Linear regression without correspondence

Daniel Hsu

Columbia University

October 3, 2017

Joint work with Kevin Shi (Columbia University) and Xiaorui Sun (Microsoft Research).
Linear regression without correspondence

- **Covariate vectors:** $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$
- **Responses:** $y_1, y_2, \ldots, y_n \in \mathbb{R}$
- **Model:**
  $$y_i = \bar{w}^\top x_{\bar{\pi}(i)} + \varepsilon_i, \quad i \in [n]$$
  - Unknown linear function: $\bar{w} \in \mathbb{R}^d$
  - Unknown permutation: $\bar{\pi} \in S_n$
  - Measurement errors: $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in \mathbb{R}$
    (e.g., $(\varepsilon_i)_{i=1}^n$ iid from $N(0, \sigma^2)$)
Linear regression without correspondence

- **Covariate vectors:** $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$
- **Responses:** $y_1, y_2, \ldots, y_n \in \mathbb{R}$
- **Model:**
  
  $$y_i = \bar{w}^\top x_{\bar{\pi}(i)} + \varepsilon_i, \quad i \in [n]$$

  - Unknown linear function: $\bar{w} \in \mathbb{R}^d$
  - Unknown permutation: $\bar{\pi} \in S_n$
  - Measurement errors: $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in \mathbb{R}$
    
    (e.g., $(\varepsilon_i)_{i=1}^n$ iid from $N(0, \sigma^2)$)

Correspondence between $(x_i)_{i=1}^n$ and $(y_i)_{i=1}^n$ is **unknown**.
Example #1: pose and correspondence estimation

- 3D object is captured as a 2D image.
- Some known 3D points on object map to 2D points in image.
Example #1: pose and correspondence estimation

- 3D object is captured as a 2D image.
- Some known 3D points on object map to 2D points in image.

**Goal:** Find mapping between points on object and points in image.

*Perspective projection unknown.*
Example #2: flow cytometry

1. Suspend population of cells in a fluid.
2. Pass cells, one at a time, through laser (via hydrodynamic focusing), and measure emitted light using photomultipliers.
Example #2: flow cytometry

1. Suspend population of cells in a fluid.
2. Pass cells, one at a time, through laser (via hydrodynamic focusing), and measure emitted light using photomultipliers.

Goal: Learn relationship between measurements and cell properties.

Order in which cells pass through laser is unknown.
Prior works — statistical / information-theoretic issues

Unnikrishnan, Haghighatshoar, & Vetterli (2015)

**Question:** If \((x_i)_{i=1}^{n}\) are iid from continuous distribution on \(\mathbb{R}^d\), then how large must \(n\) be so that noiseless measurements uniquely determine every \(\vec{w} \in \mathbb{R}^d\)?

\[ n \geq 2d \text{ is necessary and sufficient.} \]

Elhami, Scholefield, Haro, & Vetterli (2017)

Explicit construction in \(\mathbb{R}^2\): for \(n \geq 4\),

\[ x_i := \begin{bmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{bmatrix} \]

where \(\phi_i := 2\pi \cdot \frac{2i - 1}{2n - 1}, i \in [n].\)
**Unnikrishnan, Haghighatshoar, & Vetterli (2015)**

**Question:** If \((x_i)_{i=1}^n\) are iid from continuous distribution on \(\mathbb{R}^d\), then how large must \(n\) be so that noiseless measurements uniquely determine every \(\bar{w} \in \mathbb{R}^d\)?

**Answer:** \(n \geq 2d\) is necessary and sufficient.
## Prior works — statistical / information-theoretic issues

### Unnikrishnan, Haghighatshoar, & Vetterli (2015)

**Question:** If \((x_i)_{i=1}^n\) are iid from continuous distribution on \(\mathbb{R}^d\), then how large must \(n\) be so that noiseless measurements uniquely determine every \(\bar{w} \in \mathbb{R}^d\)?

**Answer:** \(n \geq 2d\) is necessary and sufficient.

### Elhami, Scholefield, Haro, & Vetterli (2017)

Explicit construction in \(\mathbb{R}^2\): for \(n \geq 4\),

\[
x_i := \begin{bmatrix} \cos(\varphi_i) \\ \sin(\varphi_i) \end{bmatrix} \quad \text{where} \quad \varphi_i := 2\pi \cdot \frac{2^{i-1} - 1}{2^n - 1}, \quad i \in [n].
\]
Question: If \((x_i)_{i=1}^n\) are iid from \(N(0, I_d)\) and \((\varepsilon_i)_{i=1}^n\) are iid from \(N(0, \sigma^2)\), then how large must signal-to-noise ratio \(\text{SNR} = \|\tilde{w}\|^2_2/\sigma^2\) be so that \(\tilde{\pi}\) can be recovered?
Prior works — statistical / information-theoretic issues

Pananjady, Wainwright, & Courtade (2016)

**Question:** If \((x_i)_{i=1}^n\) are iid from \(\mathcal{N}(0, I_d)\) and \((\varepsilon_i)_{i=1}^n\) are iid from \(\mathcal{N}(0, \sigma^2)\), then how large must signal-to-noise ratio \(\text{SNR} = \frac{\|\bar{w}\|_2^2}{\sigma^2}\) be so that \(\bar{\pi} \) can be recovered?

**Answer:** \(\log(1 + \text{SNR}) \gtrsim \log(n)\) is necessary and sufficient. Achieved by maximum likelihood / least squares estimator:

\[
(\hat{w}_{\text{mle}}, \hat{\pi}_{\text{mle}}) := \arg\min_{\mathbf{w} \in \mathbb{R}^d, \pi \in S_n} \sum_{i=1}^n \left(y_i - \mathbf{w}^\top \mathbf{x}_{\pi(i)}\right)^2.
\]
Question: If \((x_i)_{i=1}^n\) are iid from \(N(0, I_d)\) and \((\varepsilon_i)_{i=1}^n\) are iid from \(N(0, \sigma^2)\), then how large must signal-to-noise ratio \(\text{SNR} = \frac{||\bar{w}||_2^2}{\sigma^2}\) be so that \(\bar{\pi}\) can be recovered?

