Symmetric and Asymmetric $k$-center Clustering Under Stability

Colin White
Carnegie Mellon University

Joint work with Nina Balcan and Nika Haghtalab.
(metric) $k$-center Clustering

Choose fire stations, to minimize the maximum travel time to any site.
(metric) $k$-center Clustering

Choose fire stations, to minimize the maximum travel time to any site.

For a set of $|S|=n$ points and distance metric $d$: 
Choose fire stations, to minimize the maximum travel time to any site.

For a set of $|S| = n$ points and distance metric $d$:

• Choose $k$ centers from $S$, assign each point to closest center.
(metric) $k$-center Clustering

Choose fire stations, to minimize the maximum travel time to any site.

For a set of $|S|=n$ points and distance metric $d$:

- Choose $k$ centers from $S$, assign each point to closest center.
- Goal: minimize the maximum radius.
(metric) $k$-center Clustering

Choose fire stations, to minimize the maximum travel time to any site.

For a set of $|S| = n$ points and distance metric $d$:

• Choose $k$ centers from $S$, assign each point to closest center.
• Goal: minimize the maximum radius.

$$ r^* = \min_{c_1, \ldots, c_k} \max_{p \in S} \min_{c_i} d(c_i, p) $$
(metric) $k$-center Clustering

Choose fire stations, to minimize the maximum travel time to any site.

For a set of $|S|=n$ points and distance metric $d$:

- Choose $k$ centers from $S$, assign each point to closest center.
- Goal: minimize the maximum radius.

$$r^* = \min_{c_1, \ldots, c_k} \max_{p \in S} \min_{c_i} d(c_i, p)$$
(metric) \( k \)-center Clustering

Choose fire stations, to minimize the maximum travel time to any site.

For a set of \(|S|=n\) points and distance metric \(d\):

- Choose \(k\) centers from \(S\), assign each point to closest center.
- Goal: minimize the maximum radius.

\[
r^* = \min_{c_1, \ldots, c_k} \max_{p \in S} \min_{c_i} d(c_i, p)
\]
Asymmetric $k$-center Clustering

Relax the condition that $d$ is symmetric
Asymmetric $k$-center Clustering

Relax the condition that $d$ is symmetric

Symmetric: $d(p, q) = d(q, p)$
Asymmetric $k$-center Clustering

Relax the condition that $d$ is symmetric

Symmetric: $d(p, q) = d(q, p)$

Asymmetric: not necessarily true

$$d(p, q) \leq d(p, r) + d(r, q)$$
(still directed $\Delta$-ineq)
Asymmetric $k$-center Clustering

Relax the condition that $d$ is symmetric

Symmetric: $d(p, q) = d(q, p)$

Asymmetric: not necessarily true

$$d(p, q) \leq d(p, r) + d(r, q)$$
(still directed $\Delta$-ineq)

Minimize distance from **Centers** to **points** (order matters now)
Known approximation results
Known approximation results

Most clustering objectives are NP-hard
Known approximation results

Most clustering objectives are NP-hard

[G 1985] Tight 2-approximation for symmetric k-center
Farthest first traversal
Known approximation results

Most clustering objectives are NP-hard

[G 1985] Tight $2$-approximation for symmetric $k$-center
Farthest first traversal

[V 1996] $O(\log^* n)$-approximation for asymmetric $k$-center

\[
n = 2^{2^{2^{\ldots}}} \implies \log^* (n)
\]
Known approximation results

Most clustering objectives are NP-hard

[G 1985] Tight 2-approximation for symmetric k-center
    Farthest first traversal

[V 1996] \( O(\log^* n) \) -approximation for asymmetric k-center

\[
\begin{align*}
n &= 2^{2^{2\cdots}} \\
&= \log^*(n)
\end{align*}
\]

[C et al. 2005] matching lower bound

- First natural problem to have a tight approximation factor not in
  \( O(1) \) or \( \text{polylog}(n) \)
Beyond the worst-case
Beyond the worst-case

- \( \log^* n \) is not desirable in practice
- The NP-hard instances are often contrived and particular
- Theory does not always match up with practice
- E.g. Simplex algorithm and smoothed analysis
Beyond the worst-case

- $\log^*n$ is not desirable in practice
- The NP-hard instances are often contrived and particular
- Theory does not always match up with practice
- E.g. Simplex algorithm and smoothed analysis

Perturbation Resilience

- Small fluctuations don’t change the optimal clustering drastically.
- Captures the uncertainty in data
Beyond the worst-case

