, then

$$|T| \leq |\Gamma| \cdot k \cdot \left(\frac{12}{\varepsilon}\right)^D$$

Properties of the coreset

Proxy radius

Let T be a coreset on S constructed as above, and let π be its proxy function. Then

$$d(x, \pi(x)) \leq \frac{\varepsilon}{3} OPT_{unf} \leq \frac{\varepsilon}{3} OPT_{fair}$$

Size

If S belongs to a metric space with doubling dimension D, then

$$|T| \leq |\Gamma| \cdot k \cdot \left(\frac{12}{\varepsilon}\right)^D$$

One copy per color

Clusters

Balls covering each k-cluster

A revised linear program, on the coreset

Let $C \subseteq T$ be a set of centers found by GMM *on the coreset* and a radius guess R

How much of the weight of t is assigned to c

Assign all the weight

$$
\begin{aligned}
z_{t,c} &\geq 0 && \substack{t\in T\\ c\in C} \text{ if } d(t,c) \leq R \\
\sum_{c\in C} z_{x,c} &= w(t) && \forall t \in T \\
\beta_\ell \sum_{t\in T} z_{t,c} \leq \sum_{t'\in T_\ell} z_{t',c} &\leq \alpha_\ell \sum_{t\in T} z_{t,c} && \forall c \in C, \ell \in \Gamma
\end{aligned}
$$

A revised linear program, on the coreset

Let $C \subseteq T$ be a set of centers found by GMM *on the coreset* and a radius guess R

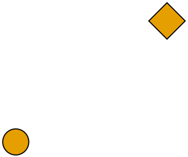
How much of the weight of t is assigned to c

Assign all the weight

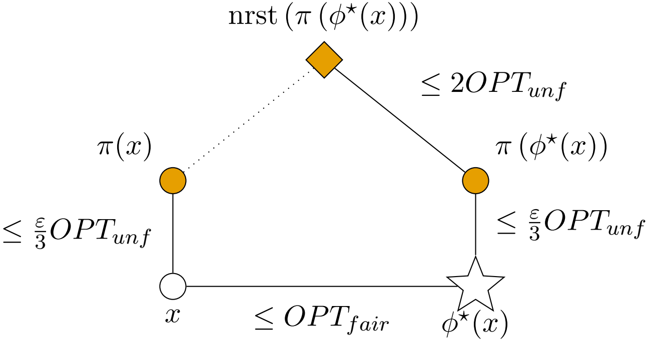
$$
\begin{aligned}
\boxed{z_{t,c}} &\geq 0 && \substack{t \in T \\ c \in C} \text{ if } d(t,c) \leq R \\
\sum_{c \in C} z_{x,c} &= \boxed{w(t)} && \forall t \in T \\
\beta_\ell \sum_{t \in T} z_{t,c} \leq \sum_{t' \in T_\ell} z_{t',c} &\leq \alpha_\ell \sum_{t \in T} z_{t,c} && \forall c \in C, \ell \in \Gamma
\end{aligned}
$$

Which radius guess R allows for a feasible solution?

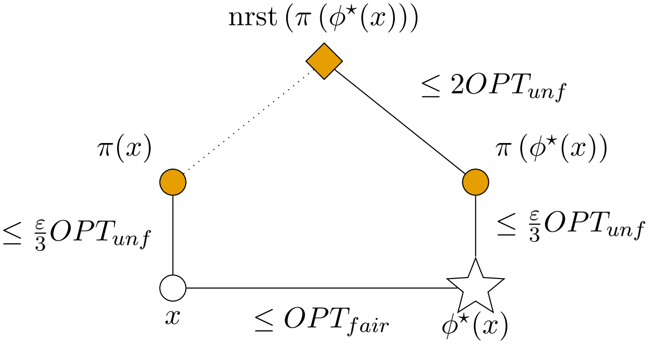
Finding the radius guess



Finding the radius guess



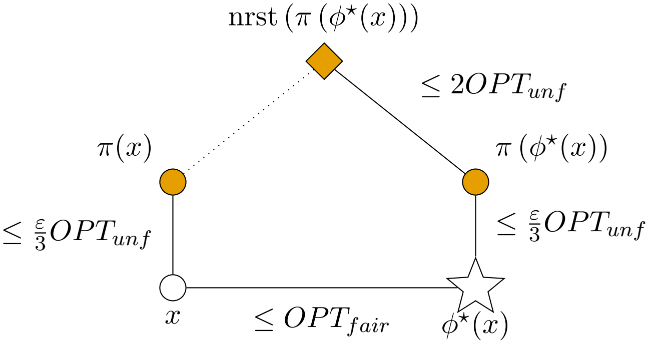
Finding the radius guess



By the triangle inequality, we have that

$$d(\pi(x), \text{nrst}\left(\pi\left(\phi^{\star}(x)\right)\right)) \leq \frac{2\varepsilon}{3}OPT_{fair}$$

Finding the radius guess



Is the assignment fair?

By a charging argument, we can build an assignment of coreset points to centers that respects the fairness constraints.

Summary

- Set $T \subseteq S$
- Proxy function $\pi : S \to T$
- Weight function $w : T \to \mathbb{N}$
- Weight assignment $\hat{\phi}\,(t, c)$, for $t \in T$ and $c \in C \subseteq T$ such that

$$\hat{\phi}\,(t, c) > 0 \quad \Rightarrow \quad d(t, c) \leq \left(3 + \frac{2}{3}\varepsilon\right) OPT_{fair}$$

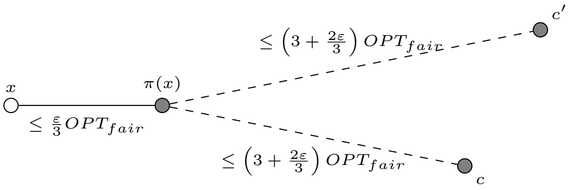
Building the final assignment

Input: The weight distribution $\hat{\phi}(t,c)$, the proxy function $\pi(\cdot)$, the set S, the coreset T, the coreset centers C

for $x \in S$ **do**

- $t \leftarrow \pi(x)$;
- $c \leftarrow$ arbitrary $c \in C : \hat{\phi}(t,c) > 0$;
- $\phi(x) \leftarrow c$;
- $\hat{\phi}(t,c) \leftarrow \hat{\phi}(t,c) - 1$;

return C, ϕ



Summary

Approximation

$$3 + \varepsilon$$

Linear program size

$$\min\{2^{k-1}k|\Gamma|, k \cdot \boxed{|\Gamma| \cdot k \cdot \left(\frac{12}{\varepsilon}\right)^{D}}\}$$

State of the art had n here