Learning from Untrusted Data

Moses Charikar Jacob Steinhardt Gregory Valiant

Stanford University

{*moses,jsteinhardt,valiant*}*@cs.stanford.edu*

December 9, 2016

Setting (High-level)

Observe n data points.

αn are "real" (e.g. drawn from p^*), no assumptions on remaining $(1-\alpha)n$.

- Could be noisy, biased, outliers, malicious, etc.

Want to find model that fits the real data.

Motivations

Security of ML systems

- Online communities: users mis-use the like/report/etc. button
- Crowd-sourcing: lazy/fraudulent workers
- Self-driving cars: hackers could purchase car, control training data

Motivations

Security of ML systems

- Online communities: users mis-use the like/report/etc. button
- Crowd-sourcing: lazy/fraudulent workers
- Self-driving cars: hackers could purchase car, control training data

Data monotonicity

- If train $\neq$ test, more data can hurt
- Want to be free to use all the data we have

Motivations

Security of ML systems

- Online communities: users mis-use the like/report/etc. button
- Crowd-sourcing: lazy/fraudulent workers
- Self-driving cars: hackers could purchase car, control training data

Data monotonicity

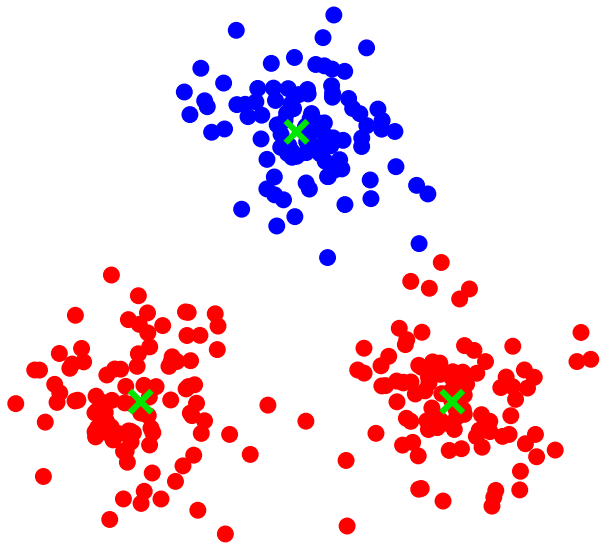
- If train $\neq$ test, more data can hurt
- Want to be free to use all the data we have

Don't want to assume $\alpha \approx 0.99$; we handle $\alpha \ll 1$ and even $\alpha < \frac{1}{2}$!

A majority of bad data

What happens if $\alpha < \frac{1}{2}$?

E.g. for $\alpha = \frac{1}{3}$:



No way to distinguish true solution from duplicates!

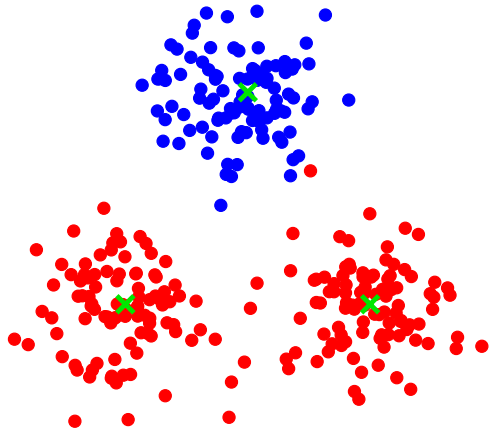
- but can narrow answer to 3 possibilities...

List-decodable learning

To handle $\alpha \leq \frac{1}{2}$, allow learner to output multiple answers:

Definition (List-decodable learning)

The learner is allowed to output m candidate answers, at least one of which must be accurate.



Typically $m = \mathcal{O}\left(\frac{1}{\alpha}\right)$.

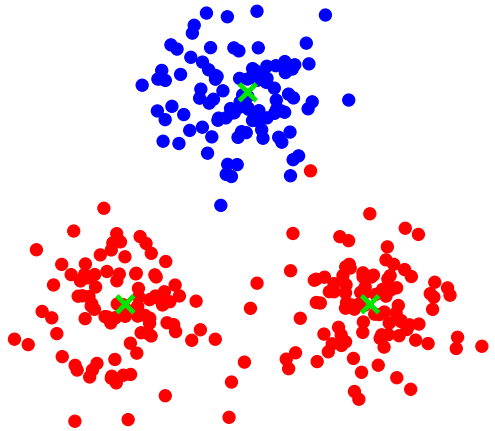
- For our results, can get $\left\lfloor \frac{1.01}{\alpha} \right\rfloor$.

List-decodable learning

To handle $\alpha \leq \frac{1}{2}$, allow learner to output multiple answers:

Definition (List-decodable learning)

The learner is allowed to output m candidate answers, at least one of which must be accurate.



Typically $m = \mathcal{O}\left(\frac{1}{\alpha}\right)$.

- For our results, can get $\left\lfloor \frac{1.01}{\alpha} \right\rfloor$.