- $\log^* n$ is not desirable in practice
- The NP-hard instances are often contrived and particular
- Theory does not always match up with practice
- E.g. Simplex algorithm and smoothed analysis

Perturbation Resilience

- Small fluctuations don’t change the optimal clustering drastically.
- Captures the uncertainty in data
Beyond the worst-case

- $\log^* n$ is not desirable in practice
- The NP-hard instances are often contrived and particular
- Theory does not always match up with practice
- E.g. Simplex algorithm and smoothed analysis

Perturbation Resilience

- Small fluctuations don’t change the optimal clustering drastically.
- Captures the uncertainty in data
Perturbation Resilience

Clustering instance \((S, d)\) is \(\alpha\)-perturbation resilient, if for any function \(d'\) such that

\[\forall p, q \in S, d(p, q) \leq d'(p, q) \leq \alpha d(p, q),\]

the optimal clustering stays the same.

Bilu & Linial ’09:
Perturbation Resilience

Bilu & Linial ’09:

Clustering instance \((S, d)\) is \(\alpha\)-perturbation resilient, if for any function \(d'\) such that

\[ \forall p, q \in S, d(p, q) \leq d'(p, q) \leq \alpha d(p, q), \]

the optimal clustering stays the same.

- Optimal clustering is unique
- It’s ok for centers to change, but not the partition.
- \(d'\) need not satisfy the \(\Delta\)-ineq
Prior Work
Prior Work

- [B L 2009] Exact alg for max cut under $\Omega(\sqrt{n})$-PR
Prior Work

- [B L 2009] Exact alg for max cut under $\Omega(\sqrt{n})$-PR
- [A B S 2010] Exact alg for center-based clustering under 3-PR
Prior Work

- [B L 2009] Exact alg for max cut under $\Omega(\sqrt{n})$-PR
- [A B S 2010] Exact alg for center-based clustering under 3-PR
- [B L 2011] Exact alg for center-based clustering under $1 + \sqrt{2}$-PR
Prior Work

- [B L 2009] Exact alg for max cut under $\Omega(\sqrt{n})$-PR
- [A B S 2010] Exact alg for center-based clustering under 3-PR
- [B L 2011] Exact alg for center-based clustering under $1 + \sqrt{2}$-PR
- [M M V 2014] Exact alg for min multiway cut under 4-PR and max cut clustering under $\Omega(\sqrt{\log n \log \log n})$-PR.
Prior Work

- [B L 2009] Exact alg for max cut under $\Omega(\sqrt{n})$-PR
- [A B S 2010] Exact alg for center-based clustering under 3-PR
- [B L 2011] Exact alg for center-based clustering under $1 + \sqrt{2}$-PR
- [M M V 2014] Exact alg for min multiway cut under 4-PR and max cut clustering under $\Omega(\sqrt{\log n \log \log n})$-PR.
- More results: TSP, Nash Equilibria...
Prior Work

- [B L 2009] Exact alg for max cut under $\Omega(\sqrt{n})$-PR
- [A B S 2010] Exact alg for center-based clustering under 3-PR
- [B L 2011] Exact alg for center-based clustering under $1 + \sqrt{2}$-PR
- [M M V 2014] Exact alg for min multiway cut under 4-PR and max cut clustering under $\Omega(\sqrt{\log n \log \log n})$-PR.
- More results: TSP, Nash Equilibria...

Our results:
- Exact alg for asymmetric k-center under 3-PR
- Exact alg for k-center under 2-PR, tight lower bound
- Robust results
Asymmetric $k$-center under 3-PR

**Theorem:** Polynomial algorithm for AKC under 3-PR.
Asymmetric $k$-center under 3-PR

Theorem: Polynomial algorithm for AKC under 3-PR.

- Idea: “bad” points for which $d(p, c_i) \gg d(c_i, p)$ are hard to deal with
Asymmetric $k$-center under 3-PR

**Theorem:** Polynomial algorithm for AKC under 3-PR.

- Idea: “bad” points for which $d(p, c_i) >> d(c_i, p)$ are hard to deal with
Asymmetric $k$-center under 3-PR

**Theorem:** Polynomial algorithm for AKC under 3-PR.

- Idea: “bad” points for which $d(p, c_i) >> d(c_i, p)$ are hard to deal with
Asymmetric $k$-center under 3-PR

**Theorem**: Polynomial algorithm for AKC under 3-PR.

- Idea: “bad” points for which $d(p, c_i) \gg d(c_i, p)$ are hard to deal with.
Asymmetric $k$-center under 3-PR

**Theorem**: Polynomial algorithm for AKC under 3-PR.

- Idea: “bad” points for which $d(p, c_i) >> d(c_i, p)$ are hard to deal with.
Asymmetric $k$-center under 3-PR

**Theorem:** Polynomial algorithm for AKC under 3-PR.

- Idea: “bad” points for which $d(p, c_i) >> d(c_i, p)$
- are hard to deal with
- Can we find a subset of points that behave “symmetrically”? 

