Recovery of sparse signals from a mixture of linear samples

Arya Mazumdar Soumyabrata Pal

University of Massachusetts Amherst

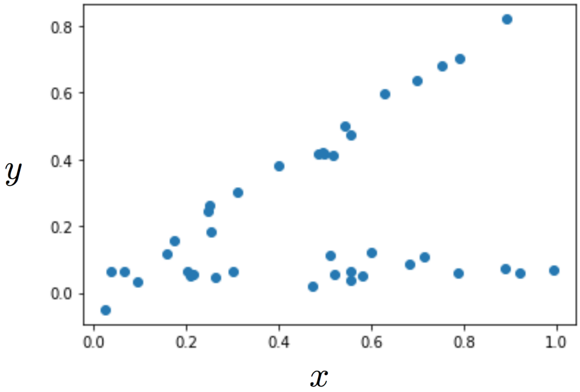
June 15, 2020

ICML 2020

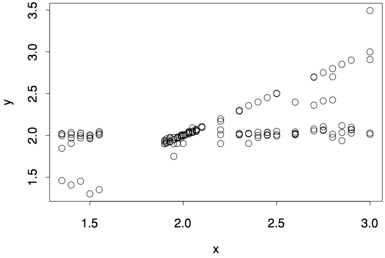
A relationship between features and labels

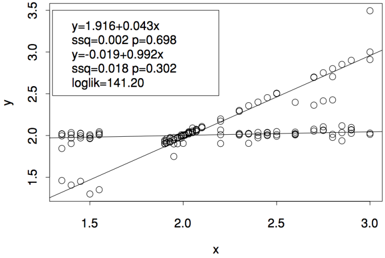
x : `feature` and y : `label`.

Consider the tuple (x, y) with $y = f(x)$:



Example: Music Perception





Music Perception

- Cohen 1980
- De Veaux, 1989;
- Viele and Tong, 2002

Application of Mixture of ML Models

- Multi-modal data, Heterogeneous data
- Recent Works: Stadler, Buhlmann, De Geer, 2010; Faria and Soromenho, 2010; Chaganty and Liang, 2013
- Yi, Caramanis, Sanghavi 2014-2016: Algorithms
- An expressive and rich model
- Modeling a complicated relation as a mixture of simple components
- Advantage: Clean theoretical analysis

Semi-supervised Active Learning framework: Advantages

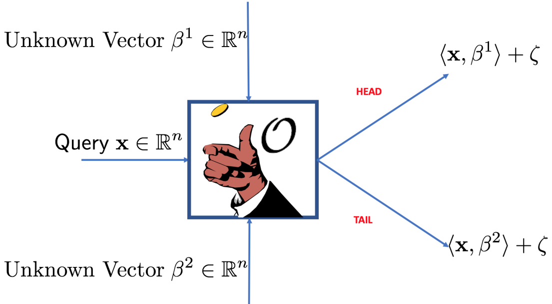
- In this framework, we can carefully design data to query for labels.
- **Objective:** Recover the parameters of the models with minimum number of queries/samples.
- **Advantage:**
  1. Can avoid millions of parameters used by a deep learning model to fit the data!
  2. Learn with significantly less amount of data!
  3. We can use crowd-knowledge which is difficult to incorporate in algorithm.
- Crowdsourcing/ Active Learning has become very popular but is expensive (Dasgupta et. al., Freund et. al.)

Mixture of *sparse* linear regression

- Suppose we have two unknown distinct vectors $\beta^1, \beta^2 \in \mathbb{R}^n$ and an oracle $\mathcal{O} : \mathbb{R}^n \to \mathbb{R}$.
- We assume that β^1, β^2 have k significant entries where $k << n$.
- The oracle $\mathcal{O}$ takes input a vector $\boldsymbol{x} \in \mathbb{R}^n$ and return noisy output (*sample*) $y \in \mathbb{R}$:

$$y = \langle \boldsymbol{x}, \beta \rangle + \zeta$$

  where $\beta \sim_U \{\beta^1, \beta^2\}$ and $\zeta \sim \mathcal{N}(0, \sigma^2)$ with known σ.
- *Generalization of Compressed Sensing*