Answer: \(\log(1 + \text{SNR}) \gtrsim \log(n)\) is necessary and sufficient. Achieved by maximum likelihood / least squares estimator:

\[
(\hat{w}_{\text{mle}}, \hat{\pi}_{\text{mle}}) := \arg\min_{w \in \mathbb{R}^d, \pi \in S_n} \sum_{i=1}^n \left(y_i - w^\top x_{\pi(i)}\right)^2.
\]

Note: If correspondence between \((x_i)_{i=1}^n\) and \((y_i)_{i=1}^n\) is known (i.e., standard linear regression setting), then just need \(\text{SNR} \gtrsim d/n\).
Prior works — computational issues

Pananjady, Wainwright, & Courtade (2016)

**Least squares problem**
Given \((x_i)_{i=1}^n\) from \(\mathbb{R}^d\) and \((y_i)_{i=1}^n\) from \(\mathbb{R}\), find

\[
(\hat{w}_{mle}, \hat{\pi}_{mle}) := \arg \min_{w \in \mathbb{R}^d, \pi \in S_n} \sum_{i=1}^n \left( y_i - w^\top x_{\pi(i)} \right)^2.
\]
Prior works — computational issues

Pananjady, Wainwright, & Courtade (2016)

**Least squares problem**

Given \((x_i)_{i=1}^n\) from \(\mathbb{R}^d\) and \((y_i)_{i=1}^n\) from \(\mathbb{R}\), find

\[
(\hat{w}_{\text{mle}}, \hat{\pi}_{\text{mle}}) := \arg \min_{\mathbf{w} \in \mathbb{R}^d, \pi \in S_n} \sum_{i=1}^n (y_i - \mathbf{w}^\top \mathbf{x}_{\pi(i)})^2.
\]

- \(d = 1\): \(O(n \log n)\)-time algorithm based on sorting.

Elhami, Scholefield, Haro, & Vetterli (2017)

\([d = 2]\) \(O(n^3)\)-time algorithm with

\[\|\hat{w} - \bar{w}\|_2 \leq O(2n \cdot \|\varepsilon\|_\infty)\]

when \((x_i)_{i=1}^n\) are exponentially spaced (in angle) on unit circle.
Prior works — computational issues

Pananjady, Wainwright, & Courtade (2016)

**Least squares problem**
Given \((x_i)_{i=1}^n\) from \(\mathbb{R}^d\) and \((y_i)_{i=1}^n\) from \(\mathbb{R}\), find

\[
(\hat{w}_{mle}, \hat{\pi}_{mle}) := \arg\min_{w \in \mathbb{R}^d, \pi \in S_n} \sum_{i=1}^n \left( y_i - w^\top x_{\pi(i)} \right)^2.
\]

* \(d = 1\): \(O(n \log n)\)-time algorithm based on sorting.
* \(d = \Omega(n)\): NP-hard.
Prior works — computational issues

Pananjady, Wainwright, & Courtade (2016)

Least squares problem
Given \((x_i)_{i=1}^n\) from \(\mathbb{R}^d\) and \((y_i)_{i=1}^n\) from \(\mathbb{R}\), find

\[
(\hat{w}_{\text{mle}}, \hat{\pi}_{\text{mle}}) := \arg \min_{\mathbf{w} \in \mathbb{R}^d, \pi \in S_n} \sum_{i=1}^n \left( y_i - \mathbf{w}^\top x_{\pi(i)} \right)^2.
\]

- \(d = 1\): \(O(n \log n)\)-time algorithm based on sorting.
- \(d = \Omega(n)\): NP-hard.

Elhami, Scholefield, Haro, & Vetterli (2017)

\([d = 2]\) \(O(n^3)\)-time algorithm with \(\|\hat{\mathbf{w}} - \bar{\mathbf{w}}\|_2 \leq O(2^n \cdot \|\mathbf{e}\|_\infty)\) when \((x_i)_{i=1}^n\) are exponentially spaced (in angle) on unit circle.
Our contributions

1. Algorithm for least squares that gives \((1 + \epsilon)\)-approximation in time \((n/\epsilon)^{O(k)} + \text{poly}(n, d)\), where \(k = \dim(\text{span}(\mathbf{x}_i)_{i=1}^n)\).
Our contributions

1. Algorithm for least squares that gives $(1 + \epsilon)$-approximation in time $(n/\epsilon)^{O(k)} + \text{poly}(n,d)$, where $k = \dim(\text{span}(x_i)_{i=1}^n)$.

2. poly$(n,d)$-time* algorithm that exactly recovers $\bar{w}$ and $\bar{\pi}$ (with high probability) when $(x_i)_{i=1}^n$ are iid from $\mathcal{N}(0, I_d)$, $\varepsilon_i \equiv 0$, and $n \geq d + 1$. (*After appropriate discretization.)
1. Algorithm for least squares that gives \((1 + \epsilon)\)-approximation in time \((n/\epsilon)^{O(k)} + \text{poly}(n, d)\), where \(k = \dim(\text{span}(x_i)_{i=1}^n)\).

2. \text{poly}(n, d)\text{-time*} algorithm that exactly recovers \(\bar{w}\) and \(\bar{\pi}\) (with high probability) when \((x_i)_{i=1}^n\) are iid from \(\mathcal{N}(0, I_d)\), \(\epsilon_i \equiv 0\), and \(n \geq d + 1\). (*After appropriate discretization.)

3. Information-theoretic lower bounds on SNR for approximate recovery of \(\bar{w}\) when \((\epsilon_i)_{i=1}^n\) are iid from \(\mathcal{N}(0, \sigma^2)\), and \((x_i)_{i=1}^n\) are iid from \(\mathcal{N}(0, I_d)\) or \(\text{Uniform}([-1, 1]^d)\).
1. Approximation algorithm for least squares problem
Least squares problem

Given \((x_i)^n_{i=1}\) from \(\mathbb{R}^d\) and \((y_i)^n_{i=1}\) from \(\mathbb{R}\), find

\[
(\hat{w}_{\text{mle}}, \hat{\pi}_{\text{mle}}) := \arg \min_{w \in \mathbb{R}^d, \pi \in S_n} \sum_{i=1}^{n} \left(y_i - w^\top x_{\pi(i)}\right)^2.
\]

- \(d = 1\): \(O(n \log n)\)-time algorithm based on sorting [PWC’16].
- \(d = \Omega(n)\): NP-hard to decide if OPT = 0 [PWC’16].
- Naïve brute-force search: \(\Omega(|S_n|) = \Omega(n!)\).
Least squares problem \((d = 1)\)

Given \((x_i)_{i=1}^n\) and \((y_i)_{i=1}^n\) from \(\mathbb{R}\), find

\[
(\hat{w}_{\text{mle}}, \hat{\pi}_{\text{mle}}) := \arg\min_{w \in \mathbb{R}, \pi \in S_n} \sum_{i=1}^n (y_i - wx_{\pi(i)})^2.
\]
Least squares problem \((d = 1)\)

Given \((x_i)_{i=1}^{n}\) and \((y_i)_{i=1}^{n}\) from \(\mathbb{R}\), find

\[
(\hat{w}_{\text{mle}}, \hat{\pi}_{\text{mle}}) := \arg \min_{w \in \mathbb{R}, \pi \in S_n} \sum_{i=1}^{n} \left( y_i - wx_{\pi(i)} \right)^2.
\]

Fix \(w \in \mathbb{R}\), and suppose (WLOG) \(w \geq 0\). Then

\[
\min_{\pi \in S_n} \sum_{i=1}^{n} \left( y_i - wx_{\pi(i)} \right)^2 = \sum_{j=1}^{n} \left( y(j) - wx(j) \right)^2
\]

where \(x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}\) and \(y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(n)}\).
Least squares problem \((d = 1)\)

Given \((x_i)_{i=1}^{n}\) and \((y_i)_{i=1}^{n}\) from \(\mathbb{R}\), find

\[
(\hat{w}_{\text{mle}}, \hat{\pi}_{\text{mle}}) := \arg \min_{w \in \mathbb{R}, \pi \in S_n} \sum_{i=1}^{n} \left( y_i - wx_{\pi(i)} \right)^2.
\]

Fix \(w \in \mathbb{R}\), and suppose (WLOG) \(w \geq 0\). Then

\[
\min_{\pi \in S_n} \sum_{i=1}^{n} \left( y_i - wx_{\pi(i)} \right)^2 = \sum_{j=1}^{n} \left( y(j) - wx(j) \right)^2
\]

where \(x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}\) and \(y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(n)}\).

