Robust Mixture Learning when Outliers Overwhelm Small Groups

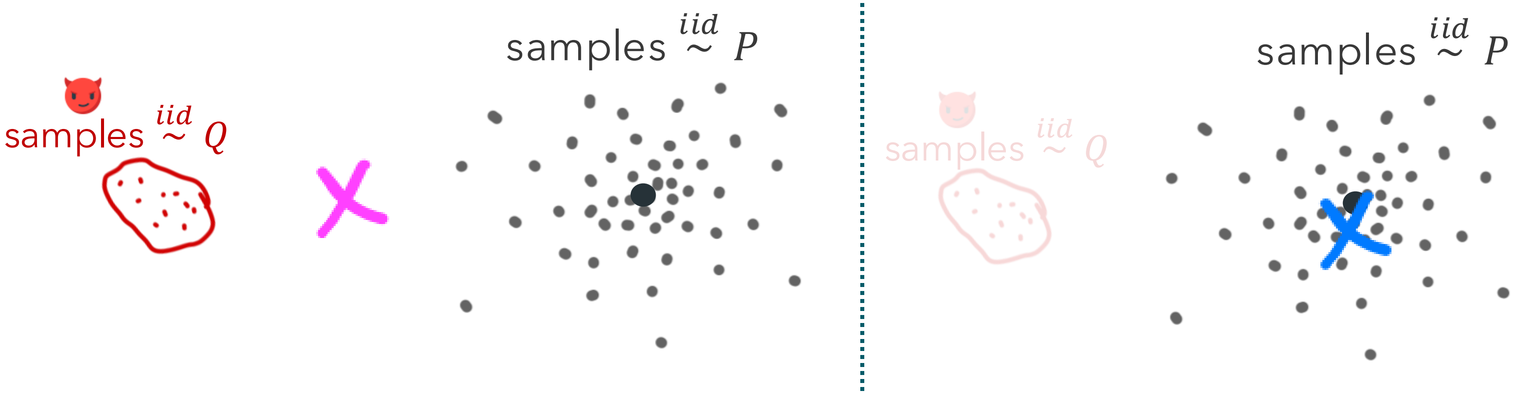
Fanny Yang

Statistical Machine Learning group,
Department of Computer Science, ETH Zurich

joint work with:

Objective I: Robustness *against outliers*

Example: high-dimensional robust **mean estimation** under additive contamination



non-robust estimate

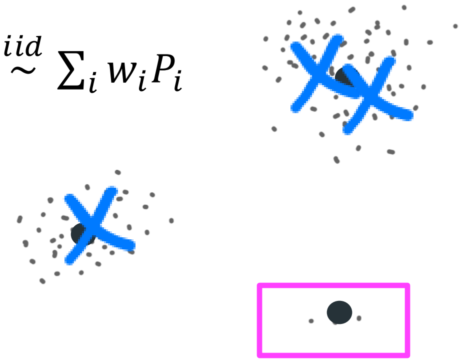
robust estimate (by filtering outliers)

*classical robust statistics problem, pioneered '60s by Anscombe, Huber, Tukey etc.

Objective II: Good small group performance

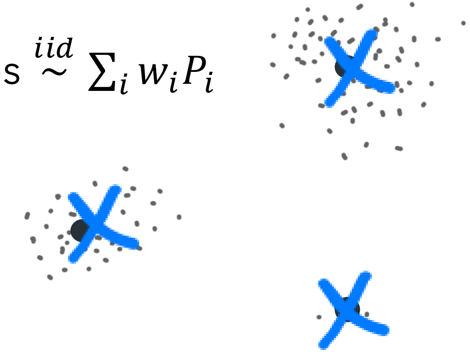
Example: preserving **minority group** representation ("group fairness" in this talk)

