Individually Fair Clustering with Outliers

Binita Maity, Shrutimoy Das, Anirban Dasgupta

Scalable Data Science Lab
Computer Science and Engineering
Indian Institute of Technology Gandhinagar, India

Abstract

Individual fairness guarantees are often desirable properties to have, but they become hard to formalize when the dataset contains outliers. Here, we investigate the problem of developing an individually fair k-clustering algorithm for datasets that contain outliers. That is, given n points and k centers, we want that for each point which is not an outlier, there must be a center within $\frac{n}{k}$ nearest neighbours of the given point.

Our Contributions

1. Novel center initialization algorithm, BaseCent, discards fairness-based outliers.
2. Local search-based method for individually fair k-means with outliers.
3. Bound on the number of outliers discarded.
4. $O(1)$ approximation to the (γ, k, m)-fair clustering cost.
5. Empirical validation demonstrating scalability.

Fair radius $r(\cdot)$

Radius of the ball containing the nearest n/k neighbours of a point $v \in X$.

(γ, k, m)*- fair clustering excluding outliers*

Given points X in a metric space (X, d), a k-clustering using S centers and m outliers is (γ, k)-fair if for all $v \in X \setminus Z$, $d(v, S) \leq \gamma r(v)$, where Z is the outlier set ($|Z| \leq m$). Cost:

$$\min_{|S|\leq k, |Z|\leq m} \sum_{v \in X\setminus Z, u \in S} d(v,u)^p \quad \text{s.t. } d(v,S) \leq \gamma r(v)$$

BaseCent Seeding Algorithm

Algorithm 1 *BaseCent*

Input: $X, \delta(\cdot), \gamma$
Output: Anchor points S_0, fairness-based outliers Z_0

$S \leftarrow \emptyset, Z_0 \leftarrow \emptyset, S_0 \leftarrow \emptyset$
while $\exists p \in X : d(p, S) > \gamma\delta(p)$ **do**
  $p^* \leftarrow \arg\min\{\delta(p') : p' \in \{p \in X \mid d(p, S) > \gamma\delta(p)\}\}$
  $S \leftarrow S \cup \{p^*\}$
end while
Discard the last m points as outliers; let Z_0 be discarded set
Let remaining anchors be S_{rem}
$X' \leftarrow X \setminus Z_0$
Let $P = \{P_i\}_{i=1}^{|S_{\text{rem}}|}$, each P_i contains points assigned to anchor zone S^i_{rem}
for $i = 1$ to $|S_{\text{rem}}|$ **do**
  $P_i \leftarrow \emptyset$
end for
if $|S_{\text{rem}}| > k$ **then**
  for each $a \in X'$ **do**
    for $i = 1$ to $|S_{\text{rem}}|$ **do**
      if $d(a, S^i_{\text{rem}}) < (\gamma + 2)\delta(a)$ **then**
        $P_i \leftarrow P_i \cup \{a\}$
      end if
    end for
  end for
  $X'' \leftarrow X'$
  while $X'' \neq \emptyset$ **do**
    $j \leftarrow \arg\max_j |P_j \cap X''|$
    $S_0 \leftarrow S_0 \cup \{S^j_{\text{rem}}\}$
    $X'' \leftarrow X'' \setminus P_j$
    $S_{\text{rem}} \leftarrow S_{\text{rem}} \setminus \{S^j_{\text{rem}}\}$
  end while
else
  $S_0 \leftarrow S_{\text{rem}}$
end if
return S_0, Z_0

How LSFO Works

1. *BaseCent* (Initialization)
Create anchor zones and discard fairness based outliers

2. ConstrainedLS++ (Refinement)
Improve centers per anchor zone

3. Cost Based Outlier Removal
Drop points to significantly improve k-means cost

4. Fairness Preserving Swaps
Feasible swaps to reduce cost

5. Iterate steps $2-4$ Until Convergence

Lemma 1

Suppose $n - m$ points are covered by $k + r$ γ-anchor zones $(0 \leq r \leq m)$, with the m points having the largest fair radii being discarded as fairness-based outliers. Then, there exists a set of k γ'-anchor zones that covers $n - 2m$ points, with $\gamma' = \gamma + 2$.