\(\therefore\) As observed by [PWC'16], can find \(\hat{\pi}_{\text{mle}}\) (and \(\hat{w}_{\text{mle}}\)) by sorting.
Alternating minimization

Pick initial $\hat{w} \in \mathbb{R}^d$ (e.g., randomly).
Loop until convergence:

$$\hat{\pi} \leftarrow \arg\min_{\pi \in S_n} \sum_{i=1}^{n} \left( y_i - \hat{w}^\top x_{\pi(i)} \right)^2 .$$

$$\hat{w} \leftarrow \arg\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \left( y_i - w^\top x_{\hat{\pi}(i)} \right)^2 .$$

▶ Each loop-iteration efficiently computable.
▶ But can get stuck in local minima. So try many initial $\hat{w} \in \mathbb{R}^d$.

(questions: How many restarts? How many iterations?)
Alternating minimization

Pick initial $\hat{\mathbf{w}} \in \mathbb{R}^d$ (e.g., randomly). Loop until convergence:

\[
\hat{\pi} \leftarrow \arg \min_{\pi \in S_n} \sum_{i=1}^{n} \left( y_i - \hat{\mathbf{w}}^\top \mathbf{x}_{\pi(i)} \right)^2.
\]

\[
\hat{\mathbf{w}} \leftarrow \arg \min_{\mathbf{w} \in \mathbb{R}^d} \sum_{i=1}^{n} \left( y_i - \mathbf{w}^\top \mathbf{x}_{\hat{\pi}(i)} \right)^2.
\]

- Each loop-iteration efficiently computable.
Alternating minimization

Pick initial $\hat{w} \in \mathbb{R}^d$ (e.g., randomly).
Loop until convergence:

\[
\hat{\pi} \leftarrow \arg\min_{\pi \in S_n} \sum_{i=1}^{n} \left( y_i - \hat{w}^\top x_{\pi(i)} \right)^2.
\]

\[
\hat{w} \leftarrow \arg\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \left( y_i - w^\top x_{\hat{\pi}(i)} \right)^2.
\]

- Each loop-iteration efficiently computable.
- But can get stuck in local minima. So try many initial $\hat{w} \in \mathbb{R}^d$. 
Alternating minimization

Pick initial \( \hat{w} \in \mathbb{R}^d \) (e.g., randomly).
Loop until convergence:

\[
\hat{\pi} \leftarrow \arg \min_{\pi \in S_n} \sum_{i=1}^{n} \left( y_i - \hat{w}^\top x_{\pi(i)} \right)^2.
\]

\[
\hat{w} \leftarrow \arg \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \left( y_i - w^\top x_{\hat{\pi}(i)} \right)^2.
\]

- Each loop-iteration efficiently computable.
- But can get stuck in local minima. So try many initial \( \hat{w} \in \mathbb{R}^d \).

(Questions: How many restarts? How many iterations?)
Approximation result

Theorem

There is an algorithm that given any inputs \((x_i)_{i=1}^n, (y_i)_{i=1}^n\), and \(\epsilon \in (0, 1)\), returns a \((1 + \epsilon)\)-approximate solution to the least squares problem in time

\[
\left( \frac{n}{\epsilon} \right)^{O(k)} + \text{poly}(n, d),
\]

where \(k = \dim(\text{span}(x_i)_{i=1}^n)\).
Beating brute-force search: “realizable” case

“Realizable” case: Suppose there exist $w_\star \in \mathbb{R}^d$ and $\pi_\star \in S_n$ s.t.

$$y_i = w_\star^T x_{\pi_\star(i)}, \quad i \in [n].$$
Beating brute-force search: “realizable” case

“Realizable” case: Suppose there exist $w_\star \in \mathbb{R}^d$ and $\pi_\star \in S_n$ s.t.

$$y_i = w_\star^\top x_{\pi_\star}(i), \quad i \in [n].$$

Solution is determined by action of $\pi_\star$ on $d$ points (assume $\dim(\text{span}(x_i)_{i=1}^d) = d$).
Beating brute-force search: “realizable” case

“Realizable” case: Suppose there exist $\mathbf{w}_* \in \mathbb{R}^d$ and $\pi_* \in S_n$ s.t.

$$y_i = \mathbf{w}_*^\top \mathbf{x}_{\pi_*(i)}, \quad i \in [n].$$

Solution is determined by action of $\pi_*$ on $d$ points
(assume $\dim(\text{span}(\mathbf{x}_i)_{i=1}^d) = d$).

Algorithm:

- Find subset of $d$ linearly independent points $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_d}$.
- “Guess” values of $\pi_*^{-1}(i_j) \in [d], \ j \in [d]$.
- Solve linear system $y_{\pi_*^{-1}(i_j)} = \mathbf{w}^\top \mathbf{x}_{i_j}, \ j \in [d]$, for $\mathbf{w} \in \mathbb{R}^d$.
- To check correctness of $\hat{\mathbf{w}}$: compute $\hat{y}_i := \hat{\mathbf{w}}^\top \mathbf{x}_i, \ i \in [n]$, and check
  if $\min_{\pi \in S_n} \sum_{i=1}^n (y_i - \hat{y}_{\pi(i)})^2 = 0$. 

"Guess" means "enumerate over ($n^d$) choices"; rest is $\text{poly}(n, d)$. 

Beating brute-force search: “realizable” case

“Realizable” case: Suppose there exist $\mathbf{w}_* \in \mathbb{R}^d$ and $\pi_* \in S_n$ s.t.

$$y_i = \mathbf{w}_*^\top \mathbf{x}_{\pi_*(i)}, \quad i \in [n].$$

Solution is determined by action of $\pi_*$ on $d$ points
(assume $\dim(\text{span}(\mathbf{x}_i)_{i=1}^d) = d$).

Algorithm:

- Find subset of $d$ linearly independent points $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_d}$.
- “Guess” values of $\pi_*^{-1}(i_j) \in [d], \ j \in [d]$.
- Solve linear system $y_{\pi_*^{-1}(i_j)} = \mathbf{w}^\top \mathbf{x}_{i_j}, \ j \in [d], \ \text{for} \ \mathbf{w} \in \mathbb{R}^d$.
- To check correctness of $\hat{\mathbf{w}}$: compute $\hat{y}_i := \hat{\mathbf{w}}^\top \mathbf{x}_i, \ i \in [n]$, and check if $\min_{\pi \in S_n} \sum_{i=1}^n (y_i - \hat{y}_{\pi(i)})^2 = 0$.

