How Many Clusters: An algorithmic answer

Chiranjib Bhattacharyya, Ravi Kannan, Amit Kumar

Set-Up, Notation

- Clustering Problem: Given n points in $\mathbf{R}^d$ with

Set-Up, Notation

- Clustering Problem: Given n points in $\mathbf{R}^d$ with
  - Promise: $\exists$ “nice” “Ground Truth (GT) ‘’ clustering $C_1, C_2, \ldots, C_k$.

Set-Up, Notation

- Clustering Problem: Given n points in $\mathbf{R}^d$ with
  - Promise: $\exists$ "nice" "Ground Truth (GT) '' clustering $C_1, C_2, \ldots, C_k$.
  - Find k exactly and the GT approximately.

Set-Up, Notation

- Clustering Problem: Given n points in $\mathbf{R}^d$ with
  - Promise: $\exists$ "nice" "Ground Truth (GT) '' clustering $C_1, C_2, \ldots, C_k$.
  - Find k exactly and the GT approximately.
- Examples: Data generated by mixtures.

Set-Up, Notation

- Clustering Problem: Given n points in $\mathbf{R}^d$ with
  - Promise: $\exists$ “nice” “Ground Truth (GT) ‘’ clustering $C_1, C_2, \ldots, C_k$.
  - Find k exactly and the GT approximately.
- Examples: Data generated by mixtures.
- Our results are general. Don’t assume stochastic model,

Set-Up, Notation

- Clustering Problem: Given n points in $\mathbf{R}^d$ with
  - Promise: $\exists$ "nice" "Ground Truth (GT) '' clustering $C_1, C_2, \ldots, C_k$.
  - Find k exactly and the GT approximately.
- Examples: Data generated by mixtures.
- Our results are general. Don't assume stochastic model,
- but they subsume Gaussian, log-concave ... mixtures.

Set-Up, Notation

- Clustering Problem: Given n points in $\mathbf{R}^d$ with
  - Promise: $\exists$ "nice" "Ground Truth (GT) '' clustering $C_1, C_2, \ldots, C_k$.
  - Find k exactly and the GT approximately.
- Examples: Data generated by mixtures.
- Our results are general. Don't assume stochastic model,
- but they subsume Gaussian, log-concave ... mixtures.
- Previous (provable) poly time algorithms, even for special mixtures, assumed k given.

Set-Up, Notation

- Clustering Problem: Given n points in $\mathbf{R}^d$ with
  - Promise: $\exists$ "nice" "Ground Truth (GT) '' clustering $C_1, C_2, \ldots, C_k$.
  - Find k exactly and the GT approximately.
- Examples: Data generated by mixtures.
- Our results are general. Don't assume stochastic model,
- but they subsume Gaussian, log-concave ... mixtures.
- Previous (provable) poly time algorithms, even for special mixtures, assumed k given.
- **Notation** Set S of points (in $\mathbf{R}^d$), $\mu(S), \sigma(S)$ are respy their mean and (maximum) standard deviation (in a 1-d projection).

Importance of knowing k

- An objective function like $k-$means, $k-$center, ...can be defined

Importance of knowing k

- An objective function like $k-$means, $k-$center, ...can be defined
  - Formulation as optimization problem.

Importance of knowing k

- An objective function like $k-$means, $k-$center, ...can be defined
  - Formulation as optimization problem.
  - Led to elegant algorithms/heuristics, eg. Lloyd's for $k-$means.

Importance of knowing k

- An objective function like $k-$means, $k-$center, ...can be defined
  - Formulation as optimization problem.
  - Led to elegant algorithms/heuristics, eg. Lloyd's for $k-$means.
  - Exact k needed. k dictates what cluster is.
- Many algorithms work in $k-$dim. SVD projection.
  - For mixture of k Gaussians, $k-$SVD projection "works". Vempala, Wang; Achlioptas, McSherry; Kannan, Salmasian, Vempala
  - Even without a mixture model, under "proximity" and given k, SVD works.Kumar, Kannan; Awashti, Sheffet

Importance of knowing k

- An objective function like $k-$means, $k-$center, ...can be defined
  - Formulation as optimization problem.
  - Led to elegant algorithms/heuristics, eg. Lloyd's for $k-$means.
  - Exact k needed. k dictates what cluster is.
- Many algorithms work in $k-$dim. SVD projection.
  - For mixture of k Gaussians, $k-$SVD projection "works". Vempala, Wang; Achlioptas, McSherry; Kannan, Salmasian, Vempala
  - Even without a mixture model, under "proximity" and given k, SVD works.Kumar, Kannan; Awashti, Sheffet
- Distance based Clustering: Points at distance $\leq \tau$ from data point are "cluster-mates" . Need τ, don't need k. Beware:

Importance of knowing k

- An objective function like $k-$means, $k-$center, ...can be defined
  - Formulation as optimization problem.
  - Led to elegant algorithms/heuristics, eg. Lloyd's for $k-$means.
  - Exact k needed. k dictates what cluster is.
- Many algorithms work in $k-$dim. SVD projection.
  - For mixture of k Gaussians, $k-$SVD projection "works". Vempala, Wang; Achlioptas, McSherry; Kannan, Salmasian, Vempala
  - Even without a mixture model, under "proximity" and given k, SVD works.Kumar, Kannan; Awashti, Sheffet
- Distance based Clustering: Points at distance $\leq \tau$ from data point are "cluster-mates" . Need τ, don't need k. Beware:
  - For mixture of 2 std Gaussians, means $\Omega^*(1)$ apart, no τ works!