Approximation Guarantees

LSFO achieves constant-factor approximation under relaxed radius $\gamma' = \gamma + 2$.

Bounding the Number of Outliers

The BaseCent algorithm first discards m fairness-based outliers.
During LSFO, the objective improves by at least a $(1 - \frac{\epsilon}{k})$ factor per iteration.
At most, m additional outliers can be discarded in each iteration.
Thus, after $O\left(\frac{k}{\epsilon}\log(n\Delta)\right)$ iterations.

The total number of outliers is bounded by $|Z| \leq m + \dfrac{mk}{\epsilon}\log(n\Delta)$.

Runtime Analysis

- BaseCent: $O(ndk + nk^2)$
- Constrained LS++: $O\left(\frac{ndk^2}{\epsilon}\log(n\Delta)\right)$
- Swaps (LSFO): $O(n^2k)$ per iteration
- Total LSFO Runtime $O\left(\frac{ndk^3}{\epsilon^2}\log^2(n\Delta) + \frac{n^2k^2}{\epsilon}\log(n\Delta)\right)$

Results

Table 1. : Comparison of cost, fairness bound ratio, and runtime for the Adult and Bank datasets ($40K$ samples). The proposed LSFO method attains superior fairness and efficiency across all values of k.

Dataset	k	m	k-means cost	ρ	Time (sec)
Adult	5	595	$7.55E+04$	1.44	31
	10	707	$4.57E+04$	1.22	242
	15	755	$3.81E+04$	1.07	776
	20	772	$3.30E+04$	1.07	1552
	30	905	$2.60E+04$	1.05	3821
Bank	5	463	$1.82E+04$	2.20	24.29
	10	495	$1.06E+04$	1.59	82.40
	15	725	$6.95E+03$	1.29	316.00
	20	647	$6.06E+03$	1.32	470.00
	30	828	$4.66E+03$	1.20	966.00