“Guess” means “enumerate over $\binom{n}{d}$ choices”; rest is $\text{poly}(n,d)$.
Beating brute-force search: general case

**General case**: solution may not be determined by only $d$ points.

\[
\text{Follows from result of Dereziński and Warmuth (2017) on volume sampling.}
\]

\[
\Rightarrow n \tilde{O}(d) \text{-time algorithm with approximation ratio } d+1,
\]

\[
\text{or } n \tilde{O}(d/\epsilon) \text{-time algorithm with approximation ratio } 1 + \epsilon.
\]

Better way to get $1 + \epsilon$: exploit first-order optimality conditions (i.e., "normal equations") and \( \epsilon \)-nets.

Overall time:
\[
\left( \frac{n}{\epsilon} \right) \tilde{O}(k) + \text{poly}(n,d) \text{ for } k = \dim(\text{span}(x_i)_{i=1}^n).
\]
Beating brute-force search: general case

**General case:** solution may not be determined by only $d$ points.

But, for any RHS $b \in \mathbb{R}^n$, there exist $x_{i_1}, x_{i_2}, \ldots, x_{i_d}$ s.t. every $\hat{w} \in \arg \min_{w \in \mathbb{R}^d} \sum_{j=1}^{d} (b_{i_j} - w^\top x_{i_j})^2$ satisfies

$$\sum_{i=1}^{n} (b_i - \hat{w}^\top x_i)^2 \leq (d + 1) \cdot \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (b_i - w^\top x_i)^2.$$ 

(Follows from result of Dereziński and Warmuth (2017) on volume sampling.)
Beating brute-force search: general case

**General case**: solution may not be determined by only $d$ points.

But, for any RHS $b \in \mathbb{R}^n$, there exist $x_{i_1}, x_{i_2}, \ldots, x_{i_d}$ s.t. every $\hat{w} \in \arg \min_{w \in \mathbb{R}^d} \sum_{j=1}^{d} (b_{i_j} - w^\top x_{i_j})^2$ satisfies

$$
\sum_{i=1}^{n} (b_i - \hat{w}^\top x_i)^2 \leq (d + 1) \cdot \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (b_i - w^\top x_i)^2.
$$

(Follows from result of Dereziński and Warmuth (2017) on volume sampling.)

$\implies n^{O(d)}$-time algorithm with approximation ratio $d + 1$, or $n^{O(d/\epsilon)}$-time algorithm with approximation ratio $1 + \epsilon$. 

Better way to get $1 + \epsilon$: exploit first-order optimality conditions (i.e., “normal equations”) and $\epsilon$-nets.

Overall time: $\mathcal{O}(n/\epsilon)k + \text{poly}(n,d)$ for $k = \dim(\text{span}(x_{i_n}))$. 

15
Beating brute-force search: general case

**General case:** solution may not be determined by only $d$ points.

But, for any RHS $b \in \mathbb{R}^n$, there exist $x_{i_1}, x_{i_2}, \ldots, x_{i_d}$ s.t. every $\hat{w} \in \text{arg min}_{w \in \mathbb{R}^d} \sum_{j=1}^{d} (b_{i_j} - w^\top x_{i_j})^2$ satisfies

$$\sum_{i=1}^{n} (b_i - \hat{w}^\top x_i)^2 \leq (d + 1) \cdot \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (b_i - w^\top x_i)^2.$$

(Follows from result of Dereziński and Warmuth (2017) on volume sampling.)

$\implies n^{O(d)}$-time algorithm with approximation ratio $d + 1$, or $n^{O(d/\epsilon)}$-time algorithm with approximation ratio $1 + \epsilon$.

**Better way to get** $1 + \epsilon$: exploit first-order optimality conditions (i.e., “normal equations”) and $\epsilon$-nets.

**Overall time:** $(n/\epsilon)^{O(k)} + \text{poly}(n, d)$ for $k = \dim(\text{span}(x_i)_{i=1}^{n})$. 
Remarks

- Algorithm is justified in statistical setting by results of [PWC'16] for MLE, but guarantees also hold when inputs are worst-case.
- Algorithm is poly-time only when $k = O(1)$.

Open problems:
1. Poly-time approximation algorithm when $k = \omega(1)$.
   Perhaps in average-case or smoothed setting.
2. (Smoothed) analysis of alternating minimization. Similar to Lloyd's algorithm for Euclidean $k$-means.

Next: Algorithm for noise-free average-case setting.
Remarks

- Algorithm is justified in statistical setting by results of [PWC'16] for MLE, but guarantees also hold when inputs are worst-case.
- Algorithm is poly-time only when $k = O(1)$.

Open problems:

1. Poly-time approximation algorithm when $k = \omega(1)$. (Perhaps in average-case or smoothed setting.)
2. (Smoothed) analysis of alternating minimization. Similar to Lloyd’s algorithm for Euclidean $k$-means.
Remarks

- Algorithm is justified in statistical setting by results of [PWC’16] for MLE, but guarantees also hold when inputs are worst-case.
- Algorithm is poly-time only when $k = O(1)$.

Open problems:
1. Poly-time approximation algorithm when $k = \omega(1)$.
   (Perhaps in average-case or smoothed setting.)
2. (Smoothed) analysis of alternating minimization.
   Similar to Lloyd’s algorithm for Euclidean $k$-means.

Next: Algorithm for \textit{noise-free} average-case setting.
2. Exact recovery in the noise-free Gaussian setting
Setting

**Noise-free Gaussian linear model (with \( n + 1 \) measurements):**

\[
y_i = \bar{w}^\top x_{\bar{\pi}(i)}, \quad i \in \{0, 1, \ldots, n\}
\]

- Covariate vectors: \((x_i)_{i=0}^n\) iid from \(\mathcal{N}(0, I_d)\)
- Unknown linear function: \(\bar{w} \in \mathbb{R}^d\)
- Unknown permutation: \(\bar{\pi} \in S\{0,1,\ldots,n\}\)
Setting

**Noise-free Gaussian linear model** (with $n + 1$ measurements):

$$y_i = \bar{w}^\top x_{\bar{\pi}(i)}, \quad i \in \{0, 1, \ldots, n\}$$

- Covariate vectors: $(x_i)_{i=0}^n$ iid from $\mathcal{N}(0, I_d)$
- Unknown linear function: $\bar{w} \in \mathbb{R}^d$
- Unknown permutation: $\bar{\pi} \in S\{0,1,\ldots,n\}$

“Equivalent” problem: We’re promised that $\bar{\pi}(0) = 0$. 
Setting

Noise-free Gaussian linear model (with \( n + 1 \) measurements):

\[
y_i = \bar{w}^\top \bar{x}_{\bar{\pi}(i)}, \quad i \in \{0, 1, \ldots, n\}
\]

- Covariate vectors: \((x_i)_{i=0}^{n}\) iid from \(N(0, I_d)\)
- Unknown linear function: \(\bar{w} \in \mathbb{R}^d\)
- Unknown permutation: \(\bar{\pi} \in S\{0,1,\ldots,n\}\)

"Equivalent" problem: We’re promised that \(\bar{\pi}(0) = 0\).