samples $\overset{iid}{\sim} \sum_i w_i P_i$



list that ignores minorities

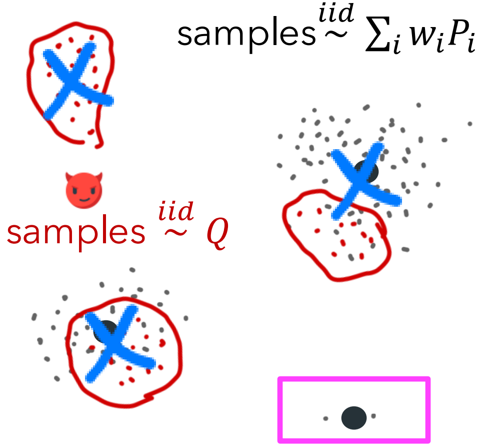
samples $\overset{iid}{\sim} \sum_i w_i P_i$



list that represents minorities

How about robustness **and** "group fairness"?

Joint problem: Minority group preservation under large additive contamination



robust for some i, but "unfair":
ignores some subpopulation

Contradicting objectives:

To obtain robustness

filter out
points that look different

To preserve minority groups:

keep
points that look different

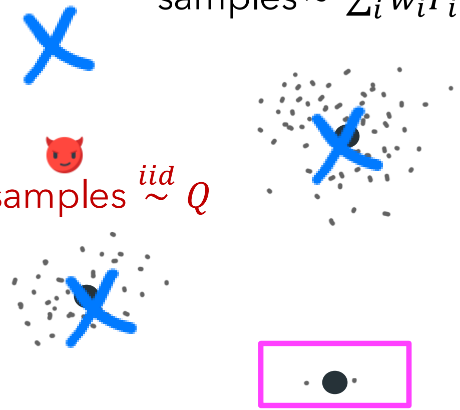
includes both outlier and minority groups
- indistinguishable to algorithm!

How about robustness **and** "group fairness"?

Joint problem: Minority group preservation under large additive contamination

samples $\overset{iid}{\sim} \sum_i w_i P_i$

samples $\overset{iid}{\sim} Q$



robust for some i, but "unfair": ignores some subpopulation

Contradicting objectives:

To obtain robustness

filter out

points that look different

To preserve minority groups:

keep

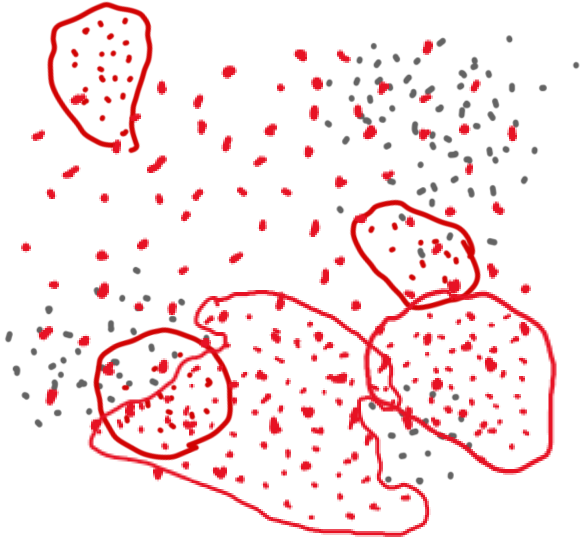
points that look different

How to formally quantify this "trade-off"?

__**Goal**__**:** *Quantify the cost* of preserving **small groups under adversarial corruptions**

Setup: Mixture learning with adversarial corruptions

$k = 3$



k: #components

$w_i \approx$ frequency of group i in dataset

$\varepsilon \in [0,1]$: corruption proportion

Sampled distribution $\mathcal{P}_X = \underbrace{\sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I)}_{\text{true signal}} + \underbrace{\varepsilon Q}_{\text{adv. corruption}}$

Q: arbitrary, adversarial

Goal: Given $X_1, \ldots, X_n \overset{iid}{\sim} \mathcal{P}_X$, recover all μ_i with $w_i > w_{low}$

with $\varepsilon > w_{low}$ (w.l.o.g. consider $w_{low} = \min_i w_i$)