[Diagram of point distribution with red and black points]
Asymmetric \( k \)-center under 3-PR

**Theorem:** Polynomial algorithm for AKC under 3-PR.

- Idea: “bad” points for which \( d(p, c_i) >> d(c_i, p) \) are hard to deal with
- Can we find a subset of points that behave “symmetrically”?

Set of all points \( p \) that the following holds:

For all points \( q \), if \( d(q, p) \leq r^* \) then \( d(p, q) \leq r^* \) as well.

\[
A = \{ p \mid \forall q, \ d(q, p) \leq r^* \implies d(p, q) \leq r^* \}
\]
Properties of Set $A$
Properties of Set $A$

$$A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}$$
Properties of Set \( A \)

\[
A = \{p \mid \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}
\]
Properties of Set $A$

$$A = \{ p \mid \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}$$
Properties of Set $A$

$A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}$
Properties of Set $A$

$$A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}$$

We would like to show that $A$ is representative of the clustering instance
We would like to show that $A$ is representative of the clustering instance.

**Fact 1:** All centers are in $A$. 

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]
Properties of Set $A$

$$A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}$$

We would like to show that $A$ is representative of the clustering instance

**Fact 1:** All centers are in $A$.

**Fact 2:** In $A$: Two points within distance $r^*$ are from the same cluster.
Properties of Set $A$

$$A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}$$

We would like to show that $A$ is representative of the clustering instance

**Fact 1:** All centers are in $A$.

**Fact 2:** In $A$: Two points within distance $r^*$ are from the same cluster.

**Fact 3:** Outside $A$: For $p \not\in A$ and $q \in A$ that has the smallest $d(q, p)$ and $q$ belong to the same cluster.
Algorithm

1. Create the set \( A \).
2. Threshold \( A \) based on \( r^* \).
3. Add edge from \( p \not\in A \) to \( q \in A \) with smallest \( d(q,p) \).
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$. 
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$. 
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$. 
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$. 
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$. 
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q, p)$.
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$. 
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$. 
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$. 
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$. 

Diagram: A collection of points with some edges connecting them, illustrating the algorithm's steps.
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$. 

![Diagram showing a set of points with some points marked as belonging to the set $A$ and others not.]
1. Create the set $A$.

2. Threshold $A$ based on $r^*$.

3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.

✓ Has the centers.
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.

✓ Has the centers.

Fact 1
All centers are in $A$. 

Has the centers.
1. Create the set \( A \).
2. Threshold \( A \) based on \( r^* \).
3. Add edge from \( p \not\in A \) to \( q \in A \) with smallest \( d(q,p) \).

✓ Has the centers.
1. Create the set $A$.

2. Threshold $A$ based on $r^*$.

3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.

✓ Has the centers.
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^\ast$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.

✓ Has the centers.
Algorithm:

1. Create the set $A$.
2. Threshold $A$ based on $r^*$. 
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.

✓ Has the centers.
✓ A center and its points in $A$ have edges.
1. Create the set $A$.

2. Threshold $A$ based on $r^*$.

3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.

**Definition of radius**

Points are at distance $\leq r^*$ from their center.
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.
- No edge between points from two different optimal clusters.
Algorithm

1. Create the set $A$.

2. Threshold $A$ based on $r^*$. 

3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.
- No edge between points from two different optimal clusters.

**Fact 2**

In $A$: Two points $\leq r^*$ are from the same cluster.
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.
- No edge between points from two different optimal clusters.
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.

✓ Has the centers.
✓ A center and its points in $A$ have edges.
✓ No edge between points from two different optimal clusters.
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^*$. 
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.
- No edge between points from two different optimal clusters.
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.
- No edge between points from two different optimal clusters.
- Each set corresponds to one cluster.
**Algorithm**

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.
- No edge between points from two different optimal clusters.
- Each set corresponds to one cluster.