So can just consider \(\bar{\pi}\) as unknown permutation over \(\{1, 2, \ldots, n\}\).
Setting

**Noise-free Gaussian linear model** (with \( n + 1 \) measurements):

\[
y_i = \bar{\mathbf{w}}^\top \mathbf{x}_{\bar{\pi}(i)} , \quad i \in \{0, 1, \ldots, n\}
\]

- Covariate vectors: \((\mathbf{x}_i)_{i=0}^n\) iid from \(\mathcal{N}(0, \mathbf{I}_d)\)
- Unknown linear function: \(\bar{\mathbf{w}} \in \mathbb{R}^d\)
- Unknown permutation: \(\bar{\pi} \in S\{0,1,\ldots,n\}\)

“Equivalent” problem: We’re promised that \(\bar{\pi}(0) = 0\).
So can just consider \(\bar{\pi}\) as unknown permutation over \(\{1, 2, \ldots, n\}\).

**Number of measurements:**
If \(n + 1 \geq d\), then recovery of \(\bar{\pi}\) gives exact recovery of \(\bar{\mathbf{w}}\) (a.s.).
Setting

**Noise-free Gaussian linear model** (with $n + 1$ measurements):

$$y_i = \bar{w}^\top x_{\bar{\pi}(i)}, \quad i \in \{0, 1, \ldots, n\}$$

- Covariate vectors: $(x_i)_{i=0}^n$ iid from $\mathcal{N}(0, I_d)$
- Unknown linear function: $\bar{w} \in \mathbb{R}^d$
- Unknown permutation: $\bar{\pi} \in S\{0, 1, \ldots, n\}$

"Equivalent" problem: We’re promised that $\bar{\pi}(0) = 0$.

So can just consider $\bar{\pi}$ as unknown permutation over $\{1, 2, \ldots, n\}$.

**Number of measurements:**

If $n + 1 \geq d$, then recovery of $\bar{\pi}$ gives exact recovery of $\bar{w}$ (a.s.).

We’ll assume $n + 1 \geq d + 1$ (i.e., $n \geq d$).
Setting

**Noise-free Gaussian linear model** (with \( n + 1 \) measurements):

\[
y_i = \bar{w}^\top x_{\bar{\pi}(i)}, \quad i \in \{0, 1, \ldots, n\}
\]

- Covariate vectors: \((x_i)_{i=0}^n\) iid from \(N(0, I_d)\)
- Unknown linear function: \(\bar{w} \in \mathbb{R}^d\)
- Unknown permutation: \(\bar{\pi} \in S\{0,1,\ldots,n\}\)

"Equivalent" problem: We’re promised that \(\bar{\pi}(0) = 0\). So can just consider \(\bar{\pi}\) as unknown permutation over \(\{1, 2, \ldots, n\}\).

**Number of measurements:**
If \(n + 1 \geq d\), then recovery of \(\bar{\pi}\) gives exact recovery of \(\bar{w}\) (a.s.).
We’ll assume \(n + 1 \geq d + 1\) (i.e., \(n \geq d\)).

**Claim:** \(n \geq d\) suffices to recover \(\bar{\pi}\) with high probability.
Exact recovery result

Theorem

Fix any $\bar{w} \in \mathbb{R}^d$ and $\bar{\pi} \in S_n$, and assume $n \geq d$. Suppose $(x_i)_{i=0}^n$ are drawn iid from $\mathcal{N}(\mathbf{0}, I_d)$, and $(y_i)_{i=0}^n$ satisfy

$$y_0 = \bar{w}^\top x_0; \quad y_i = \bar{w}^\top x_{\bar{\pi}(i)}, \quad i \in [n].$$

There is an algorithm that, given inputs $(x_i)_{i=0}^n$ and $(y_i)_{i=0}^n$, returns $\bar{\pi}$ and $\bar{w}$ with high probability.
Main idea: hidden subset

Measurements:

\[ y_0 = \bar{w}^\top x_0; \quad y_i = \bar{w}^\top x_{\bar{\pi}(i)}, \quad i \in [n]. \]
Main idea: hidden subset

Measurements:

\[ y_0 = \bar{\mathbf{w}}^\top \mathbf{x}_0 ; \quad y_i = \bar{\mathbf{w}}^\top \mathbf{x}_{\bar{\pi}(i)} , \quad i \in [n]. \]

**For simplicity:** assume \( n = d \), and \( \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d \) orthonormal.
Main idea: hidden subset

Measurements:

\[ y_0 = \bar{w}^\top x_0; \quad y_i = \bar{w}^\top x_{\bar{\pi}(i)}, \quad i \in [n]. \]

**For simplicity:** assume \( n = d \), and \( x_1, x_2, \ldots, x_d \) orthonormal.

\[
y_0 = \bar{w}^\top x_0 = \sum_{j=1}^{d} (\bar{w}^\top x_j) (x_j^\top x_0)
\]
Main idea: hidden subset

Measurements:

\[ y_0 = \bar{w}^\top x_0; \quad y_i = \bar{w}^\top x_{\bar{\pi}(i)}, \quad i \in [n]. \]

For simplicity: assume \( n = d \), and \( x_1, x_2, \ldots, x_d \) orthonormal.

\[ y_0 = \bar{w}^\top x_0 = \sum_{j=1}^{d} (\bar{w}^\top x_j) (x_j^\top x_0) \]

\[ = \sum_{j=1}^{d} y_{\bar{\pi}^{-1}(j)} (x_j^\top x_0) \]
Main idea: hidden subset

Measurements:

\[ y_0 = \mathbf{w}^\top \mathbf{x}_0 ; \quad \quad y_i = \mathbf{w}^\top \mathbf{x}_{\bar{\pi}(i)} , \quad i \in [n]. \]

For simplicity: assume \( n = d \), and \( \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d \) orthonormal.

\[ y_0 = \mathbf{w}^\top \mathbf{x}_0 = \sum_{j=1}^{d} (\mathbf{w}^\top \mathbf{x}_j) (\mathbf{x}_j^\top \mathbf{x}_0) \]

\[ = \sum_{j=1}^{d} y_{\bar{\pi}^{-1}(j)} (\mathbf{x}_j^\top \mathbf{x}_0) \]

\[ = \sum_{i=1}^{d} \sum_{j=1}^{d} \mathbb{1}\{\bar{\pi}(i) = j\} \cdot y_i (\mathbf{x}_j^\top \mathbf{x}_0). \]
Reduction to subset sum

\[ y_0 = \sum_{i=1}^{d} \sum_{j=1}^{d} 1\{\pi(i) = j\} \cdot y_i \begin{pmatrix} x_j^\top \end{pmatrix} x_0 \]

\[ c_{i,j} \]

Promised that a size-\(d\) subset of the \(c_{i,j}\) sum to \(T\). Correct subset corresponds to \((i,j) \in \mathbb{Z}^2 \) s.t. \(\pi(i) = j\).