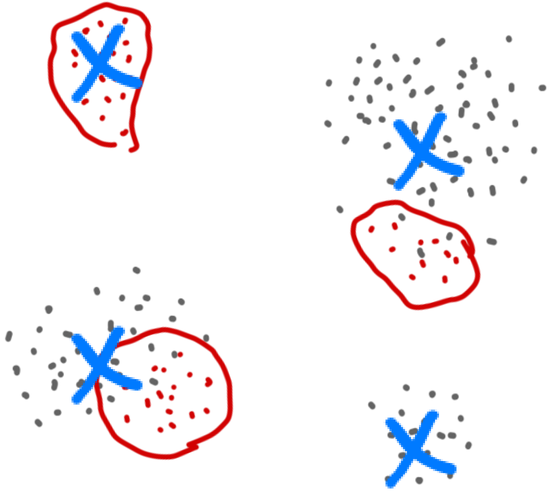
Q1. How to recover all means under corruptions?

Q2. What's the cost of choosing small $\mathrm{w}_{\mathrm{low}} < \varepsilon$?

*Gaussian inliner components for simplicity of presentation, paper considers distributions with subgaussian moments

Q1: How to recover all means under corruptions?

$k = 3$



Problem:

For large $\varepsilon > w_{low}$, outliers indistinguishable from true minority

→ filtering and outputting *a list of fixed size* k won't work!

List-decodable paradigm comes to the rescue:

Idea: Output *more candidates* of means than # true means!

(originated from error-correcting codes Elias '1957)

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$$

Answer: List-decodable mixture learning

$k = 3$

List-decodable mixture learning goal:

Given $w_{low} < \varepsilon$, data points from $\mathcal{P}_X$

output list $L = \{\hat{\mu}_1, \hat{\mu}_2 \dots\}$ with $|L| > k$ such that:

- for any component i with $w_i > w_{low}$, there is

  an element $\hat{\mu}$ in list L with small estimation error $||\mu_i - \hat{\mu}||_2$

- list size is small / has small list-size overhead $|L| - k$

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$$

Q2: What's the cost of small w_{low}

for poly-time algorithms

Goal: *Quantify cost of* preserving **small groups under adversarial corruptions**

Goal: *Quantify* how much ***estimation error*** *on large groups and* ***list size*** increase with w_{low}

List-decodable mixture learning goal:

Given $w_{low} < \varepsilon$, data points from $\mathcal{P}_X$

output list $L = \{\hat{\mu}_1, \hat{\mu}_2 \ldots\}$ with $|L| > k$ such that:

- for any component i with $w_i > w_{low}$, there is an element $\hat{\mu}$ in list L with small estimation error $||\mu_i - \hat{\mu}||_2$
- list size is small / has small list-size overhead $|L| - k$

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$$

Q2: What's the cost of small w_{low}

for poly-time algorithms

Goal: *Quantify cost of* preserving **small groups under adversarial corruptions**

⬇

Goal: *Quantify* how much ***estimation error*** *on large groups and* ***list size*** increase with w_{low}

What can be achieved

by a poly-time algorithm

and how "optimal" is it?

Naïve approach: list-decodable mean estimation (LD-ME)

For **each component j**, can view as mean estimation problem with large corruption proportion

rest of the mixture can be (conservatively) viewed as adversarial

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q \quad \Longrightarrow \quad \mathcal{P}_X = \underbrace{w_j \mathcal{N}(\mu_j, I)}_{\text{true signal}} + \underbrace{(1 - w_j)\tilde{Q}}_{\text{effective large corruption}}$$

Q: arbitrary, 😈 adversarial

making problem "harder"

$\tilde{Q}$: arbitrary, 😈 adversarial

$$\text{"Recovering" } \mu_j \text{ in this setting} \quad \xLeftarrow{\text{implies}} \quad \underbrace{\text{"Recovery" of } \mu_j \text{ in this setting}}_{\text{Solved by list-decodable mean estimation algorithms}}$$