Importance of knowing k

- An objective function like $k-$means, $k-$center, ...can be defined
  - Formulation as optimization problem.
  - Led to elegant algorithms/heuristics, eg. Lloyd's for $k-$means.
  - Exact k needed. k dictates what cluster is.
- Many algorithms work in $k-$dim. SVD projection.
  - For mixture of k Gaussians, $k-$SVD projection "works". Vempala, Wang; Achlioptas, McSherry; Kannan, Salmasian, Vempala
  - Even without a mixture model, under "proximity" and given k, SVD works.Kumar, Kannan; Awashti, Sheffet
- Distance based Clustering: Points at distance $\leq \tau$ from data point are "cluster-mates" . Need τ, don't need k. Beware:
  - For mixture of 2 std Gaussians, means $\Omega^*(1)$ apart, no τ works!
- Special methods for GMM's: Kalai, Moitra, Valiant; Regev, Vijayaraghavan; Kwan, Caramanis;... need k.

Nice GT: Means Separated by X Standard Deviations

- Mean-Separation Assumption $\forall C_\ell$ in GT,

Nice GT: Means Separated by X Standard Deviations

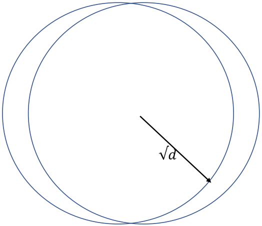
- Mean-Separation Assumption $\forall C_\ell$ in GT,
  - mean of any other cluster is $\Omega^*(\sigma(C_\ell))$ away:

Nice GT: Means Separated by X Standard Deviations

- Mean-Separation Assumption $\forall C_\ell$ in GT,
  - mean of any other cluster is $\Omega^*(\sigma(C_\ell))$ away:
  - $|\mu(C_\ell) - \mu(C_{\ell'}| \geq \sigma(C_\ell)\frac{\log^4 n}{w_0^c}$, w_0 = Min weight of a cluster.

Nice GT: Means Separated by X Standard Deviations

- Mean-Separation Assumption $\forall C_\ell$ in GT,
  - mean of any other cluster is $\Omega^*(\sigma(C_\ell))$ away:
  - $|\mu(C_\ell) - \mu(C_{\ell'}| \geq \sigma(C_\ell)\frac{\log^4 n}{w_0^c}$, w_0 = Min weight of a cluster.
- 2 Gaussians $\Omega(\sigma)$ apart. In most 1-d projections close to $k = 1$.



Infer k from plot of $k-$means cost?

- An often-used criterion for finding k: Plot k versus the optimal $k-$means cost.

Infer k from plot of $k-$means cost?

- An often-used criterion for finding k: Plot k versus the optimal $k-$means cost.
- Take biggest drop. Elbow Method Hartigan; Milligan, Cooper. No proof in the generality of set-up here.

Infer k from plot of k−means cost?

- An often-used criterion for finding k: Plot k versus the optimal k−means cost.
- Take biggest drop. Elbow Method Hartigan; Milligan, Cooper. No proof in the generality of set-up here.
- Under assumption of a big drop, can find clustering: Ostravsky, Rabani, Schulman, Swamy. Such big drops not present in general under our setting.

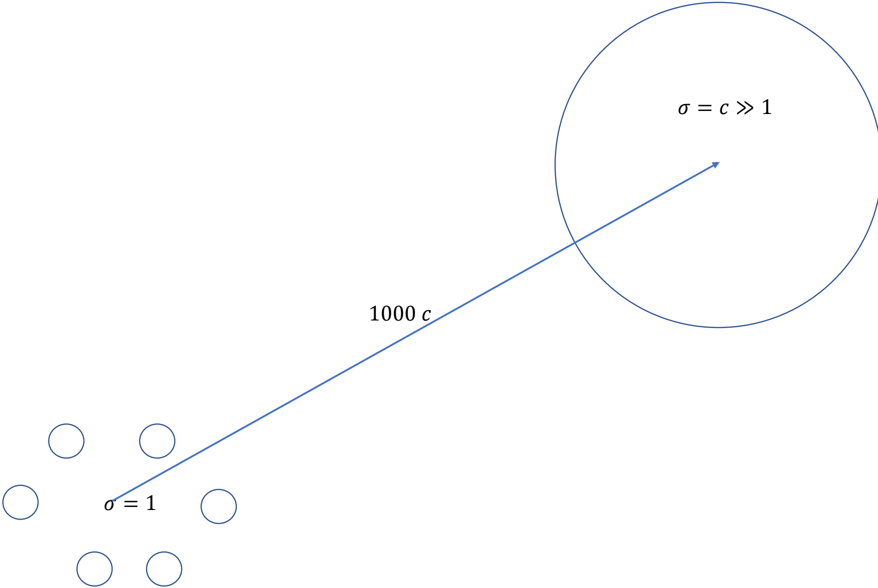
Infer k from plot of $k-$means cost?