Next: How to solve Subset Sum efficiently?
Reduction to subset sum

\[ y_0 = \sum_{i=1}^{d} \sum_{j=1}^{d} 1\{\bar{\pi}(i) = j\} \cdot y_i (x_j^\top x_0) \]

- \( d^2 \) “source” numbers \( c_{i,j} := y_i (x_j^\top x_0) \), “target” sum \( T := y_0 \).

Promised that a size-\( d \) subset of the \( c_{i,j} \) sum to \( T \).
Reduction to subset sum

\[ y_0 = \sum_{i=1}^{d} \sum_{j=1}^{d} 1\{\bar{\pi}(i) = j\} \cdot y_i \left( x_j^\top x_0 \right) \]

- \( d^2 \) “source” numbers \( c_{i,j} := y_i(x_j^\top x_0) \), “target” sum \( T := y_0 \).
  Promised that a size-\( d \) subset of the \( c_{i,j} \) sum to \( T \).

- **Correct subset** corresponds to \((i, j) \in [d]^2\) s.t. \( \bar{\pi}(i) = j \).
Reduction to subset sum

\[ y_0 = \sum_{i=1}^{d} \sum_{j=1}^{d} 1\{\bar{\pi}(i) = j\} \cdot y_i (\mathbf{x}_j^\top \mathbf{x}_0) \]

- \( d^2 \) “source” numbers \( c_{i,j} := y_i (\mathbf{x}_j^\top \mathbf{x}_0) \), “target” sum \( T := y_0 \).
  Promised that a size-\( d \) subset of the \( c_{i,j} \) sum to \( T \).
- Correct subset corresponds to \( (i, j) \in [d]^2 \) s.t. \( \bar{\pi}(i) = j \).

Next: How to solve Subset Sum efficiently?
Reducing subset sum to shortest vector problem

Lagarias & Odlyzko (1983): random instances of Subset Sum efficiently solvable when $N$ source numbers chosen independently and u.a.r. from sufficiently wide interval of $\mathbb{Z}$. 

Main idea: (w.h.p.) every incorrect subset will “miss” the target sum $T$ by noticeable amount.

Reduction: construct lattice basis in $\mathbb{R}^{N+1}$ such that

$\begin{bmatrix} b_0 & b_1 & \cdots & b_N \end{bmatrix} = 0 \mathbf{I}_N \beta T - \beta c_1 \cdots - \beta c_N$ for sufficiently large $\beta > 0$.

Using Lenstra, Lenstra, & Lovász (1982) algorithm to find approximately shortest vector reveals correct subset.
Reducing subset sum to shortest vector problem

Lagarias & Odlyzko (1983): random instances of Subset Sum efficiently solvable when $N$ source numbers chosen independently and u.a.r. from sufficiently wide interval of $\mathbb{Z}$.

Main idea: (w.h.p.) every incorrect subset will “miss” the target sum $T$ by noticeable amount.
Reducing subset sum to shortest vector problem

Lagarias & Odlyzko (1983): random instances of Subset Sum efficiently solvable when $N$ source numbers chosen independently and u.a.r. from sufficiently wide interval of $\mathbb{Z}$.

Main idea: (w.h.p.) every incorrect subset will “miss” the target sum $T$ by noticeable amount.

Reduction: construct lattice basis in $\mathbb{R}^{N+1}$ such that

- correct subset of basis vectors gives short lattice vector $v_*$;
- any other lattice vector $\not\propto v_*$ is more than $2^{N/2}$-times longer.
Reducing subset sum to shortest vector problem

**Lagarias & Odlyzko (1983):** random instances of Subset Sum efficiently solvable when \( N \) source numbers chosen independently and u.a.r. from sufficiently wide interval of \( \mathbb{Z} \).

*Main idea:* (w.h.p.) every incorrect subset will “miss” the target sum \( T \) by noticeable amount.

*Reduction:* construct lattice basis in \( \mathbb{R}^{N+1} \) such that
  - correct subset of basis vectors gives short lattice vector \( \mathbf{v}_\star \);
  - any other lattice vector \( \not \propto \mathbf{v}_\star \) is more than \( 2^{N/2} \)-times longer.

\[
\begin{bmatrix}
  b_0 & b_1 & \cdots & b_N \\
\end{bmatrix} = \begin{bmatrix}
  0 & I_N \\
  \beta T & -\beta c_1 & \cdots & -\beta c_N \\
\end{bmatrix}
\]

for sufficiently large \( \beta > 0 \).
Reducing subset sum to shortest vector problem

Lagarias & Odlyzko (1983): random instances of Subset Sum efficiently solvable when $N$ source numbers chosen independently and u.a.r. from sufficiently wide interval of $\mathbb{Z}$.

Main idea: (w.h.p.) every incorrect subset will “miss” the target sum $T$ by noticeable amount.

Reduction: construct lattice basis in $\mathbb{R}^{N+1}$ such that

- correct subset of basis vectors gives short lattice vector $v_*$;
- any other lattice vector $\propto v_*$ is more than $2^{N/2}$-times longer.

$$\begin{bmatrix} b_0 & b_1 & \cdots & b_N \end{bmatrix} = \begin{bmatrix} 0 & I_N \\ \beta T & -\beta c_1 & \cdots & -\beta c_N \end{bmatrix}$$

for sufficiently large $\beta > 0$.

Our random subset sum instance

**Catch:** Our source numbers $c_{i,j} = y_i x_j^T x_0$ are **not independent**, and not **uniformly distributed** on some wide interval of $\mathbb{Z}$. 
Our random subset sum instance

**Catch:** Our source numbers \( c_{i,j} = y_i x_j^\top x_0 \) are not independent, and not uniformly distributed on some wide interval of \( \mathbb{Z} \).

- Instead, have some joint density derived from \( \mathcal{N}(0,1) \).
Our random subset sum instance

**Catch:** Our source numbers \( c_{i,j} = y_i x_j^\top x_0 \) are **not independent**, and **not uniformly distributed** on some wide interval of \( \mathbb{Z} \).

- Instead, have some joint density derived from \( \mathcal{N}(0, 1) \).

- To show that Lagarias & Odlyzko reduction still works, need Gaussian anti-concentration for quadratic and quartic forms.
Our random subset sum instance

**Catch:** Our source numbers \( c_{i,j} = y_i x_j^\top x_0 \) are **not independent**, and **not uniformly distributed** on some wide interval of \( \mathbb{Z} \).

- Instead, have some joint density derived from \( N(0, 1) \).

