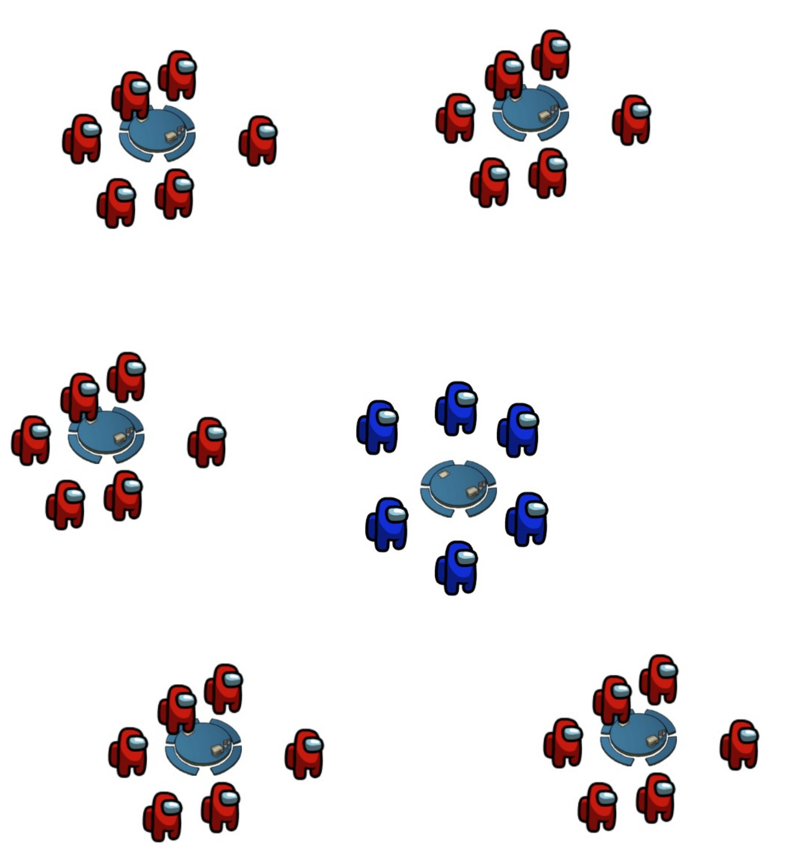
Recent Advances in List-Decodable Mean Estimation

Presenter: Kevin Tian
Stanford University

Joint work with:
Ilias Diakonikolas, Daniel Kongsgaard, Daniel Kane, Jerry Li

List-decodable mean estimation: the problem



Input: Dataset $\{X_i\}$ of size n: α fraction i.i.d. from bounded covariance D, others arbitrary

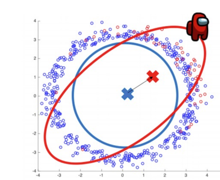
Output: hypothesis list L of size $O(k) := O(\frac{1}{\alpha})$ containing at least one "good" mean estimate

Lower bound: list size $O(k)$, $O(dk)$ samples, and estimation error $O(\alpha^{-0.5})$ necessary

Flexible, robust statistical model!
- Robust analog of learning mixture models
- Apps: crowd-sourcing, community detection
- Semi-verified learning: "a data prism"

Previous approaches

"Filtering": [DKKLMS17, Ste18]



Intuition: substantial mean deviation induces covariance deviation!

"Spectral signatures scoring": remove at least one outlier in nearly-linear time.

"Simple" algorithm: $O(nd^2)$ time

"Large α regime"

"Fast filtering": [DHL19]

$\mathbf{Y}_{t+1} \propto \exp(\log \mathbf{Y}_t + \eta \mathbf{G}_t)$

Goal: bound covariance operator norm. → Regret against spectrahedron.

"Quantum entropy scoring": remove against multiple directions at once.

"Fast" algorithm: $O(nd)$ time

*Bounded second moments only. Stronger assumptions → improved rates.

A murkier landscape…

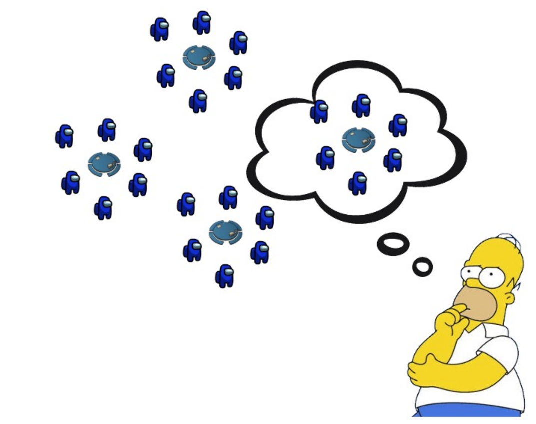
Work	Approach	Runtime	Comments
[CSV17]	Iterative clustering + convex program	$\mathrm{poly}(d)$	Introduced list-decodable problem
[DKS18]	Polynomial optimization	$\mathrm{poly}(d)$	Gaussian only, gets better error rates
[CMY20]	Ky Fan positive SDP solver	$\widetilde{O}(ndk^6)$	Optimal error rate, list size, samples
[DKK20]	Fast 1-d multifilter + pruning tree	$\widetilde{O}(n^2dk^2)$	One-dimensional filtering (speedup?)
[VW02]	k-PCA	$O(ndk)$	Well-separated Gaussian mixtures

Question 1: can we design simpler algorithms "decoupling" filtering and clustering?