Mixture of *sparse* linear regression

- We also define the Signal-to-Noise Ratio (SNR) for a query $\boldsymbol{x}$ as:

$$\mathrm{SNR}(\boldsymbol{x}) \triangleq \frac{\mathbb{E}|\langle \mathbf{x}, \boldsymbol{\beta}^1 - \boldsymbol{\beta}^2 \rangle|^2}{\mathbb{E}\zeta^2} \quad \text{and} \quad \mathrm{SNR} = \max_{\boldsymbol{x}} \mathrm{SNR}(\boldsymbol{x})$$

- **Objective:** For each $\beta \in \{\beta^1, \beta^2\}$, we want to recover $\hat{\beta}$ such that

$$||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}|| \leq c||\boldsymbol{\beta} - \boldsymbol{\beta}_{(k)}|| + \gamma$$

where $\boldsymbol{\beta}_{(k)}$ is the best k-sparse approximation of β with minimum queries for a fixed SNR.

Previous and Our results

- First studied by Yin et.al. (2019) who made following assumptions
  1. the unknown vectors are exactly k-sparse, i.e., has at most k nonzero entries;
  2. $\beta_j^1 \neq \beta_j^2$ for each $j \in \text{supp}\boldsymbol{\beta}^1 \cap \text{supp}\boldsymbol{\beta}^2$
  3. for some $\epsilon > 0$, $\boldsymbol{\beta}^1, \boldsymbol{\beta}^2 \in \{0, \pm\epsilon, \pm 2\epsilon, \pm 3\epsilon, \dots\}^n$.

  and showed query complexity exponential in σ/ϵ.
- Krishnamurthy et. al. (2019) removed the first two assumptions but their query complexity was still exponential in $(\sigma/\epsilon)^{2/3}$.
- We get rid of all assumptions and need a query complexity of

$$O\left(\frac{k \log n \log^2 k}{\log(\sigma\sqrt{\text{SNR}}/\gamma)} \max\left(1, \frac{\sigma^4}{\gamma^4\sqrt{\text{SNR}}} + \frac{\sigma^2}{\gamma^2}\right)\right)$$

  which is polynomial in σ.

Insight 1: Compressed Sensing

1. If $\beta^1 = \beta^2$ (single unknown vector), the objective is exactly the same as in Compressed sensing.
2. It is well known (Candes and Tao) that for the following $m \times n$ matrix $\boldsymbol{A}$ with $m = O(k \log n)$,

$$\boldsymbol{A} \triangleq \frac{1}{\sqrt{m}} \begin{bmatrix} \mathcal{N}(0,1) & \mathcal{N}(0,1) & \dots \\ \vdots & \ddots & \\ \mathcal{N}(0,1) & \dots & \mathcal{N}(0,1) \end{bmatrix}$$

   using its rows as queries is sufficient in the CS setting.
3. Can we cluster the samples in our framework?

Insight 2: (Gaussian mixtures)

1. For a given $\mathbf{x} \in \mathbb{R}^n$, repeating $\boldsymbol{x}$ as query to the oracle gives us samples which are distributed according to

$$\frac{1}{2}\mathcal{N}(\langle \boldsymbol{x}, \boldsymbol{\beta}^1 \rangle, \sigma^2) + \frac{1}{2}\mathcal{N}(\langle \boldsymbol{x}, \boldsymbol{\beta}^2 \rangle, \sigma^2).$$

2. With known σ^2, how many samples do we need to recover $\langle \boldsymbol{x}, \boldsymbol{\beta}^1 \rangle, \langle \boldsymbol{x}, \boldsymbol{\beta}^2 \rangle$?

Recover means of Gaussian mixture with same & known variance

Input: Obtain samples from a mixture of Gaussians $\mathcal{M}$ with two components

$$\mathcal{M} \triangleq \frac{1}{2}\mathcal{N}(\mu_1, \sigma^2) + \frac{1}{2}\mathcal{N}(\mu_2, \sigma^2).$$

EM Algorithm	**Method of moments**	**Fit a single Gaussian**

Output: Return $\hat{\mu}_1, \hat{\mu}_2$.