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$$

Caveats of existing poly-time algorithms

Algorithm outputs	Prior work using LD-ME	Our algorithm (arbitrary sep.)	Our algorithm (well-separated)	Lower bounds* (well-separated)
List of size	$O(1/w_{\mathrm{low}})$	$O(1/w_{\mathrm{low}})$	$k + O(\varepsilon/w_{\mathrm{low}})$	$k + \lfloor \varepsilon/w_{\mathrm{low}} \rfloor$
Estimation error $\lVert\hat{\mu} - \mu_i\rVert_2$ *w.h.p.* for $w_i > w_{low}$	$O(\sqrt{\log \frac{1}{w_{\mathrm{low}}}})$	$O(\sqrt{\log \frac{1}{w_i}})$	$O(\sqrt{\log \frac{w_i + \varepsilon}{w_i}})$	$\Omega(\sqrt{\log \frac{w_i + \varepsilon}{w_i}})$

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$$

Caveats of existing poly-time algorithms

DKS '18

Algorithm outputs	Prior work using LD-ME	Our algorithm (arbitrary sep.)	Our algorithm (well-separated)	Lower bounds* (well-separated)
List of size	$O(1/w_{\text{low}})$	$O(1/w_{\text{low}})$	$k + O(\varepsilon/w_{\text{low}})$	$k + \lfloor \varepsilon/w_{\text{low}} \rfloor$
Estimation error $\|\|\hat{\mu} - \mu_i\|\|_2$ *w.h.p.* for $w_i > w_{low}$	$O\left(\sqrt{\log \frac{1}{w_{\text{low}}}}\right)$	$O\left(\sqrt{\log \frac{1}{w_i}}\right)$	$O\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$	$\Omega\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$

Big components suffer same large error as small ones!

Challenges in this setting:

1. w_{low} too small as an estimate for weight of big clusters
2. Other inlier clusters <u>unnecessarily</u> treated as outliers by many clusters

*Diakonikolas, Kane, Stewart. "List-decodable robust mean estimation and learning mixtures of spherical Gaussians", '18

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$$

Guarantees for our poly-time algorithm in a nutshell

DBTWNSS**Y** '24

Algorithm outputs	Prior work using LD-ME	Our algorithm (arbitrary sep.)	Our algorithm (well-separated)	Lower bounds* (well-separated)
List of size	$O(1/w_{\text{low}})$	$O(1/w_{\text{low}})$	$k + O(\varepsilon/w_{\text{low}})$	$k + \lfloor \varepsilon/w_{\text{low}} \rfloor$
Estimation error $\|\|\hat{\mu} - \mu_i\|\|_2$ *w.h.p.* for $w_i > w_{low}$	$O\left(\sqrt{\log \frac{1}{w_{\text{low}}}}\right)$	$O\left(\sqrt{\log \frac{1}{w_i}}\right)$	$O\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$	$\Omega\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$

error depends
on cluster size

Challenges in this setting:

1. w_{low} too small as an estimate for weight of big clusters
2. Other inlier clusters <u>unnecessarily</u> treated as outliers by many clusters

$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$

Guarantees for our poly-time algorithm in a nutshell

DBTWNSS**Y** '24

$||\mu_i - \mu_j||_2$ large

Algorithm outputs	Prior work using LD-ME	Our algorithm (arbitrary sep.)	Our algorithm (well-separated)	Lower bounds* (well-separated)
List of size	$O(1/w_{\text{low}})$	$O(1/w_{\text{low}})$	$k + O(\varepsilon/w_{\text{low}})$	$k + \lfloor \varepsilon/w_{\text{low}} \rfloor$
Estimation error $\lvert\lvert\hat{\mu} - \mu_i\rvert\rvert_2$ w.h.p. for $w_i > w_{low}$	$O\left(\sqrt{\log \frac{1}{w_{\text{low}}}}\right)$	$O\left(\sqrt{\log \frac{1}{w_i}}\right)$	$O\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$	$\Omega\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$

much better when $w_i \approx \varepsilon$

"oracle error" as if it's LD-ME excluding other inlier clusters i.e. $\mathcal{P}_X \propto w_j \mathcal{N}(\mu_j, I) + \varepsilon Q$ leveraging separation