- To show that Lagarias & Odlyzko reduction still works, need Gaussian anti-concentration for quadratic and quartic forms.

**Key lemma:** (w.h.p.) for every \( Z \in \mathbb{Z}^{d \times d} \) that is not an integer multiple of permutation matrix corresponding to \( \bar{\pi} \),

\[
T - \sum_{i,j} Z_{i,j} \cdot c_{i,j} \geq \frac{1}{2^{\text{poly}(d)}} \cdot \|\bar{w}\|_2.
\]
Some details

- In general, $x_1, x_2, \ldots, x_n$ are not (exactly) orthonormal, but similar reduction works via Moore-Penrose pseudoinverse.
Some details

- In general, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are not (exactly) orthonormal, but similar reduction works via Moore-Penrose pseudoinverse.

- Reduction uses real coefficients in lattice basis.
  
  For LLL to run in poly-time, need to round $(\mathbf{x}_i)_{i=0}^n$ and $\bar{\mathbf{w}}$ coefficients to finite-precision rational numbers.
  
  Similar to drawing $(\mathbf{x}_i)_{i=0}^n$ iid from discretized $\mathcal{N}(\mathbf{0}, I_d)$. 

Some details

- In general, $x_1, x_2, \ldots, x_n$ are not (exactly) orthonormal, but similar reduction works via Moore-Penrose pseudoinverse.

- Reduction uses real coefficients in lattice basis. For LLL to run in poly-time, need to round $(x_i)_{i=0}^n$ and $\bar{w}$ coefficients to finite-precision rational numbers.
  Similar to drawing $(x_i)_{i=0}^n$ iid from discretized $N(0, I_d)$.

- Algorithm strongly exploits assumption of noise-free measurements; likely fails in presence of noise.
Some details

- In general, \(x_1, x_2, \ldots, x_n\) are not (exactly) orthonormal, but similar reduction works via Moore-Penrose pseudoinverse.

- Reduction uses real coefficients in lattice basis.
  
  For LLL to run in poly-time, need to round \((x_i)_{i=0}^n\) and \(\bar{w}\) coefficients to finite-precision rational numbers.
  
  Similar to drawing \((x_i)_{i=0}^n\) iid from discretized \(\mathcal{N}(0, I_d)\).

- Algorithm strongly exploits assumption of noise-free measurements; likely fails in presence of noise.

- Similar algorithm used by Andoni, H., Shi, & Sun (2017) for different problems (phase retrieval / correspondence retrieval).
Connections to prior works

- Unnikrishnan, Haghighatshoar, & Vetterli (2015)

**Recall:** [UHV’15] show that \( n \geq 2d \) is necessary for measurements to uniquely determine every \( \bar{w} \in \mathbb{R}^d \).

- Pananjady, Wainwright, & Courtade (2016)

Noise-free setting: signal-to-noise conditions trivially satisfied whenever \( \bar{w} \neq 0 \).

Noisy setting: recovering \( \bar{w} \) may be easier than recovering \( \bar{w}' \in \mathbb{R}^d \).

Next: Limits for recovering \( \bar{w} \).
Connections to prior works

- Unnikrishnan, Haghighatshoar, & Vetterli (2015)

  Recall: [UHV’15] show that $n \geq 2d$ is necessary for measurements to uniquely determine every $\bar{w} \in \mathbb{R}^d$.

  Our result: For fixed $\bar{w} \in \mathbb{R}^d$, $d + 1$ measurements suffice to recover $\bar{w}$; same covariate vectors may fail for other $\bar{w}' \in \mathbb{R}^d$.

  (C.f. “for all” vs. “for each” results in compressive sensing.)
Connections to prior works

▶ Unnikrishnan, Haghighatshoar, & Vetterli (2015)

**Recall**: [UHV’15] show that $n \geq 2d$ is necessary for measurements to uniquely determine every $\bar{w} \in \mathbb{R}^d$.

**Our result**: For fixed $\bar{w} \in \mathbb{R}^d$, $d + 1$ measurements suffice to recover $\bar{w}$; same covariate vectors may fail for other $\bar{w}' \in \mathbb{R}^d$.

(C.f. “for all” vs. “for each” results in compressive sensing.)

▶ Pananjady, Wainwright, & Courtade (2016)

**Noise-free setting**: signal-to-noise conditions trivially satisfied whenever $\bar{w} \neq 0$. 
Connections to prior works

▶ **Unnikrishnan, Haghighatshoar, & Vetterli (2015)**

**Recall:** [UHV’15] show that $n \geq 2d$ is necessary for measurements to uniquely determine every $\bar{w} \in \mathbb{R}^d$.

**Our result:** For fixed $\bar{w} \in \mathbb{R}^d$, $d + 1$ measurements suffice to recover $\bar{w}$; same covariate vectors may fail for other $\bar{w}' \in \mathbb{R}^d$.

(C.f. “for all” vs. “for each” results in compressive sensing.)

▶ **Pananjady, Wainwright, & Courtade (2016)**

**Noise-free setting:** signal-to-noise conditions trivially satisfied whenever $\bar{w} \neq 0$.

**Noisy setting:** recovering $\bar{w}$ may be easier than recovering $\bar{\pi}$. 
Connections to prior works

- **Unnikrishnan, Haghighatshoar, & Vetterli (2015)**
  
  **Recall:** [UHV’15] show that \( n \geq 2d \) is necessary for measurements to uniquely determine every \( \bar{w} \in \mathbb{R}^d \).

  **Our result:** For **fixed** \( \bar{w} \in \mathbb{R}^d \), \( d + 1 \) measurements suffice to recover \( \bar{w} \); same covariate vectors may fail for other \( \bar{w}' \in \mathbb{R}^d \).

  (C.f. “for all” vs. “for each” results in compressive sensing.)

- **Pananjady, Wainwright, & Courtade (2016)**
  
  **Noise-free setting:** signal-to-noise conditions trivially satisfied whenever \( \bar{w} \neq 0 \).

  **Noisy setting:** recovering \( \bar{w} \) may be easier than recovering \( \bar{\pi} \).

**Next:** Limits for recovering \( \bar{w} \).
3. Lower bounds on SNR for approximate recovery
Setting

Linear model with Gaussian noise

\[ y_i = \bar{w}^\top x_{\bar{\pi}(i)} + \varepsilon_i, \quad i \in [n] \]

- Covariate vectors: \((x_i)_{i=1}^n\) iid from \(P\)
- Measurement errors: \((\varepsilon_i)\) iid from \(N(0, \sigma^2)\)
- Unknown linear function: \(\bar{w} \in \mathbb{R}^d\)
- Unknown permutation: \(\bar{\pi} \in S_n\)
Setting

**Linear model with Gaussian noise**

\[ y_i = \bar{w}^\top x_{\bar{\pi}(i)} + \varepsilon_i, \quad i \in [n] \]