**Fact 3**

$p$ is in the same optimal cluster as the point in $A$ closest to it.
Algorithm

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \notin A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.
- No edge between points from two different optimal clusters.
- Each set corresponds to one cluster.
**Algorithm**

1. Create the set $A$.
2. Threshold $A$ based on $r^*$.
3. Add edge from $p \not\in A$ to $q \in A$ with smallest $d(q,p)$.

- Has the centers.
- A center and its points in $A$ have edges.
- No edge between points from two different optimal clusters.
- Each set corresponds to one cluster.
A Useful Lemma
A Useful Lemma

Lemma

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$. 

82
A Useful Lemma

Lemma

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$. 
A Useful Lemma

Lemma
For all \( p \in C_i \) and \( i \neq j \), \( d(p, c_j) > 2r^* \).

Proof: If not, For all \( q \in C_j \), we have \( d(p, q) \leq 3r^* \).
A Useful Lemma

Lemma
For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*.$

Proof: If not, For all $q \in C_j$, we have $d(p, q) \leq 3r^*.$
**A Useful Lemma**

**Lemma**

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$. 

**Proof:** If not, For all $q \in C_j$, we have $d(p, q) \leq 3r^*$. 

[Diagram showing points $p$ and $q$ with distances marked]
A Useful Lemma

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$.  

**Proof:** If not, for all $q \in C_j$, we have $d(p, q) \leq 3r^*$. 

![Diagram showing points and distances]
A Useful Lemma

Proof: If not, For all \( q \in C_j \), we have \( d(p, q) \leq 3r^* \).
A Useful Lemma

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$.  

Proof: If not, For all $q \in C_j$, we have $d(p, q) \leq 3r^*$.  

3-perturbation:  
Increase all distances by a factor 3, except $d(p, C_j)$, increase by a factor 3 up to $3r^*$.  

A Useful Lemma

Lemma

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$. 

Proof: If not, For all $q \in C_j$, we have $d(p, q) \leq 3r^*$. 

3-perturbation: 

Increase all distances by a factor 3, except $d(p, C_j)$, increase by a factor 3 up to $3r^*$. 

$q$ $p$
A Useful Lemma

Lemma

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$. 

Proof: If not, For all $q \in C_j$, we have $d(p, q) \leq 3r^*$. 

3-perturbation: 

Increase all distances by a factor 3, except $d(p, C_j)$, increase by a factor 3 up to $3r^*$. 

For all $q \in C_j$, $d'(p, q) \leq 3r^*$. 

91
A Useful Lemma

**Lemma**

For all \( p \in C_i \) and \( i \neq j \), \( d(p, c_j) > 2r^* \).

**Proof:** If not, For all \( q \in C_j \), we have \( d(p, q) \leq 3r^* \).

**3-perturbation:**

Increase all distances by a factor 3, except \( d(p, C_j) \), increase by a factor 3 up to \( 3r^* \).

For all \( q \in C_j \), \( d'(p, q) \leq 3r^* \).

\( p \) is a center of \( C_j \) with cost \( \leq 3r^* \).
A Useful Lemma

**Lemma**

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$. 

**Proof:** If not, For all $q \in C_j$, we have $d(p, q) \leq 3r^*$.

3-perturbation:

Increase all distances by a factor 3, except $d(p, C_j)$, increase by a factor 3 up to $3r^*$.

For all $q \in C_j$, $d'(p, q) \leq 3r^*$.

$\rightarrow$ $p$ is a center of $C_j$ with cost $\leq 3r^*$.

$\rightarrow$ The original clustering has cost $\geq 3r^*$. 

A Useful Lemma

For all $p \in C_i$ and $i \neq j$, $d(p, c_j) > 2r^*$. 

**Proof:** If not, For all $q \in C_j$, we have $d(p, q) \leq 3r^*$. 

3-perturbation: 

Increase all distances by a factor 3, except $d(p, C_j)$, increase by a factor 3 up to $3r^*$. 

For all $q \in C_j$, $d'(p, q) \leq 3r^*$. 

$\rightarrow p$ is a center of $C_j$ with cost $\leq 3r^*$. 

$\rightarrow$ The original clustering has cost $\geq 3r^*$. 

Contradicts perturbation resilience.
Proof of Fact 1 & 2
Proof of Fact 1 & 2

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]
Proof of Fact 1 & 2

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in \( A \).
**Proof of Fact 1 & 2**

\[
A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}
\]

**Fact 1:** All centers are in \( A \).
Proof of Fact 1 & 2

\[ A = \{p \mid \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^*\} \]

**Fact 1:** All centers are in \( A \).

All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.
Proof of Fact 1 & 2

$$A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \}$$

Fact 1: All centers are in $A$.