- An often-used criterion for finding k: Plot k versus the optimal $k-$means cost.
- Take biggest drop. Elbow Method Hartigan; Milligan, Cooper. No proof in the generality of set-up here.
- Under assumption of a big drop, can find clustering: Ostravsky, Rabani, Schulman, Swamy. Such big drops not present in general under our setting.
- Gap Statistic Take k with good ratio of $k-$means cost to 1-means under a prior. Tibshirani; . Proofs for special cases.

Infer k from plot of $k-$means cost?

- An often-used criterion for finding k: Plot k versus the optimal $k-$means cost.
- Take biggest drop. Elbow Method Hartigan; Milligan, Cooper. No proof in the generality of set-up here.
- Under assumption of a big drop, can find clustering: Ostravsky, Rabani, Schulman, Swamy. Such big drops not present in general under our setting.
- Gap Statistic Take k with good ratio of $k-$means cost to 1-means under a prior. Tibshirani; . Proofs for special cases.
- Even for spherical Gaussian mixtures, not true that largest drop occurs at correct k. In fact, for every m, there is a $m-$component spherical GMM with largest drop at $k = 2$

k in spherical Gaussian Mixtures



Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.
- Usual Fix for identifiability of k: Assume k given.

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.
- Usual Fix for identifiability of k: Assume k given.
- Theoretically, k being given helps:

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.
- Usual Fix for identifiability of k: Assume k given.
- Theoretically, k being given helps:
  - Define objective functions, eg $k-$means. Reduces to Optimization.

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.
- Usual Fix for identifiability of k: Assume k given.
- Theoretically, k being given helps:
  - Define objective functions, eg $k-$means. Reduces to Optimization.
  - Indirectly define what a cluster is.

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.
- Usual Fix for identifiability of k: Assume k given.
- Theoretically, k being given helps:
  - Define objective functions, eg $k-$means. Reduces to Optimization.
  - Indirectly define what a cluster is.
  - Also: $n^{1-\varepsilon}$ approximation of k is NP-hard.

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.
- Usual Fix for identifiability of k: Assume k given.
- Theoretically, k being given helps:
  - Define objective functions, eg $k-$means. Reduces to Optimization.
  - Indirectly define what a cluster is.
  - Also: $n^{1-\varepsilon}$ approximation of k is NP-hard.
- Practically: How do we know k?

Conditions for “nice” GT

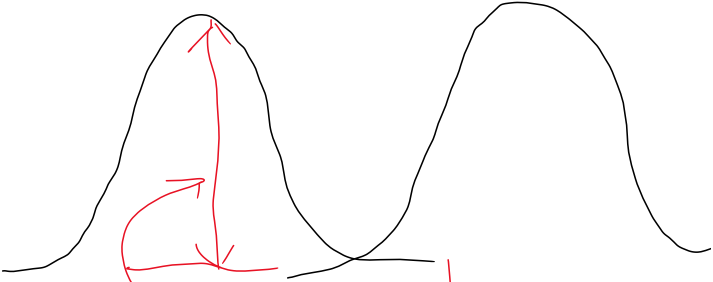
- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.
- Usual Fix for identifiability of k: Assume k given.
- Theoretically, k being given helps:
    - Define objective functions, eg $k-$means. Reduces to Optimization.
    - Indirectly define what a cluster is.
    - Also: $n^{1-\varepsilon}$ approximation of k is NP-hard.
- Practically: How do we know k?
- Worse: In non-TCS talks, someone always asks question. My answer so far: vigorous handwaving

Conditions for “nice” GT

- $k = n$ consistent with Mean Sep. Need more than Mean-Sep
- Add Min Wt. assumption: $w_0 \geq n^{-.01}$.
- Still k not identifiable from data. Eg. $k = 1$ works.
- Usual Fix for identifiability of k: Assume k given.
- Theoretically, k being given helps:
  - Define objective functions, eg $k-$means. Reduces to Optimization.
  - Indirectly define what a cluster is.
  - Also: $n^{1-\varepsilon}$ approximation of k is NP-hard.
- Practically: How do we know k?
- Worse: In non-TCS talks, someone always asks question. My answer so far: vigorous handwaving
- Here: A proper answer.