Challenges in this setting:

1. w_{low} too small as an estimate for weight of big clusters
2. Other inlier clusters <u>unnecessarily</u> treated as outliers by many clusters

*Dmitriev, Buhai et al. "Robust Mixture Learning when Outliers Overwhelm Small Groups", 2024

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$$

Guarantees for our poly-time algorithm in a nutshell

$||\mu_i - \mu_j||_2$ large

DBTWNSS**Y** '24

Algorithm outputs	Prior work using LD-ME	Our algorithm (arbitrary sep.)	Our algorithm (well-separated)	Lower bounds* (well-separated)
List of size	$O(1/w_{\text{low}})$	$O(1/w_{\text{low}})$	$k + O(\varepsilon/w_{\text{low}})$	$k + \lfloor \varepsilon/w_{\text{low}} \rfloor$
Estimation error $\|\|\hat{\mu} - \mu_i\|\|_2$ *w.h.p.* for $w_i > w_{low}$	$O\left(\sqrt{\log \frac{1}{w_{\text{low}}}}\right)$	$O\left(\sqrt{\log \frac{1}{w_i}}\right)$	$O\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$	$\Omega\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$

Goal: *Quantify how much* ***estimation error*** *on large groups and* ***list size*** *increase with* w_{low}

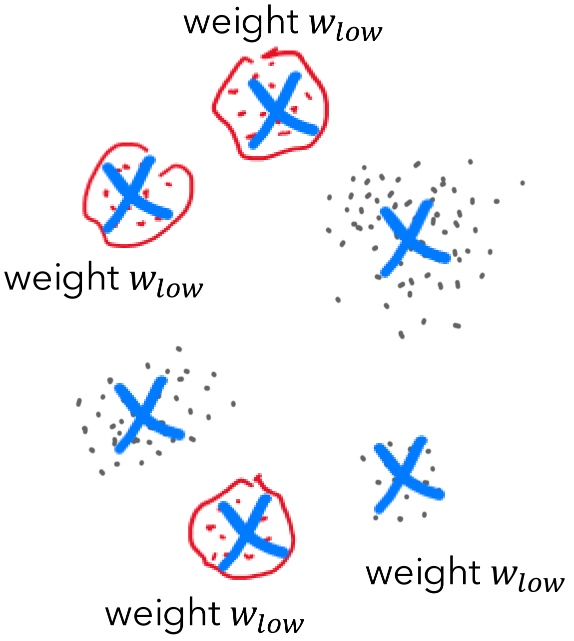
⇨

Estimation error: No cost for large groups to preserve small groups

List size: overhead increase as $1/w_{low}$

*Dmitriev, Buhai et al. "Robust Mixture Learning when Outliers Overwhelm Small Groups", 2024

How optimal? Simple lower bound on the list size



Samples from $\mathcal{P}_X = \sum_{i=1}^{3} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$ with $\epsilon = 3w_{low}$

- Using budget $\epsilon = 3w_{low}$, adversary can place

  3 (outlier/fake) clusters of weight w_{low} with εQ

- We can't tell apart the true vs. fake clusters

$\Rightarrow$ need to output $\frac{\epsilon}{w_{low}} = 3$ more means to capture true+fake

$\Rightarrow$ total list size $|L| > k + \frac{\epsilon}{w_{low}}$

$$\mathcal{P}_X = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, I) + \varepsilon Q$$

Lower bound and interpretation

$\|\mu_i - \mu_j\|_2$ large

	Prior work using LD-ME	Our algorithm (arbitrary sep.)	Our algorithm (well-separated)	Lower bounds (well-separated)
List size	$O(1/w_{\text{low}})$	$O(1/w_{\text{low}})$	$k + O(\varepsilon/w_{\text{low}})$	$k + \lfloor \varepsilon/w_{\text{low}} \rfloor$
Estimation error $\|\|\hat{\mu} - \mu_i\|\|_2$ for $w_i > w_{low}$	$O\left(\sqrt{\log \frac{1}{w_{\text{low}}}}\right)$	$O\left(\sqrt{\log \frac{1}{w_i}}\right)$	$O\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$	$\Omega\left(\sqrt{\log \frac{w_i + \varepsilon}{w_i}}\right)$