EM algorithm (Daskalakis et.al. 2017, Xu et.al. 2016)

Algorithm 1 EM($\mathbf{x}, \sigma, T$) Estimate the means $\langle \mathbf{x}, \beta^1 \rangle$, $\langle \mathbf{x}, \beta^2 \rangle$ for a query $\mathbf{x}$ using EM algorithm

Require: An oracle $\mathcal{O}$ which when queried with a vector $\mathbf{x} \in \mathbb{R}^n$ returns $\langle \mathbf{x}, \beta \rangle + \mathcal{N}(0, \sigma^2)$ where β is sampled uniformly from $\{\beta^1, \beta^2\}$.

for $i = 1, 2, \ldots, T$ **do**
    Query the oracle $\mathcal{O}$ with $\mathbf{x}$ and obtain a response y^i.
end for
Set the function $w : \mathbb{R}^3 \to \mathbb{R}$ as $w(y, \mu_1, \mu_2) = e^{-(y-\mu_1)^2/2\sigma^2} \left(e^{-(y-\mu_1)^2/2\sigma^2} + e^{-(y-\mu_2)^2/2\sigma^2} \right)^{-1}$.
Initialize $\hat{\mu}_1^0, \hat{\mu}_2^0$ randomly and $t = 0$.
while Until Convergence **do**
    $\hat{\mu}_1^{t+1} = \sum_{i=1}^{T} y_i w(y_i, \hat{\mu}_1^t, \hat{\mu}_2^t) / \sum_{i=1}^{T} w(y_i, \hat{\mu}_1^t, \hat{\mu}_2^t)$.
    $\hat{\mu}_2^{t+1} = \sum_{i=1}^{T} y_i w(y_i, \hat{\mu}_2^t, \hat{\mu}_1^t) / \sum_{i=1}^{T} w(y_i, \hat{\mu}_2^t, \hat{\mu}_1^t)$.
    $t \leftarrow t + 1$.
end while
Return $\hat{\mu}_1^t, \hat{\mu}_2^t$

Method of Moments (Hardt and Price 2015)

- Estimate the first and second central moments

Samples from the mixture

$$y^1 \quad y^2 \quad y^3 \quad y^4 \quad \dots \quad y^T$$

Divide into batches

$$S_1^i = \sum_{j \in \text{ Batch } i} \frac{y^j}{t}$$

$$S_2^i = \sum_{j \in \text{ Batch } i} \frac{(y^j - S_1^i)^2}{t-1}$$

$$\hat{M}_1 = \text{median}(\{S_1^i\}_{i=1}^B)$$

$$\hat{M}_2 = \text{median}(\{S_2^i\}_{i=1}^B)$$

- Set up system of equations to calculate $\hat{\mu}_1, \hat{\mu}_2$ where

$$\hat{\mu}_1 + \hat{\mu}_2 = 2\hat{M}_1, \ (\hat{\mu}_1 - \hat{\mu}_2)^2 = 4\hat{M}_2 - 4\sigma^2$$

Fit a single Gaussian (Daskalakis et. al. 2017)

Estimate the mean $\hat{M}_1$ and return as both $\hat{\mu}_1, \hat{\mu}_2$

Samples from the mixture

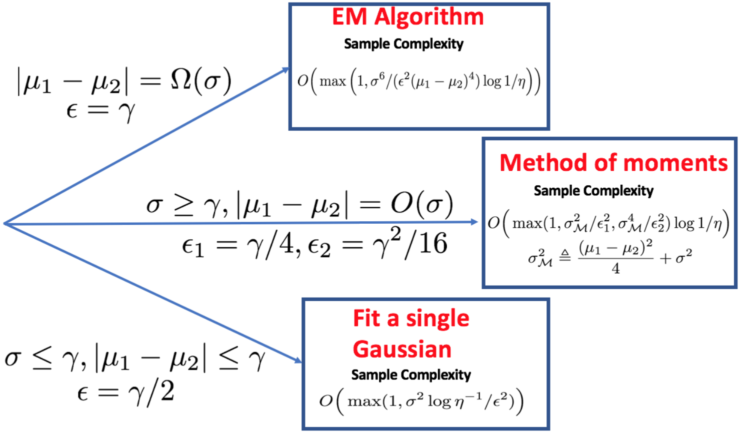
$$y^1 \quad y^2 \quad y^3 \quad y^4 \quad \ldots \quad y^T$$