Baby Eg.: $d = 1$, one or two Gaussians

$d = 1$ GT mixture of one or 2 Gaussians satisfying Mean Sep and Min



$$\int pdf \gg \frac{1}{\sqrt{2\pi}\,\sigma}$$

wt.

Anti-Concentration: Last Condition for “Nice”

- Set S of points in $\mathbf{R}^d$ satisfies **anti-concentration** if

Anti-Concentration: Last Condition for “Nice”

- Set S of points in $\mathbf{R}^d$ satisfies **anti-concentration** if
  - $\forall$ lines L, $\forall$ intervals $I, |I| \geq \varepsilon\sigma(S^L)$,

Anti-Concentration: Last Condition for “Nice”

- Set S of points in $\mathbf{R}^d$ satisfies **anti-concentration** if
  - $\forall$ lines L, $\forall$ intervals $I, |I| \geq \varepsilon\sigma(S^L)$,
    - $|S^L \cap I| \leq c|I|\sigma(S^L)$. ($x^L$ is proj on L.)

Anti-Concentration: Last Condition for "Nice"

- Set S of points in $\mathbf{R}^d$ satisfies **anti-concentration** if
  - $\forall$ lines L, $\forall$ intervals $I, |I| \geq \varepsilon\sigma(S^L)$,
    - $|S^L \cap I| \leq c|I|\sigma(S^L)$. ($x^L$ is proj on L.)
- **Clustering is nice if each cluster has anti-concentration and Mean Sep, Min Wt. hold.**

Anti-Concentration: Last Condition for "Nice"

- Set S of points in $\mathbf{R}^d$ satisfies **anti-concentration** if
  - $\forall$ lines L, $\forall$ intervals $I, |I| \geq \varepsilon\sigma(S^L)$,
    - $|S^L \cap I| \leq c|I|\sigma(S^L)$. ($x^L$ is proj on L.)
- **Clustering is nice if each cluster has anti-concentration and Mean Sep, Min Wt. hold.**
- Identifiability Theorem $\mathscr{C}_1, \mathscr{C}_2$ 2 nice clusterings of data $\implies \#(\mathscr{C}_1) = \#(\mathscr{C}_2)$ and distance$(\mathscr{C}_1, \mathscr{C}_2) \leq \varepsilon n$. (Proof 3 pages)

Anti-Concentration: Last Condition for "Nice"

- Set S of points in $\mathbf{R}^d$ satisfies **anti-concentration** if
  - $\forall$ lines L, $\forall$ intervals $I, |I| \geq \varepsilon\sigma(S^L)$,
    - $|S^L \cap I| \leq c|I|\sigma(S^L)$. ($x^L$ is proj on L.)
- **Clustering is nice if each cluster has anti-concentration and Mean Sep, Min Wt. hold.**
- Identifiability Theorem $\mathscr{C}_1, \mathscr{C}_2$ 2 nice clusterings of data $\implies \#(\mathscr{C}_1) = \#(\mathscr{C}_2)$ and distance$(\mathscr{C}_1, \mathscr{C}_2) \leq \varepsilon n$. (Proof 3 pages)
- Theorem says: k is pinned down by data (no additional info needed)

Anti-Concentration: Last Condition for "Nice"

- Set S of points in $\mathbf{R}^d$ satisfies **anti-concentration** if
  - $\forall$ lines L, $\forall$ intervals $I, |I| \geq \varepsilon\sigma(S^L)$,
    - $|S^L \cap I| \leq c|I|\sigma(S^L)$. ($x^L$ is proj on L.)
- **Clustering is nice if each cluster has anti-concentration and Mean Sep, Min Wt. hold.**
- Identifiability Theorem $\mathscr{C}_1, \mathscr{C}_2$ 2 nice clusterings of data $\implies \#(\mathscr{C}_1) = \#(\mathscr{C}_2)$ and distance$(\mathscr{C}_1, \mathscr{C}_2) \leq \varepsilon n$. (Proof 3 pages)
- Theorem says: k is pinned down by data (no additional info needed)
- Paper also has poly time algorithm to find exact k, approx. nice clustering (assuming exists).

Anti-Concentration: Last Condition for “Nice”

- Set S of points in $\mathbf{R}^d$ satisfies **anti-concentration** if
  - $\forall$ lines L, $\forall$ intervals $I, |I| \geq \varepsilon\sigma(S^L)$,
    - $|S^L \cap I| \leq c|I|\sigma(S^L)$. ($x^L$ is proj on L.)
- **Clustering is nice if each cluster has anti-concentration and Mean Sep, Min Wt. hold.**
- Identifiability Theorem $\mathscr{C}_1, \mathscr{C}_2$ 2 nice clusterings of data $\implies \#(\mathscr{C}_1) = \#(\mathscr{C}_2)$ and distance$(\mathscr{C}_1, \mathscr{C}_2) \leq \varepsilon n$. (Proof 3 pages)
- Theorem says: k is pinned down by data (no additional info needed)
- Paper also has poly time algorithm to find exact k, approx. nice clustering (assuming exists).
- Considerably harder. Here, some intuition/ideas behind it.