Alternative: *semi-verified model*.

What can the adversary do?

Intuition: adversary can perturb cluster. . .

What can the adversary do?

Intuition: adversary can perturb cluster. . .

What can the adversary do?

Intuition: adversary can perturb cluster. . .

What can the adversary do?

Intuition: adversary can perturb cluster. . .

What can the adversary do?

Intuition: adversary can perturb cluster. . .

. . . but large perturbations detected as outliers.

What can the adversary do?

Intuition: adversary can perturb cluster. . .

. . . but large perturbations detected as outliers.

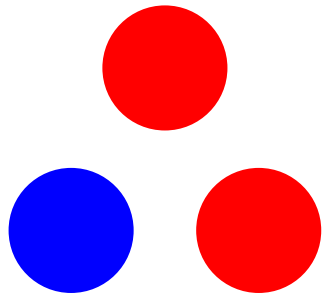
What can the adversary do?

Intuition: adversary can perturb cluster. . .

. . . but large perturbations detected as outliers.

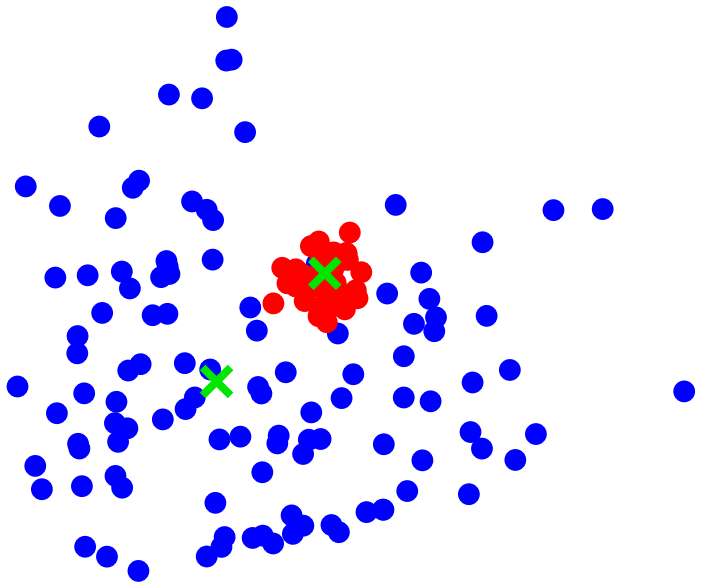
Adversary can also create fake clusters.

- Algorithm: clustering + outlier removal.



Clustering: caveats

Not enough to look at distances:



In high dimensions, can perturb mean by $\sqrt{d}$.

Want to consider global structure $\Longrightarrow$ spectral clustering.

1. Setting & Results

2. Algorithm

Setting: Mean Estimation

Observe αn samples from p^*, $(1-\alpha)n$ arbitrary samples.

Assume p^* has **bounded first moments**:

$$\mathbb{E}[|\langle x - \mu, v\rangle|] \leq \sigma \|v\|_2 \text{ for all } v \in \mathbb{R}^d.$$

Main result:

Theorem

Suppose that p^* has bounded first moments, and we observe $n \geq \frac{d}{\alpha}$ samples. Then with high probability, we can output $m \leq \frac{2}{\alpha}$ candidate means $\hat{\mu}_1, \ldots, \hat{\mu}_m$ such that $\min_{j=1}^{m} \|\hat{\mu}_j - \mu\|_2 \leq \tilde{O}(\sigma/\sqrt{\alpha})$.

Note: can handle more general settings, e.g. fitting exponential families.

Interpretation

Estimate mean to error $\frac{\sigma}{\sqrt{\alpha}}$.

- σ: appears even for $d = 1$, $\alpha = \frac{3}{4}$ (median estimator).
- α: lower bound of $\sigma \cdot \sqrt{\log(1/\alpha)}$, minimax error unknown.
- Non-vanishing error as $n \to \infty$: adversary can perturb cluster without detection.

Why is this good?

- No parametric assumptions
- No dependence on dimension d
- Works even if $\alpha \ll 1$

Comparisons: Mean Estimation

Two recent results on mean estimation if $\alpha > \frac{1}{2}$:

- LRV'16: error $\tilde{\mathcal{O}}(\sigma\sqrt{1-\alpha})$ if p^* has bounded 4th moments
- DKKLMS'16: error $\tilde{\mathcal{O}}(\sigma(1-\alpha))$ if p^* is sub-Gaussian

Our result (holds even if $\alpha \leq \frac{1}{2}$):

- error $\tilde{\mathcal{O}}(\sigma/\sqrt{\alpha})$ if p^* has bounded 1st moments

Comparisons: Clustering

Given samples from mixture $p = \alpha_1 p_1 + \cdots + \alpha_k p_k$, can cluster.

- By thinking of each p_i as p^*.
- List-decodable learning: output means of all clusters.