Results

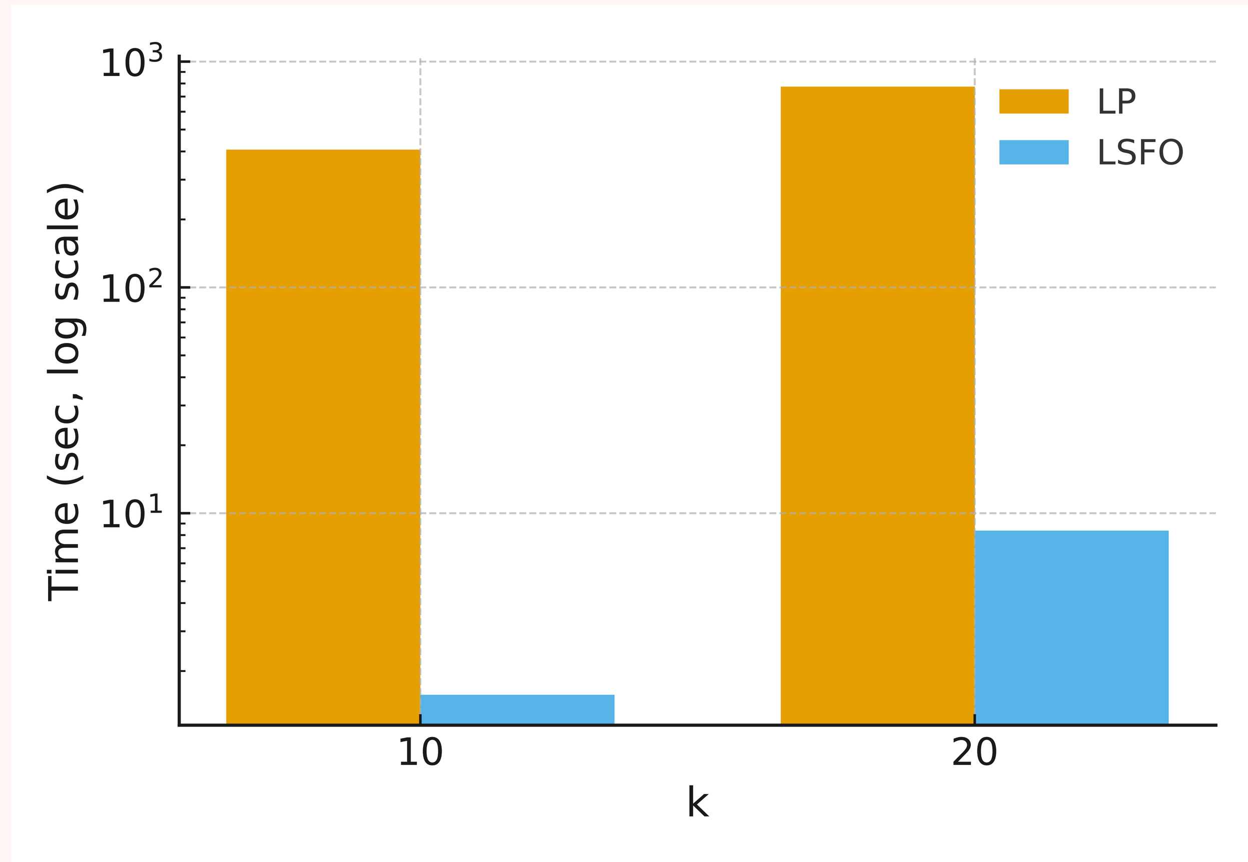

Figure 1. Comparison of runtime for LP and LSFO algorithms at $k = 10, 20$. LSFO is over two orders of magnitude faster than the LP-based approach.

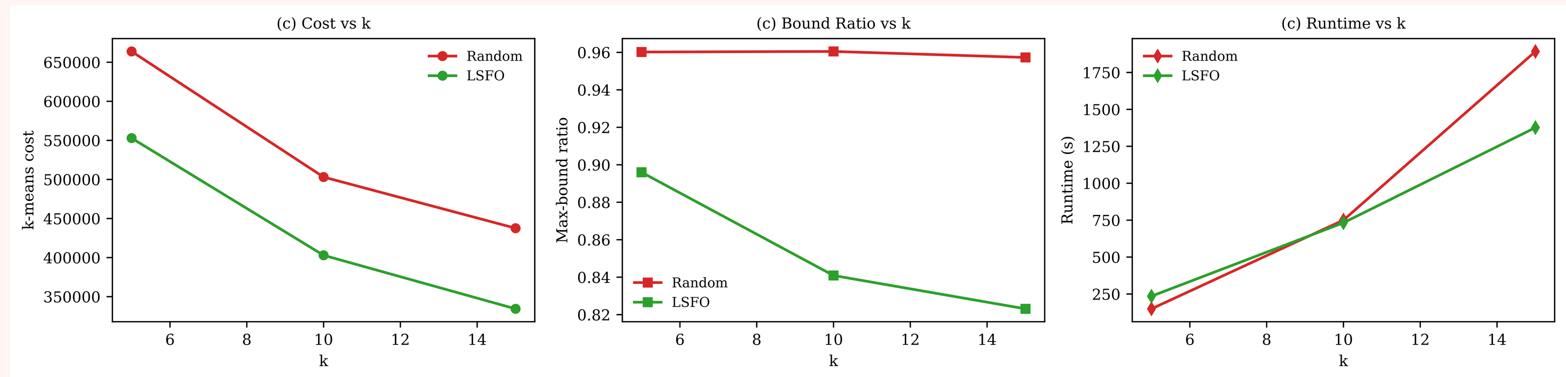

Figure 2. Results on the Covertype dataset with $100k$ points when outliers are discarded randomly vs outliers computed using LSFO.