All $q \in C_i$ have $d(c_i, q) \leq r^*$ already.
Proof of Fact 1 & 2

\[ A = \{p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in \( A \).

All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.

All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).
Proof of Fact 1 & 2

\[ A = \{ p \mid \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in A.

All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.

All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In A: Two points \( \leq r^* \) are from the same cluster.
Proof of Fact 1 & 2

\[ A = \{ p \mid \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in \( A \).

- All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.
- All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In \( A \): Two points \( \leq r^* \) are from the same cluster.

If not, there exists \( p, q \) from different clusters in \( A \) such that \( d(p, q) \leq r^* \).
Proof of Fact 1 & 2

\[ A = \{p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in A.

All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.

All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In A: Two points \( \leq r^* \) are from the same cluster.

If not, there exists \( p, q \) from different clusters in A such that \( d(p, q) \leq r^* \).
**Proof of Fact 1 & 2**

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in A.

All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.

All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In A: Two points \( \leq r^* \) are from the same cluster.

If not, there exists \( p, q \) from different clusters in A such that \( d(p, q) \leq r^* \).
Proof of Fact 1 & 2

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in \( A \).

All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.

All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In \( A \): Two points \( \leq r^* \) are from the same cluster.

If not, there exists \( p, q \) from different clusters in \( A \) such that \( d(p, q) \leq r^* \).

All \( q \in A \cap C_j, d(q, c_j) \leq r^* \).
**Proof of Fact 1 & 2**

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in \( A \).

- All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.
- All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In \( A \): Two points \( \leq r^* \) are from the same cluster.

If not, there exists \( p, q \) from different clusters in \( A \) such that \( d(p, q) \leq r^* \).

- All \( q \in A \cap C_j \), \( d(q, c_j) \leq r^* \).
Proof of Fact 1 & 2

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in \( A \).

All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.

All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In \( A \): Two points \( \leq r^* \) are from the same cluster.

If not, there exists \( p, q \) from different clusters in \( A \) such that \( d(p, q) \leq r^* \).

All \( q \in A \cap C_j \), \( d(q, c_j) \leq r^* \).

So,
**Proof of Fact 1 & 2**

\[ A = \{p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in \( A \).

All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.

All \( q \notin C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In \( A \): Two points \( \leq r^* \) are from the same cluster.

If not, there exists \( p, q \) from different clusters in \( A \) such that \( d(p, q) \leq r^* \).

All \( q \in A \cap C_j \), \( d(q, c_j) \leq r^* \).

So, \( d(p, c_j) \leq d(p, q) + d(q, c_j) \leq 2r^* \)
Proof of Fact 1 & 2

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** All centers are in \( A \).

- All \( q \in C_i \) have \( d(c_i, q) \leq r^* \) already.
- All \( q \not\in C_i \) are \( d(q, c_i) \geq 2r^* \) away (by the lemma).

**Fact 2:** In \( A \): Two points \( \leq r^* \) are from the same cluster.

If not, there exists \( p, q \) from different clusters in \( A \) such that \( d(p, q) \leq r^* \).

- All \( q \in A \cap C_j \), \( d(q, c_j) \leq r^* \).
- So, \( d(p, c_j) \leq d(p, q) + d(q, c_j) \leq 2r^* \)

Contradicts the lemma.
Proof of Fact 3

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

Fact 1: If \( p \in C_i \), then \( \arg\min_{q' \in A} d(q', p) \in C_i \)
Proof of Fact 3

\[ A = \{ p | \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** If \( p \in C_i \), then \( \arg\min_{q' \in A} d(q', p) \in C_i \)

Assume \( \exists p \in C_i \) such that \( \arg\min_{q' \in A} d(q', p) = q \notin C_i \)
Proof of Fact 3

\[ A = \{ p \mid \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** If \( p \in C_i \), then \( \arg\min_{q' \in A} d(q', p) \in C_i \)

Assume \( \exists p \in C_i \) such that

\( \arg\min_{q' \in A} d(q', p) = q \notin C_i \)
Proof of Fact 3

\[ A = \{ p \mid \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

**Fact 1:** If \( p \in C_i \), then \( \text{argmin}_{q' \in A} d(q', p) \in C_i \)

Assume \( \exists p \in C_i \) such that

\( \text{argmin}_{q' \in A} d(q', p) = q \notin C_i \)