Existing results (in terms of mean separation $\Delta = \min_{i \neq j} \|\mu_i - \mu_j\|_2$):

- AM'05: can cluster if $\Delta \geq \tilde{\Omega}(\sigma/\sqrt{\alpha})$, 2nd moments
- KK'10/AS'12: can cluster if $\Delta \geq \tilde{\Omega}(\sigma\sqrt{k})$, 2nd moments

Our result: can cluster if $\Delta \geq \tilde{\Omega}(\sigma/\sqrt{\alpha})$, 1st moments.

- Robustness for free!

Algorithm: General Idea

Typical approach: estimate mean by minimizing squared error:

$$\mu = \text{argmin}_\mu \sum_{i=1}^{n} \|\mu - x_i\|_2^2 + \lambda h(\mu)$$

Issue: single outlier can create arbitrary error.

Better idea: give each point its own mean μ_i.

- Constrain them to be close, then cluster.
- Gives wiggle room to accomodate bad points / multiple clusters.

Algorithm: Optimization

Solve following optimization:

$$\begin{aligned} \underset{\mu_1,\ldots,\mu_n,Y}{\text{minimize}} \quad & \sum_{i=1}^{n} \|\mu_i - x_i\|_2^2 + \lambda \operatorname{tr}(Y) \\ \text{subject to} \quad & \mu_i \mu_i^\top \preceq Y \text{ for all } i. \end{aligned}$$

Constraints all μ_i to lie within a **small ellipse**.

- Similar to spectral clustering (penalize nuclear norm of $[\mu_1 \ \cdots \ \mu_n]$).
- Better at handling small clusters.

Issue: Outliers

Adversaries can essentially remove regularization in given direction:

$$x_i = (-10^{100}, 0, 0, \ldots)$$

μ_i will move in direction of x_i regardless of other x_j

- Intuition: outliers pushing their worldview on the rest
- Ellipse expands: other μ_j free to move towards x_i
- Trick good μ_j into overfitting along x_i direction

Need some way of detecting and removing these outlier points.

Idea: Average Nearby Points

If i is a real data point, should have αn other μ_j that "agree" with μ_i.

- Maybe this can help
- Idea: average the αn means μ_j that are closest to x_i
- If result $\tilde{\mu}_i$ has $\|\tilde{\mu}_i - x_i\|_2 \gg \|\mu_i - x_i\|_2$, then i is an outlier

Averaging: Key Properties

Averaging $\mu_{1:n}$ to get $\tilde{\mu}_{1:n}$ has two key properties:

- All μ_i lie within a small ellipse
- $\sum_{i \in C} \|\tilde{\mu}_i - x_i\|_2^2$ is not much larger than $\sum_{i \in C} \|\mu_i - x_i\|_2^2$ for good points C

Second property: low error on good points

- Outlier detection: compute $z_i = \|\tilde{\mu}_i - x_i\|_2^2 - \|\mu_i - x_i\|_2^2$
- Re-weight points: point i gets weight $1 - \eta z_i$

Can show this removes more mass from bad points than good points

Main Algorithm

Initialize $c \leftarrow [1; \cdots; 1] \in \mathbb{R}^n$

while true do

  Let $F(\mu_{1:n}, Y) = \sum_{i=1}^n c_i \|\mu_i - x_i\|_2^2 + \lambda \operatorname{tr}(Y)$.

  $\hat{\mu}_{1:n}, \hat{Y} \leftarrow \textsc{Minimize}(F(\mu_{1:n}, Y)$ subject to $\mu_i \mu_i^\top \preceq Y)$

  Let $\tilde{\mu}_i = \sum_j a_j \hat{\mu}_j$, where a is the solution to ▷ average nearby points

$$\begin{aligned} \text{minimize} \quad & \left\| \textstyle\sum_j a_j \hat{\mu}_j - x_i \right\|_2^2 \\ \text{subject to} \quad & 0 \le a_j \le \tfrac{2}{\alpha n} \text{ for all } j, \quad \textstyle\sum_j a_j = 1 \end{aligned}$$

  if $\operatorname{tr}(Y) \le \frac{6r^2}{\alpha}$ **then** ▷ Check for outliers

    return $\hat{\mu}_{1:n}, \hat{Y}$ ▷ Not many outliers, can return

  else ▷ Re-weight points to down-weight outliers

    $c_i \leftarrow c_i \left(1 - \frac{z_i}{z_{\max}}\right)$, where $z_i = \|\tilde{\mu}_i - x_i\|_2^2 - \|\hat{\mu}_i - x_i\|_2^2$

  end if

end while

Algorithm: Clean-up Steps

At end, need to cluster points μ_i, return centers of each cluster.

- At most $\frac{2}{\alpha}$ clusters of size $\geq \frac{1}{2}\alpha n$
- Also re-run algorithm on each cluster

Algorithm: Summary

- Give each datum its own parameter μ_i
- Constrain to ellipse to encourage consistency
- Downweight outliers (based on average of nearby points)

Discussion

Future directions

- Scale to high dimensions ($\log(n)$ SVD calls?)
- Regression, etc.
- Norms other than ℓ_2