Sort

$$y^2 \quad y^7 \quad y^{10} \quad y^5 \quad \ldots \quad y^{26}$$

Return average of first and third quartiles

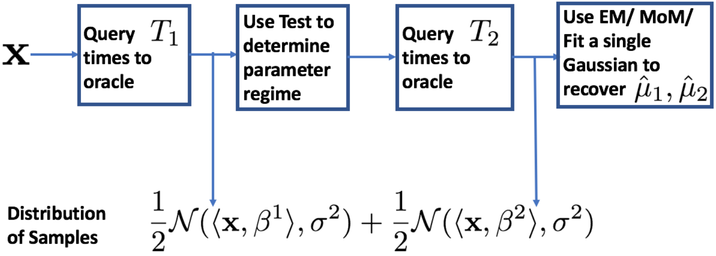
How to choose which algorithm to use



We can design a test to infer the parameter regime correctly.

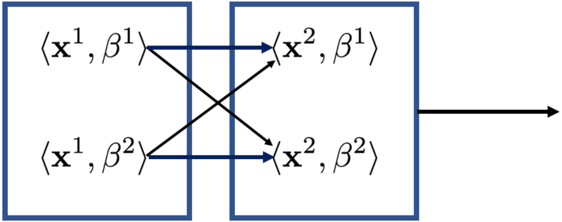
Stage 1: Denoising

We sample $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_{n\times n})$.



- For unknown permutation $\pi : \{1,2\} \to \{1,2\}$, $\hat{\mu}_1, \hat{\mu}_2$ satisfies $\left|\hat{\mu}_i - \mu_{\pi(i)}\right| \leq \gamma$.
- We can show that $\mathbb{E}(T_1 + T_2) \leq O\Big(\big(\frac{\sigma^5}{\gamma^4 ||\boldsymbol{\beta}^1 - \boldsymbol{\beta}^2||_2} + \frac{\sigma^2}{\gamma^2}\big) \log \eta^{-1}\Big)$
- We follow identical steps for $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^m$.

Stage 2: Alignment across queries



Query $\mathbf{x}^1 + \mathbf{x}^2, \mathbf{x}^1 - \mathbf{x}^2$

$\langle \mathbf{x}^1 + \mathbf{x}^2, \beta^1 \rangle$ $\langle \mathbf{x}^1 - \mathbf{x}^2, \beta^1 \rangle$

$\langle \mathbf{x}^1 + \mathbf{x}^2, \beta^2 \rangle$ $\langle \mathbf{x}^1 - \mathbf{x}^2, \beta^2 \rangle$

Only one of the sums is close to

$$\langle \mathbf{x}^1 + \mathbf{x}^2, \beta^1 \rangle$$

Or only one of the differences is close to

$$\langle \mathbf{x}^1 - \mathbf{x}^2, \beta^1 \rangle$$

Stage 3: Cluster & Recover

- After the denoising and alignment steps, we are able to recover two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ of length $m = O(k \log n)$ each such that

$$\left| \boldsymbol{u}[i] - \langle \boldsymbol{x}^i, \boldsymbol{\beta}^{\pi(1)} \rangle \right| \leq 10\gamma; \left| \boldsymbol{v}[i] - \langle \boldsymbol{x}^i, \boldsymbol{\beta}^{\pi(2)} \rangle \right| \leq 10\gamma$$

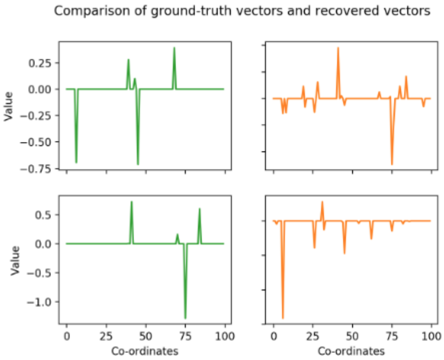
  for some permutation $\pi : \{1, 2\} \to \{1, 2\}$ for all $i \in [m]$ w.p. at least $1 - \eta$.
- We now solve the following convex optimization problems to recover $\hat{\beta}^{\pi(1)}, \hat{\beta}^{\pi(2)}$.