No Large Sub-Cluster (NLSC) Property

- A subset S of data points has NLSC if:

No Large Sub-Cluster (NLSC) Property

- A subset S of data points has NLSC if:
  - $\forall T \subseteq S, |T| \geq c\sqrt{n}, \sigma(T) \geq \frac{|T|}{12|S|}\sigma(S)$.

No Large Sub-Cluster (NLSC) Property

- A subset S of data points has NLSC if:
  - $\forall T \subseteq S, |T| \geq c\sqrt{n}, \sigma(T) \geq \frac{|T|}{12|S|}\sigma(S)$.
- **Lemma** Anti-concentration implies NLSC.

Main Theorem

- Main Theorem There is a poly time alg. which given data with a GT clustering $\{C_1, C_2, \ldots, C_k\}$, satisfying

Main Theorem

- Main Theorem There is a poly time alg. which given data with a GT clustering $\{C_1, C_2, \ldots, C_k\}$, satisfying
  - Mean Sep

Main Theorem

- Main Theorem There is a poly time alg. which given data with a GT clustering $\{C_1, C_2, \ldots, C_k\}$, satisfying
  - Mean Sep
  - Min Wt. and

Main Theorem

- Main Theorem There is a poly time alg. which given data with a GT clustering $\{C_1, C_2, \ldots, C_k\}$, satisfying
  - Mean Sep
  - Min Wt. and
  - each C_ℓ satisfying) NLSC

Main Theorem

- Main Theorem There is a poly time alg. which given data with a GT clustering $\{C_1, C_2, \ldots, C_k\}$, satisfying
  - Mean Sep
  - Min Wt. and
  - each C_ℓ satisfying) NLSC
- finds k exactly and GT approximately.

Main Theorem

- Main Theorem There is a poly time alg. which given data with a GT clustering $\{C_1, C_2, \ldots, C_k\}$, satisfying
    - Mean Sep
    - Min Wt. and
    - each C_ℓ satisfying) NLSC
- finds k exactly and GT approximately.
- **Corollary** For stochastic mixture of pdf's, each pdf satisfying anti-concentration, mixture satisfying Mean Sep. and Min wt. can find k exactly and GT approximately provided number of samples is at least $O^*(1/w_0)$ times max no. of samples needed to learn mean and Std. Dev. of a single component.
- **Corollary** For any log-concave mixture satisfying Mean Sep and Min Wt., the algorithm finds k exactly and GT approximately.

First Cut: Data $+w_0$ are given

1. Set S of data which minimizes $\sigma(S)$ subject to $|S| = w_0 n$ essentially contained in single C_ℓ. Call S "nucleus" of C_ℓ

First Cut: Data $+w_0$ are given

1. Set S of data which minimizes $\sigma(S)$ subject to $|S| = w_0 n$ essentially contained in single C_ℓ. Call S "nucleus" of C_ℓ
2. An Attempt: Find S with $|S| = w_0 n$ minimizing $\sigma(S)$.

First Cut: Data $+w_0$ are given

1. Set S of data which minimizes $\sigma(S)$ subject to $|S| = w_0 n$ essentially contained in single C_ℓ. Call S "nucleus" of C_ℓ
2. An Attempt: Find S with $|S| = w_0 n$ minimizing $\sigma(S)$.
3. Peel S off, repeat. Next S may be from same or different cluster.

First Cut: Data $+w_0$ are given

1. Set S of data which minimizes $\sigma(S)$ subject to $|S| = w_0 n$ essentially contained in single C_ℓ. Call S "nucleus" of C_ℓ
2. An Attempt: Find S with $|S| = w_0 n$ minimizing $\sigma(S)$.
3. Peel S off, repeat. Next S may be from same or different cluster.
4. If S has $\mu(S)$ within $O^*(\sigma(S') + \sigma(S))$ of an already found S', S, S' (we prove) are (essentially) from same C_ℓ, so, discard S.

First Cut: Data $+w_0$ are given

1. Set S of data which minimizes $\sigma(S)$ subject to $|S| = w_0 n$ essentially contained in single C_ℓ. Call S "nucleus" of C_ℓ
2. An Attempt: Find S with $|S| = w_0 n$ minimizing $\sigma(S)$.
3. Peel S off, repeat. Next S may be from same or different cluster.
4. If S has $\mu(S)$ within $O^*(\sigma(S') + \sigma(S))$ of an already found S', S, S' (we prove) are (essentially) from same C_ℓ, so, discard S.
5. At termination, we prove k S's (one nucleus per cluster) left.