Lower bound interpretation for

- List size: smaller list would miss a small true component if all of adversarial proportion ε used to place clusters of size w_{low}
- Estimation error: smaller order for all $w_i > w_{low}$ would require exponential list size

Our meta-algorithm framework for LD-ML

Outer stage: separates dataset into sets $T_1, T_2, \ldots$ exploiting separation (reduces LD-mixture learning to LD-mean estimation)

Inner stage: For each set T_i, run list-codable mean estimation (LD-ME) with unknown weights via a wrapper that uses

base learner as black-box:

given data from adv. corrupted model with known $\epsilon > \frac{1}{2}$, outputs small list

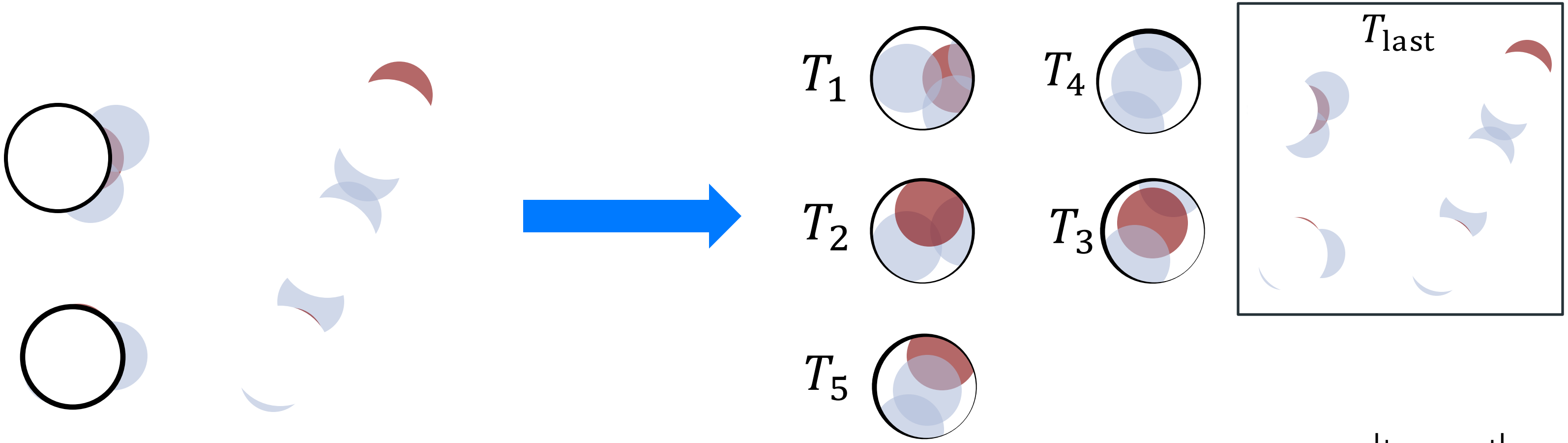
Error guarantees of meta-algorithm inherits error guarantees of base learners!

*for large clusters can improve error further using RME base learner designed for $\epsilon < \frac{1}{2}$

Algorithm: Outer stage (cluster isolation)

Goal: split inlier clusters into separate sets

Approach: finding and isolating regions of high concentration



(for $||\mu_i - \mu_j||_2$ large)