Construct an \( \alpha \)-perturbation in which

\( q \) becomes the center of \( C_j \)

\( q \) must capture \( p \)
Proof of Fact 3

\[ A = \{ p \mid \forall q, d(q, p) \leq r^* \implies d(p, q) \leq r^* \} \]

Fact 1: If \( p \in C_i \), then \( \arg\min_{q' \in A} d(q', p) \in C_i \)

Assume \( \exists p \in C_i \) such that \( \arg\min_{q' \in A} d(q', p) = q \notin C_i \)

Construct an \( \alpha \)-perturbation in which \( q \) becomes the center of \( C_j \)

\( q \) must capture \( p \)

Contradiction
AKC vs $k$-center
AKC vs $k$-center

✓ Polytime algorithm for AKC under 3-PR.
AKC vs $k$-center

✓ Polytime algorithm for AKC under 3-PR.
   Also: Polytime algorithm for $k$-center under 2-PR
AKC vs $k$-center

✓ Polytime algorithm for AKC under 3-PR.
   Also: Polytime algorithm for $k$-center under 2-PR

• Both have a (2-\(\varepsilon\))-PR lower bound
AKC vs $k$-center

- Polytime algorithm for AKC under 3-PR.
  Also: Polytime algorithm for $k$-center under 2-PR
- Both have a $(2-\varepsilon)$-PR lower bound

In fact, AKC and $k$-center are equivalent in difficulty under 2-approximation stability
Lower Bounds

Hardness:

No polynomial time algorithm for symmetric $k$-center under $(2-\varepsilon)$-perturbation resilience, unless NP=RP.
Lower Bounds

**Hardness:**

No polynomial time algorithm for symmetric $k$-center under $(2-\varepsilon)$-perturbation resilience, unless NP=RP.

Reduction by unambiguous-perfect dominating set, used to show $(2-\varepsilon)$-center proximity is NP-hard.
No polynomial time algorithm for symmetric $k$-center under $(2-\varepsilon)$-perturbation resilience, unless NP=RP.

Reduction by unambiguous-perfect dominating set, used to show $(2-\varepsilon)$-center proximity is NP-hard

Perfect: each vertex hit by exactly one dominator
- NP-Hard [BR’14]
Lower Bounds

Hardness:

No polynomial time algorithm for \textit{symmetric} \(k\)-center under \((2-\varepsilon)\)-perturbation resilience, unless \(\text{NP}=\text{RP}\).

Reduction by unambiguous-perfect dominating set, used to show \((2-\varepsilon)\)-center proximity is \(\text{NP}\)-hard

Perfect: each vertex hit by exactly one dominator
• \(\text{NP-Hard} \ [\text{BR’14}]\)

Unambiguous: at most one solution
• \(\text{U3SAT} \text{ is hard unless } \text{NP}=\text{RP} \ [\text{VV’86}]\)
Robust Stability Conditions
Robust Stability Conditions

$\alpha$-perturbation resilience:
Optimal clustering does not change under $\alpha$-perturbations.
Robust Stability Conditions

$\alpha$-perturbation resilience:
Optimal clustering does not change under $\alpha$-perturbations.

Robust: $(\alpha, \varepsilon)$-perturbation resilience
Optimal clustering changes by $\leq \varepsilon$ under $\alpha$-perturbations.
Robust Stability Conditions

\( \alpha \)-perturbation resilience:
Optimal clustering does not change under \( \alpha \)-perturbations.

Robust: \((\alpha, \varepsilon)\)-perturbation resilience
Optimal clustering changes by \( \leq \varepsilon \) under \( \alpha \)-perturbations.

Results:
• Single linkage finds exact k-center under \((4, \varepsilon)\)- PR.
• More results for robust approximation stability.
Conclusion

• Polytime alg for AKC under 3-PR
• Polytime alg for $k$-center under 2-PR, tight

Theoretical Significance
• First time a problem with no constant factor approximation has an exact algorithm, when assuming just constant stability
• First tight results in this area
• Symmetric and asymmetric become nearly same difficulty

Practical Significance
• Only a small window of values for which perturbation resilience is interesting
Open Questions

• $\alpha, \varepsilon$-PR for asymmetric k-center
• Gap between $(2-\varepsilon)$-PR hardness and 3-PR for asymmetric k-center
• Can we also get 2-PR for $k$-median and $k$-means??

Questions?