$$\boldsymbol{A} = \frac{1}{\sqrt{m}}[\boldsymbol{x}^1 \quad \boldsymbol{x}^2 \quad \boldsymbol{x}^3 \quad \ldots \quad \boldsymbol{x}^m]^T$$

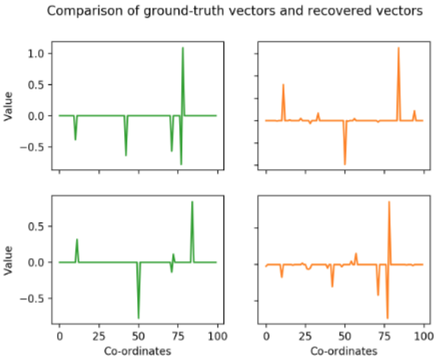
$$\hat{\boldsymbol{\beta}}^{\pi(1)} = \min_{\boldsymbol{z} \in \mathbb{R}^n} ||\boldsymbol{z}||_1 \text{ s.t. } ||\boldsymbol{Az} - \frac{\boldsymbol{u}}{\sqrt{m}}||_2 \leq 10\gamma$$

$$\hat{\boldsymbol{\beta}}^{\pi(2)} = \min_{\boldsymbol{z} \in \mathbb{R}^n} ||\boldsymbol{z}||_1 \text{ s.t. } ||\boldsymbol{Az} - \frac{\boldsymbol{v}}{\sqrt{m}}||_2 \leq 10\gamma$$

Simulations



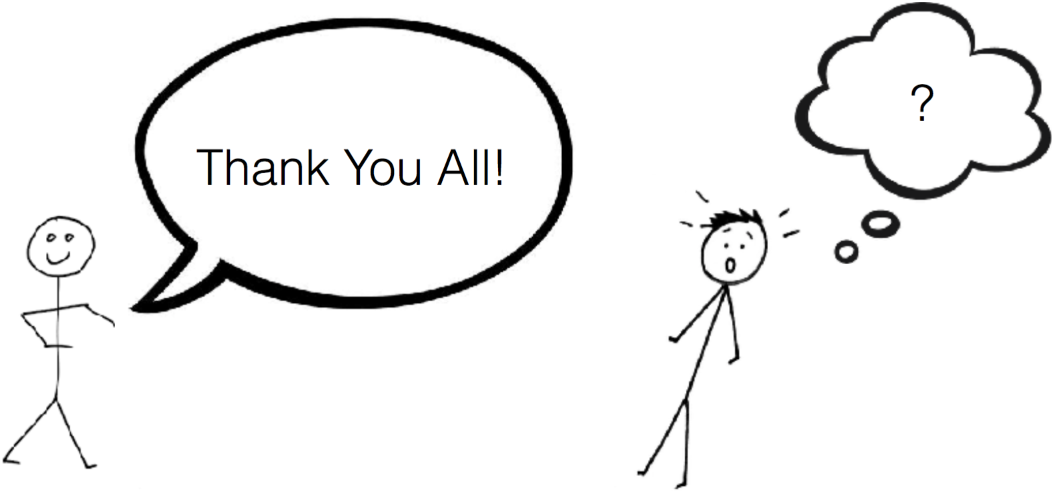
(b) The 100-dimensional ground truth vectors β^1 and β^2 with sparsity $k = 5$ plotted in green (left) and the recovered vectors (using Algorithm 8) $\hat{\beta}^1$ and $\hat{\beta}^2$ plotted in orange (right) using a batch-size ~ 100 for each of 150 random gaussian queries. The order of the recovered vectors and the ground truth vectors is reversed.



(c) The 100-dimensional ground truth vectors β^1 and β^2 with sparsity $k = 5$ plotted in green (left) and the recovered vectors (using Algorithm 8) $\hat{\beta}^1$ and $\hat{\beta}^2$ plotted in orange (right) using a batch-size ~ 600 for each of 150 random gaussian queries. The order of the recovered vectors and the ground truth vectors is reversed.

Conclusion and Future Work

- Our work removes any assumption for two unknown vectors that previous papers depended on.
- Our algorithm contains all main ingredients for extension to larger L. The main technical bottleneck is tight bounds in untangling Gaussian mixtures for more than two components.
- Can we handle other noise distributions?
- Lower bounds on query complexity?


Thank You All!
?