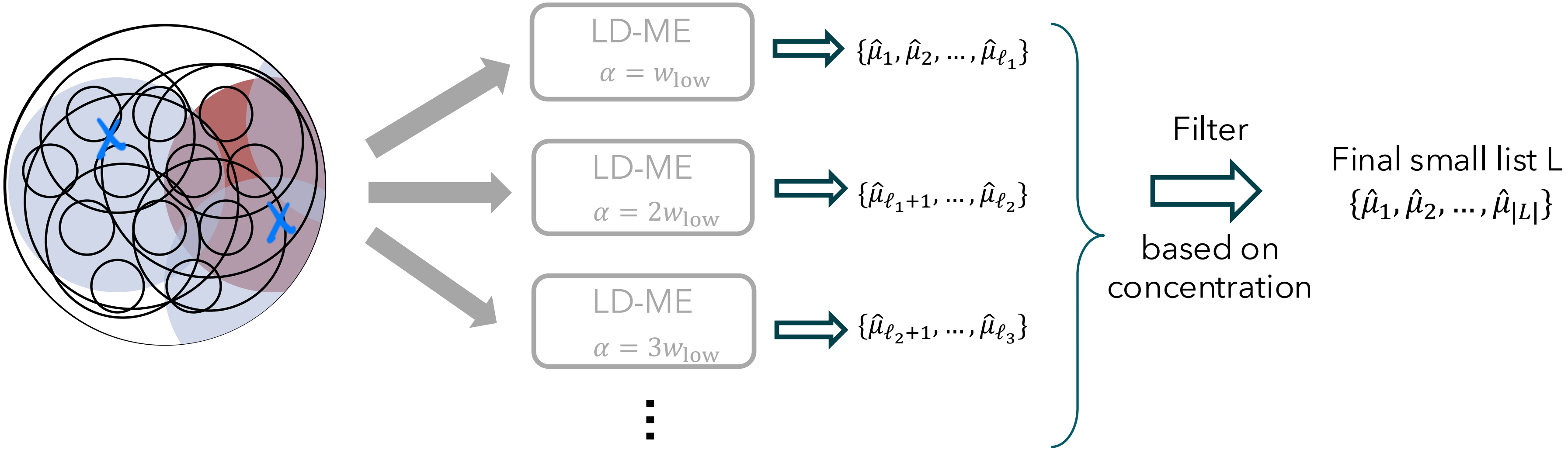
Set of all data points

Sets T_i of points from at most 1 inlier cluster + T_{last} with rest

Algorithm: Inner stage (mean estimation)

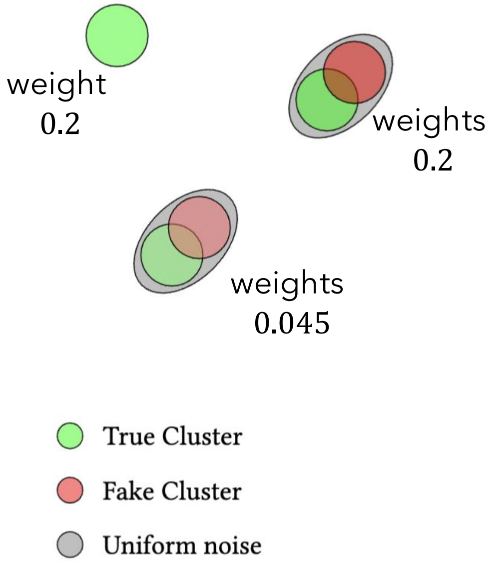
Goal for each T_i: small list & error for the inlier components

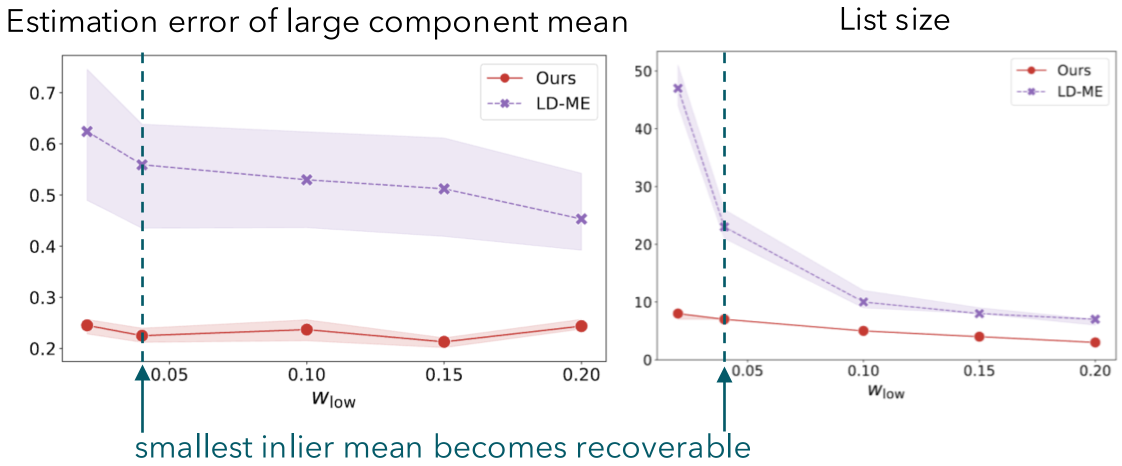
Approach: list-decodable mean-estimation base learners ($k = 1$) with different inlier weight proportion α & filter