First Cut: Data $+w_0$ are given

1. Set S of data which minimizes $\sigma(S)$ subject to $|S| = w_0 n$ essentially contained in single C_ℓ. Call S “nucleus” of C_ℓ
2. An Attempt: Find S with $|S| = w_0 n$ minimizing $\sigma(S)$.
3. Peel S off, repeat. Next S may be from same or different cluster.
4. If S has $\mu(S)$ within $O^*(\sigma(S') + \sigma(S))$ of an already found S', S, S' (we prove) are (essentially) from same C_ℓ, so, discard S.
5. At termination, we prove k S's (one nucleus per cluster) left.
6. The nuclii can be used also to grow all clusters.

First Cut: Data $+w_0$ are given

1. Set S of data which minimizes $\sigma(S)$ subject to $|S| = w_0 n$ essentially contained in single C_ℓ. Call S "nucleus" of C_ℓ
2. An Attempt: Find S with $|S| = w_0 n$ minimizing $\sigma(S)$.
3. Peel S off, repeat. Next S may be from same or different cluster.
4. If S has $\mu(S)$ within $O^*(\sigma(S') + \sigma(S))$ of an already found S', S, S' (we prove) are (essentially) from same C_ℓ, so, discard S.
5. At termination, we prove k S's (one nucleus per cluster) left.
6. The nuclii can be used also to grow all clusters.
7. All Steps use quantities determined by DATA+w_0. Except step 2, all doable with some technical work. Focus now on Step 2.

Minimizing σ among subsets of given size

- Is NP-hard to solve exactly.

Minimizing σ among subsets of given size

- Is NP-hard to solve exactly.
- But as a bi-criterion problem (size of set, σ of set), we show can be solved approximately. Gives a poly time alg to find k given data + w_0. Stepping stone to (harder) problem of finding k given only data.

Minimizing σ among subsets of given size

- Is NP-hard to solve exactly.
- But as a bi-criterion problem (size of set, σ of set), we show can be solved approximately. Gives a poly time alg to find k given data + w_0. Stepping stone to (harder) problem of finding k given only data.
- Namely, if there is a set of cardinality αn with σ, then we show in poly time, can find a set S of cardinality $\alpha^2 n/12$ with $\sigma(S) \leq$(polylog) σ/α^3.

Minimizing σ among subsets of given size

- Is NP-hard to solve exactly.
- But as a bi-criterion problem (size of set, σ of set), we show can be solved approximately. Gives a poly time alg to find k given data + w_0. Stepping stone to (harder) problem of finding k given only data.
- Namely, if there is a set of cardinality αn with σ, then we show in poly time, can find a set S of cardinality $\alpha^2 n/12$ with $\sigma(S) \leq$(polylog) σ/α^3.
- Via Semi-Definite-Programming relaxation Plus Rounding.

Spectrally Tight Subsets: An Independent problem

- **Problem** Given set X of n points in $\mathbf{R}^d$ and $\alpha \in (0,1)$, find $S \subseteq X, |S| = \alpha n$ minimizing $||A^S||$, where, $A^S = \{x - \mu(S) : x \in S\}$.

Spectrally Tight Subsets: An Independent problem

- **Problem** Given set X of n points in $\mathbf{R}^d$ and $\alpha \in (0, 1)$, find $S \subseteq X, |S| = \alpha n$ minimizing $||A^S||$, where, $A^S = \{x - \mu(S) : x \in S\}$.
- If instead we had to min $||A^S||_F$, can do 2-approx: Just try each $x \in S$ as a center.

Spectrally Tight Subsets: An Independent problem

- **Problem** Given set X of n points in $\mathbf{R}^d$ and $\alpha \in (0,1)$, find $S \subseteq X, |S| = \alpha n$ minimizing $||A^S||$, where, $A^S = \{x - \mu(S) : x \in S\}$.
- If instead we had to min $||A^S||_F$, can do 2-approx: Just try each $x \in S$ as a center.
- Easy to see that taking a data point as center doesn't give a good approx for spectral norm even for GMM's.

Spectrally Tight Subsets: An Independent problem

- **Problem** Given set X of n points in $\mathbf{R}^d$ and $\alpha \in (0,1)$, find $S \subseteq X, |S| = \alpha n$ minimizing $||A^S||$, where, $A^S = \{x - \mu(S) : x \in S\}$.
- If instead we had to min $||A^S||_F$, can do 2-approx: Just try each $x \in S$ as a center.
- Easy to see that taking a data point as center doesn't give a good approx for spectral norm even for GMM's.
- Aside: Role of Spectral Norm instead of Frobenius norm in Clustering not studied enough.

Expanders and Spectrally tight sets

1. $S = \{a_1, a_2, \ldots a_s\}$ set of points. $\sigma(S) = ||\{a_i - \mu(S) : i = 1, 2, \ldots s\}||/\sqrt{s}$.