"Small α regime"

Question 2: is $O(ndk)$ (PCA time) really necessary in the runtime, even non-robustly?

Application: mixture models



List-decoding guarantees imply returning estimates of all cluster means.

Clustering algorithm sketch:
1. Map every datapoint to closest hypothesis.
2. Define equivalence relation by mapping distances.
3. Prune unused hypotheses.

Extends cleanly to contaminated mixtures, heavier tails! "More efficient" equivalence relations also degree of freedom.

Our results: clustering mixture models

Goal: label 1-o(1) points correctly from a *mixture model*, assuming components "nice" and "well-separated"

$$\mathcal{M} = (1-\epsilon)\sum_{i\in[k]} \alpha_i \mathcal{D}_i + \epsilon \mathcal{D}_{\mathrm{adv}} \qquad \alpha_i \geq \alpha \; \forall i \in [k], \quad \epsilon = O(\alpha)$$

Assumption	Separation	Runtime
sub-Gaussian	$\widetilde{\Omega}\left(\alpha^{-\frac{1}{2}}\right)$	$n^{1+o(1)}d$
bounded 4th moments	$\widetilde{\Omega}\left(\alpha^{-\frac{1}{2}}\right)$	$n^{1+o(1)}d$
bounded 2nd moments	$\widetilde{\Omega}\left(\alpha^{-\frac{5}{6}}\right)$	$n^{1+o(1)}d$

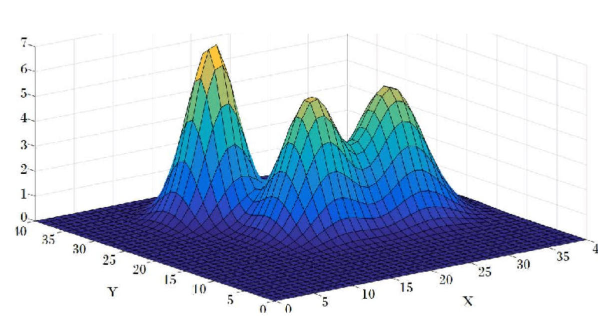
First runtime improvement to clustering GMMs since [VW02]. Matches state-of-the-art GMM separation under a covariance bound by [AM05] + robustness, heavy tails.

Open question: [VW02] used stronger concentration properties to obtain better separation for *spherical GMMs*. Can we do better for clustering sub-Gaussian mixture models?

Our approaches: simpler and faster list-decodable mean estimation

I. Robustly matching the "PCA barrier": [DKKLT20]

Decoupling from subspace identification.



Intuition: in mixture model case, k-PCA captures span of means

A new approach: "decoupling" clustering and filtering

Suffices to learn mean in k dimensions → naïve sampling!

Identifiability proof: small covariance → small mean error.

Potential function: $O(k)$-th largest eigenvalue. Identifiability proof + naïve sampling gets us the rest of the way!

SIFT: A new filtering algorithm for subspace identification.

Algorithm 1 SIFT(T,δ)

1: **Input:** $T \subset \mathbb{R}^d$ with $|T| = n$ satisfying Assumption 1, $\delta \in (0,1)$
2: $w^{(0)} \leftarrow \frac{1}{n}\mathbb{1}_T$, $t \leftarrow 0$, $\beta \leftarrow 1$, $k \leftarrow \lceil \frac{4}{\alpha} \rceil$
3: $\mathbf{V} \leftarrow \mathsf{Power}(\mathrm{Cov}_{w^{(t)}}(T), k, 0.2, \frac{\delta}{2n})$
4: $\mathbf{\Sigma} \leftarrow \mathbf{V}^\top \mathrm{Cov}_{w^{(t)}}(T)\mathbf{V}$
5: **while** $\lambda_k(\mathbf{\Sigma}) \geq \frac{4}{\sqrt{\beta}}$ **do**
6: $\quad \tau_i^{(t)} \leftarrow \left\| \mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{V}^\top (X_i - \mu_w(T)) \right\|_2^2$ for all $i \in T$
7: $\quad w_i^{(t+1)} \leftarrow \left(1 - \frac{\tau_i^{(t)}}{\tau_{\max}^{(t)}}\right) w_i^{(t)}$ for all $i \in T$, where $\tau_{\max}^{(t)} := \max_{i \in T \mid w_i^{(t)} \neq 0} \tau_i^{(t)}$
8: $\quad t \leftarrow t+1$, $\beta \leftarrow \|w^{(t)}\|_1$
9: $\quad \mathbf{V} \leftarrow \mathsf{Power}(\mathrm{Cov}_{w^{(t)}}(T), k, 0.2, \frac{\delta}{2n})$
10: $\quad \mathbf{\Sigma} \leftarrow \mathbf{V}^\top \mathrm{Cov}_{w^{(t)}}(T)\mathbf{V}$
11: **end while**
12: **return** $L := \{\mathbf{V}\mathbf{V}^\top X_i + (\mathbf{I} - \mathbf{V}\mathbf{V}^\top)\mu_{w^{(t)}}(T)$ where $i \in T$ is sampled uniformly at random$\}$, with list size $|L| = \lceil \frac{2}{\alpha}\log\frac{2}{\delta} \rceil$