Empirical performance compared to LD-ME

Comparison with LD-ME algorithm* (the only baseline with worst-case guarantees)



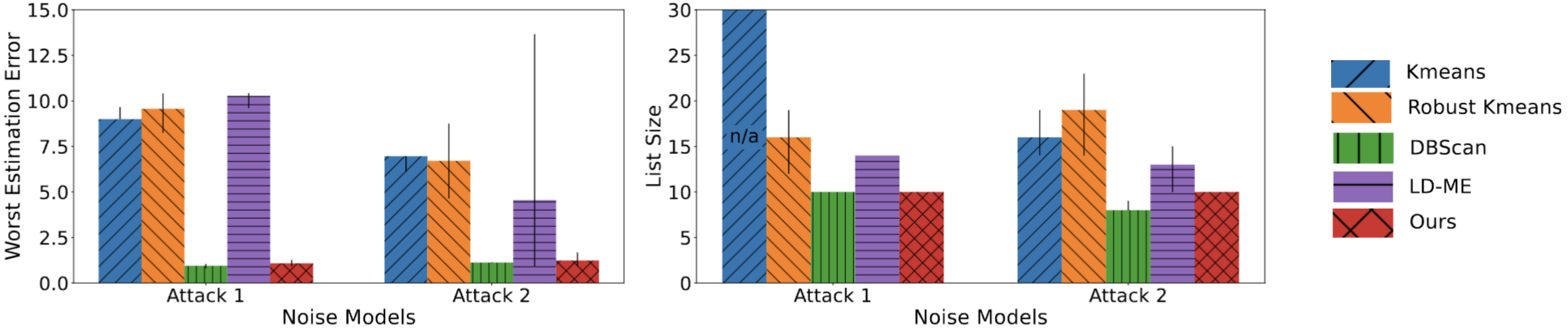


Takeaway: Large clusters estimation and list size do not suffer as much from attending to rare minorities (i.e. choosing small w_{low})

* Diakonikolas, Kane, Kongsgaard, Li, Tian. "Clustering mixture models in almost-linear time via list-decodable mean estimation" 2022

Empirical performance compared to heuristics

- Comparison with popular clustering heuristic with more inlier components
- Attack 1: Adversarial cluster; Attack 2: adversarial points connecting two inliers



Takeaways:
- For same list size, we achieve smaller small-group error
- Need smaller list to achieve same small-group error

Summary

- Studied cost of recovering small groups in the presence of large outlier proportion
  - no cost in estimation error (of larger groups)
  - list size grows with "group size" you care about as $1/w_{low}$
- Achieved by concrete poly-time algorithm for list-decodable mixture learning

  where we can plug in black-box LD-ME and RME
  - guarantees for LD-ML inherit guarantees of black box learners
  - there is optimal algorithm for Gaussian mixture
- In some preliminary experiments empirically matches heuristics like DBSCAN

SML group at ETH Zurich:
sml.inf.ethz.ch

D. Dmitriev, R. Buhai, S. Tiegel, A. Wolters, G. Novikov, A. Sanyal, D. Steurer, F. Yang.

"*Robust Mixture Learning when Outliers Overwhelm Small Groups*", NeurIPS 2024