Expanders and Spectrally tight sets

1. $S = \{a_1, a_2, \ldots a_s\}$ set of points. $\sigma(S) = ||\{a_i - \mu(S) : i = 1, 2, \ldots s\}|| / \sqrt{s}$.
2. $\sigma(S)^2 = \frac{1}{s^2} ||\{a_i - a_j : i, j = 1, \ldots, s\}||^2 = \frac{1}{s^2} \text{Max}_{|u|=1} \sum_{i,j} (u \cdot (a_i - a_j))^2 \ldots (*)$

Expanders and Spectrally tight sets

1. $S = \{a_1, a_2, \dots a_s\}$ set of points. $\sigma(S) = ||\{a_i - \mu(S) : i = 1, 2, \dots s\}||/\sqrt{s}$.
2. $\sigma(S)^2 = \frac{1}{s^2}||\{a_i - a_j : i, j = 1, \dots, s\}||^2 = \frac{1}{s^2}\text{Max}_{|u|=1}\sum_{i,j}(u\cdot(a_i - a_j))^2 ...(*)$
3. Suppose $H(V, E)$ is an $\Omega(1)$- expander graph with $|V| = s$ and $i \in V \leftrightarrow a_i$.

Expanders and Spectrally tight sets

1. $S = \{a_1, a_2, \ldots a_s\}$ set of points. $\sigma(S) = ||\{a_i - \mu(S) : i = 1, 2, \ldots s\}||/\sqrt{s}$.
2. $\sigma(S)^2 = \frac{1}{s^2}||\{a_i - a_j : i, j = 1, \ldots, s\}||^2 = \frac{1}{s^2}\text{Max}_{|u|=1}\sum_{i,j}(u\cdot(a_i - a_j))^2 \ldots (*)$
3. Suppose $H(V, E)$ is an $\Omega(1)$- expander graph with $|V| = s$ and $i \in V \leftrightarrow a_i$.
4. Cheeger $\implies$ for every unit vector u, an upper bound on (*) in terms of the sum over only the edges of H (instead of all (i, j).)

Expanders and Spectrally tight sets

1. $S = \{a_1, a_2, \ldots a_s\}$ set of points. $\sigma(S) = ||\{a_i - \mu(S) : i = 1, 2, \ldots s\}||/\sqrt{s}$.
2. $\sigma(S)^2 = \frac{1}{s^2}||\{a_i - a_j : i, j = 1, \ldots, s\}||^2 = \frac{1}{s^2}\text{Max}_{|u|=1}\sum_{i,j}(u \cdot (a_i - a_j))^2 \ldots (*)$
3. Suppose $H(V, E)$ is an $\Omega(1)$- expander graph with $|V| = s$ and $i \in V \leftrightarrow a_i$.
4. Cheeger $\implies$ for every unit vector u, an upper bound on (*) in terms of the sum over only the edges of H (instead of all (i, j).)
5. So, a spectrally tight subset S can be found by just constructing an expander graph with sum (over edges) of (*) bounded well.

Expanders and Spectrally tight sets

1. $S = \{a_1, a_2, \ldots a_s\}$ set of points. $\sigma(S) = ||\{a_i - \mu(S) : i = 1, 2, \ldots s\}||/\sqrt{s}$.
2. $\sigma(S)^2 = \frac{1}{s^2}||\{a_i - a_j : i, j = 1, \ldots, s\}||^2 = \frac{1}{s^2}\text{Max}_{|u|=1}\sum_{i,j}(u\cdot(a_i - a_j))^2 \ldots (*)$
3. Suppose $H(V, E)$ is an $\Omega(1)$- expander graph with $|V| = s$ and $i \in V \leftrightarrow a_i$.
4. Cheeger $\implies$ for every unit vector u, an upper bound on (*) in terms of the sum over only the edges of H (instead of all (i, j).)
5. So, a spectrally tight subset S can be found by just constructing an expander graph with sum (over edges) of (*) bounded well.
6. Another line of work helps: Leighton, Rao; Bernstein, Brand,..: Every dense graph contains a large expander. (Studied under "Expander Decomposition", used in Clustering :Kannan, Vempala, Vetta)

Expanders and Spectrally tight sets

1. $S = \{a_1, a_2, \ldots a_s\}$ set of points. $\sigma(S) = ||\{a_i - \mu(S) : i = 1, 2, \ldots s\}|| / \sqrt{s}$.
2. $\sigma(S)^2 = \frac{1}{s^2} ||\{a_i - a_j : i, j = 1, \ldots, s\}||^2 = \frac{1}{s^2} \text{Max}_{|u|=1} \sum_{i,j} (u \cdot (a_i - a_j))^2 \ldots (*)$
3. Suppose $H(V, E)$ is an $\Omega(1)$- expander graph with $|V| = s$ and $i \in V \leftrightarrow a_i$.
4. Cheeger $\implies$ for every unit vector u, an upper bound on (*) in terms of the sum over only the edges of H (instead of all (i, j).)
5. So, a spectrally tight subset S can be found by just constructing an expander graph with sum (over edges) of (*) bounded well.
6. Another line of work helps: Leighton, Rao; Bernstein, Brand,..: Every dense graph contains a large expander. (Studied under "Expander Decomposition", used in Clustering :Kannan, Vempala, Vetta)
7. Can substitute "expander" by "dense" in (5)