Birds' eye view:

Line 5: termination condition (bounded $O(k)$-th eigenvalue)

Line 6: "whitened" scores to prevent symmetry-breaking

Lines 7-8: standard "soft filtering" procedure

Line 12: random sampling within learned subspace

Under the hood: *saturation*, weaker "relative" criterion for list-decodable filtering.

Practical direction: robust PCA?

Fast filtering via Ky Fan SDP regret minimization.

Idea: regret minimization against *k-Fantope*

$$\{\mathbf{Y} \mid \mathbf{Y} \succeq \mathbf{0}, \|\mathbf{Y}\|_{\mathrm{op}} \leq 1, \mathrm{Tr}(\mathbf{Y}) \leq k\}$$

New technical tool: fine-grained *Ky Fan MMW*

$$\left\| \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{G}_t \right\|_k \leq \frac{2}{T}\sum_{t=0}^{T-1} \langle \mathbf{G}_t, \mathbf{Y}_t \rangle + \frac{k \log d}{\eta T}.$$

Filter against covariance matrices to halve Ky Fan norm?

Pitfalls from [DHL19]!!!

Relative notions of saturation ☠️

Whitening breaks monotonicity ☠️

Fast, robust filtering for MMW ☠️

"Gospel of DJ Khaled"

Every log iterations, either:
1. Halve the weight.
2. Peel off k dimensions.
3. Decrease Ky Fan norm.

Main result:
Optimal error + list size in $O(ndk + k^6)$.
Proof: bounded Ky Fan + run SIFT on the low-dimensional learned subspace!

II. Robustly breaking the "PCA barrier": [DKKLT21]

A new, "one-shot" MMW potential alternative.

$$\left\| \frac{1}{T}\sum_{0\leq t<T} \mathbf{G}_t \right\|_{\mathrm{op}} \leq \frac{2}{T}\sum_{0 \leq t < T}\langle \mathbf{G}_t, \mathbf{Y}_t\rangle + \frac{\log d}{\eta T}$$

MMW-based scoring

requires monotonicity

requires "burn-in" to make progress

Previous MMW framework

$$\Phi_t := \mathbf{Tr}\left(\mathbf{G}_t^{2\log d}\right)$$

$$\Phi_{t+1} \leq \frac{1}{2}\mathrm{Tr}\left(\mathbf{G}_{t+1}^{2\log d + 1}\right) + d(2)^{\log d}$$
$$\leq \frac{1}{2}\langle \mathbf{G}_t^{2\log d}, \mathbf{G}_{t+1}\rangle + d(2)^{\log d}$$

scoring against old potential

One-shot potential decrease

Goal: decrease scores "against"

$$\mathbf{M}_t := \mathbf{G}_t^{2\log d}$$

JL: suffices to decrease univariate scores along log random directions.

JL as a certificate

Rethinking the [DKK20] multifilter.

Old multifilter tree

$S \subseteq [n]$

bounded-variance majority → "Type I": $S' \subset S$ — Spectral outlier filtering

large variance → "Type II": $S_1, S_2 \subset S$ — Threshold checks

New goal: Partition into bounded-variance subsets (clustering).

New multifilter tree (one layer)

$S \subseteq [n]$

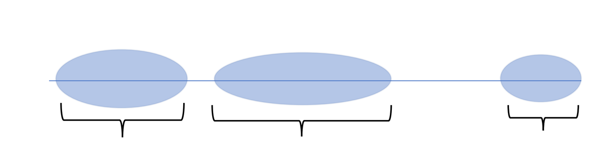
bounded-variance majority → "Cluster": $S' \subset S$ — Randomized dropout

large variance → "Split": $S_1, S_2 \subset S$ — Bucketed threshold checks

Putting it all together.

$$S \subseteq [n] \xrightarrow{\text{Random 1-d clustering}} \{S_i^{(1)}\} \xrightarrow{\text{Random 1-d clustering}} \cdots \rightarrow \{S_i^{(K)}\}, \qquad K = O(\log d)$$

Fast one-dimensional partitioning in random direction



Bounded covariance clusters

Recursive "split or cluster"

Given set, cases based on "majority" variance.
I. Large: identify a "split" (recurse)
II. Small: rapid dropout to fix variance (exit)

Invariants:
- Saturation
- Bounded set sizes
- Bounded-covariance clusters