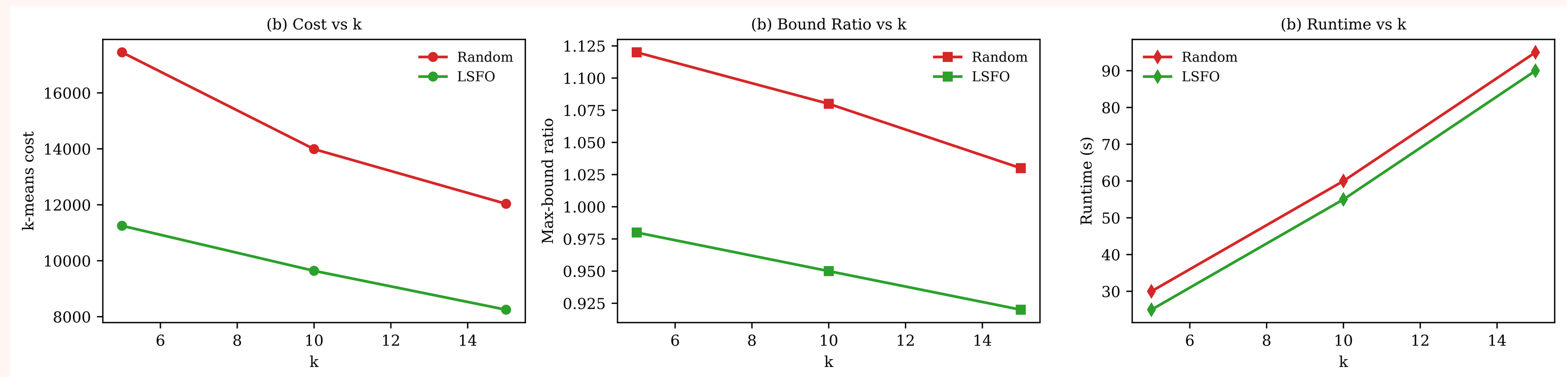

Figure 3. Results on the Adult dataset with $40k$ points when outliers are discarded randomly vs outliers computed using LSFO.

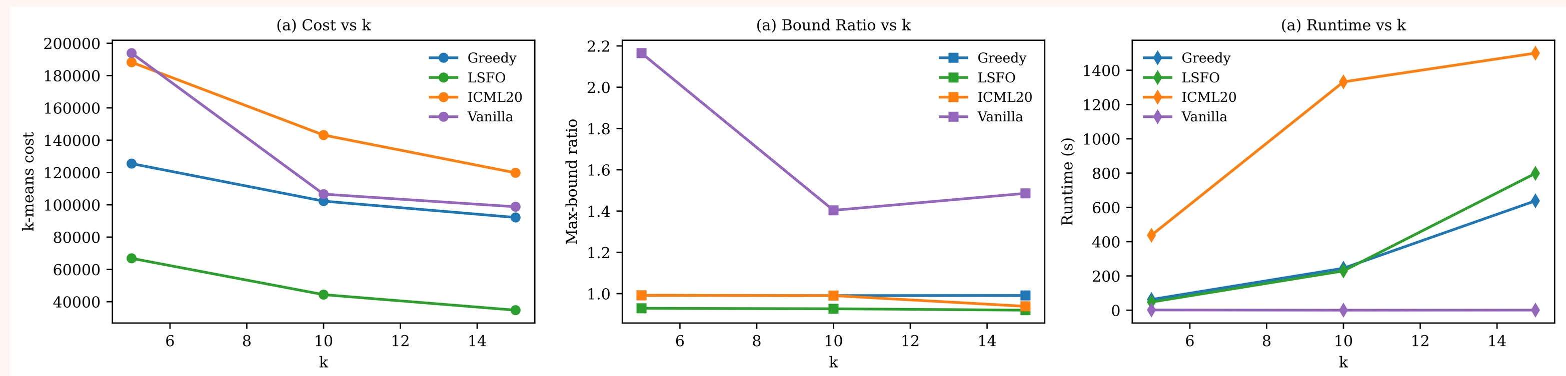

Figure 4. Results on the Adult dataset with 1000 points showing cost, fairness ratio, and runtime comparison across four algorithms.

Observations

- BaseCent initialization effectively identifies fairness-based outliers.
- LSFO achieves a lower k-means cost than prior methods while maintaining fairness.
- The algorithm scales well to large datasets, outperforming LP-based approaches in runtime.
- Carefully discarding outliers improves both fairness and clustering quality.