Dense graphs Via Semi-Definite Program

v_i vector label on data point a_i. $u_{ij} = |(a_i - a_j) \cdot u|$. Solve SDP below.
$\{(i,j) : v_i \cdot v_j \geq \Omega(1)\} \in \Omega(n^2) \rightarrow$ dense.
(4) says (*) summed over edges small.

$$\min. \quad \sigma^2$$

$$\sum_{i=1}^{n} v_i \cdot v_i = \alpha n \tag{1}$$

$$\sum_{i,j} v_i \cdot v_j \geq \alpha^2 n^2 \tag{2}$$

$$|v_i| \leq 1 \quad \forall i \in [n]. \tag{3}$$

$$\sum_{i,j} u_{ij}^2 v_i \cdot v_j \leq 2\alpha^2 \sigma^2 n^2 \qquad \forall \quad \text{unit directions } u. \tag{4}$$

$$v_i \cdot v_j \geq 0 \qquad \forall i, j \tag{5}$$

Getting by with just data, w_0 not given

- Try $w = 1, 1 - (1/n), 1 - (2/n), ...$ When we hit the correct w_0, the correct k would be found by above. What can go wrong before that ? Recall our alg finds nuclii of clusters.

Getting by with just data, w_0 not given

- Try $w = 1, 1-(1/n), 1-(2/n), ...$ When we hit the correct w_0, the correct k would be found by above. What can go wrong before that ? Recall our alg finds nuclii of clusters.
- Three possible failures:

Getting by with just data, w_0 not given

- Try $w = 1, 1-(1/n), 1-(2/n), ...$ When we hit the correct w_0, the correct k would be found by above. What can go wrong before that ? Recall our alg finds nuclii of clusters.
- Three possible failures:
  1. Some nucleus has significant parts from two different C_ℓ.

Getting by with just data, w_0 not given

- Try $w = 1, 1-(1/n), 1-(2/n), ...$ When we hit the correct w_0, the correct k would be found by above. What can go wrong before that ? Recall our alg finds nuclii of clusters.
- Three possible failures:
  1. Some nucleus has significant parts from two different C_ℓ.
  2. Two different nuclii may have large parts of a single C_ℓ.

Getting by with just data, w_0 not given

- Try $w = 1, 1-(1/n), 1-(2/n), \ldots\ldots$ When we hit the correct w_0, the correct k would be found by above. What can go wrong before that ? Recall our alg finds nuclii of clusters.
- Three possible failures:
  1. Some nucleus has significant parts from two different C_ℓ.
  2. Two different nuclii may have large parts of a single C_ℓ.
  3. One of the C_ℓ may have been left out of all nuclii.

Getting by with just data, w_0 not given

- Try $w = 1, 1-(1/n), 1-(2/n), ...$ When we hit the correct w_0, the correct k would be found by above. What can go wrong before that ? Recall our alg finds nuclii of clusters.
- Three possible failures:
  1. Some nucleus has significant parts from two different C_ℓ.
  2. Two different nuclii may have large parts of a single C_ℓ.
  3. One of the C_ℓ may have been left out of all nuclii.
- Intuitively: $\neg$ (1) $\Longrightarrow$ Each Nuleus is from exactly one C_ℓ. $\neg(2) \wedge \neg(3) \Longrightarrow$ each C_ℓ in a unique nuclues. QED.

Getting by with just data, w_0 not given

- Try $w = 1, 1-(1/n), 1-(2/n), ...$ When we hit the correct w_0, the correct k would be found by above. What can go wrong before that ? Recall our alg finds nuclii of clusters.
- Three possible failures:
  1. Some nucleus has significant parts from two different C_ℓ.
  2. Two different nuclii may have large parts of a single C_ℓ.
  3. One of the C_ℓ may have been left out of all nuclii.
- Intuitively: $\neg$ (1) $\implies$ Each Nuleus is from exactly one C_ℓ. $\neg(2) \wedge \neg(3) \implies$ each C_ℓ in a unique nuclues. QED.
- Failure 1 $\implies$ NLSC violted. Failure 2 $\implies$ the two nuclii have μ too close. Failure 3 $\implies$???? NOT enough to deal only with nuclii... Complexity...

Open Questions

- In practice: High running time SDP based solution. Faster Algorithms for k?

Open Questions

- In practice: High running time SDP based solution. Faster Algorithms for k?
- Other applications of Spectrally tight subsets.

Open Questions

- In practice: High running time SDP based solution. Faster Algorithms for k?
- Other applications of Spectrally tight subsets.
- Analog of $k-$means: Find the partition into k subsets which minimizes weighted sum of spectral norms of the sets. Algorithm ? Hueristic ? For $k-$means, it is not true in general that for data from a mixture of k Mean-Separated spherical Gaussians, approximate optimal $k-$ means partition finds the correct clustering. Prove it is true for spectral norm based measure in some generality. A. Sinop