- Covariate vectors: \((x_i)_{i=1}^n\) iid from \(\mathbb{P}\)
- Measurement errors: \((\varepsilon_i)\) iid from \(\mathcal{N}(0, \sigma^2)\)
- Unknown linear function: \(\bar{w} \in \mathbb{R}^d\)
- Unknown permutation: \(\bar{\pi} \in S_n\)

**Equivalent:** ignore \(\bar{\pi}\); observe \((x_i)_{i=1}^n\) and \(\mathcal{L}(y_i)_{i=1}^n\)

(where \(\mathcal{L} \cdot \mathcal{S}\) denotes *unordered multi-set*).
Linear model with Gaussian noise

\[ y_i = \bar{w}^\top x_{\bar{\pi}(i)} + \varepsilon_i, \quad i \in [n] \]

- Covariate vectors: \((x_i)_{i=1}^n\) iid from \(\mathbb{P}\)
- Measurement errors: \((\varepsilon_i)\) iid from \(\mathcal{N}(0, \sigma^2)\)
- Unknown linear function: \(\bar{w} \in \mathbb{R}^d\)
- Unknown permutation: \(\bar{\pi} \in S_n\)

**Equivalent:** ignore \(\bar{\pi}\); observe \((x_i)_{i=1}^n\) and \(\{y_i\}_{i=1}^n\)
(where \(\cdot\) denotes unordered multi-set).

We consider \(\mathbb{P} = \mathcal{N}(0, I_d)\) and \(\mathbb{P} = \text{Uniform}([-1, 1]^d)\).  

Note: If correspondence between \((x_i)_{i=1}^n\) and \(\{y_i\}_{i=1}^n\) is known, then just need \(\text{SNR} \gtrsim d/n\) to approximately recover \(\bar{w}\).
**Setting**

**Linear model with Gaussian noise**

\[
y_i = \bar{w}^\top x_{\bar{\pi}(i)} + \varepsilon_i, \quad i \in [n]
\]

- Covariate vectors: \((x_i)_{i=1}^n\) iid from \(\mathbb{P}\)
- Measurement errors: \((\varepsilon_i)\) iid from \(\mathcal{N}(0, \sigma^2)\)
- Unknown linear function: \(\bar{w} \in \mathbb{R}^d\)
- Unknown permutation: \(\bar{\pi} \in S_n\)

**Equivalent:** ignore \(\bar{\pi}\); observe \((x_i)_{i=1}^n\) and \(\{y_i\}_{i=1}^n\)
(where \(\cdot\) denotes *unordered multi-set*).

We consider \(\mathbb{P} = \mathcal{N}(0, I_d)\) and \(\mathbb{P} = \text{Uniform}([-1, 1]^d)\).

**Note:** If correspondence between \((x_i)_{i=1}^n\) and \(\{y_i\}_{i=1}^n\) is *known*, then just need \(\text{SNR} \gtrsim d/n\) to approximately recover \(\bar{w}\).
Uniform case

Theorem

If \((x_i)_{i=1}^n\) are iid draws from \(\text{Uniform}([-1, 1]^d)\), \((y_i)_{i=1}^n\) follow the linear model with \(N(0, \sigma^2)\) noise, and

\[\text{SNR} \leq (1 - 2c)^2\]

for some \(c \in (0, 1/2)\), then for any estimator \(\hat{w}\), there exists \(\bar{w} \in \mathbb{R}^d\) such that

\[\mathbb{E} \left[ \|\hat{w} - \bar{w}\|_2 \right] \geq c\|\bar{w}\|_2\]
Uniform case

**Theorem**

If \((x_i)_{i=1}^n\) are iid draws from Uniform\([-1, 1]^d\), \((y_i)_{i=1}^n\) follow the linear model with \(N(0, \sigma^2)\) noise, and

\[
\text{SNR} \leq (1 - 2c)^2
\]

for some \(c \in (0, \frac{1}{2})\), then for any estimator \(\hat{w}\), there exists \(\bar{w} \in \mathbb{R}^d\) such that

\[
E \left[ \|\hat{w} - \bar{w}\|_2 \right] \geq c\|\bar{w}\|_2.
\]

**Increasing sample size** \(n\) **does not help**, unlike in the “known correspondence” setting (where \(\text{SNR} \gtrsim \frac{d}{n}\) suffices).
Proof sketch

We show that no estimator can confidently distinguish between $\bar{w} = e_1$ and $\bar{w} = -e_1$, where $e_1 = (1, 0, \ldots, 0)^\top$. 
Proof sketch

We show that no estimator can confidently distinguish between $\bar{w} = e_1$ and $\bar{w} = -e_1$, where $e_1 = (1, 0, \ldots, 0)^\top$.

Let $P_{\bar{w}}$ be the data distribution with parameter $\bar{w} \in \{e_1, -e_1\}$.

**Task**: show $P_{e_1}$ and $P_{-e_1}$ are “close”, then appeal to Le Cam’s standard “two-point argument”:

$$\max_{\bar{w} \in \{e_1, -e_1\}} \mathbb{E}_{P_{\bar{w}}} \|\hat{w} - \bar{w}\|_2 \geq 1 - \|P_{e_1} - P_{-e_1}\|_{tv}.$$
Proof sketch

We show that no estimator can confidently distinguish between \( \bar{w} = e_1 \)
and \( \bar{w} = -e_1 \), where \( e_1 = (1, 0, \ldots, 0)^\top \).

Let \( P_{\bar{w}} \) be the data distribution with parameter \( \bar{w} \in \{e_1, -e_1\} \).

**Task:** show \( P_{e_1} \) and \( P_{-e_1} \) are “close”, then appeal to Le Cam’s standard “two-point argument”:

\[
\max_{\bar{w} \in \{e_1, -e_1\}} \mathbb{E}_{P_{\bar{w}}} \| \hat{w} - \bar{w} \|_2 \geq 1 - \| P_{e_1} - P_{-e_1} \|_{tv}.
\]

**Key idea:** conditional means of \( \langle y_i \rangle_{i=1}^n \) given \( \langle x_i \rangle_{i=1}^n \), under \( P_{e_1} \) and \( P_{-e_1} \), are close as unordered multi-sets.
Proof sketch (continued)

Generative process for $P_{\overline{w}}$:

1. Draw $(x_i)_{i=1}^{n} \sim \text{Uniform}([-1, 1]^d)$,
   $(\varepsilon_i)_{i=1}^{n} \sim \mathcal{N}(0, \sigma^2)$.

2. Set $u_i := \overline{w}^\top x_i$ for $i \in [n]$.

3. Set $y_i := u_i(\varepsilon_i)$ for $i \in [n]$, where $u_1 \leq u_2 \leq \cdots \leq u_n$.

Conditional distribution of $y = (y_1, y_2, \ldots, y_n)$ given $(x_i)_{i=1}^{n}$:

Under $P_{e_1}$:

$y | (x_i)_{i=1}^{n} \sim \mathcal{N}(u_1^{-1}, \sigma^2 I_n)$

Under $P_{-e_1}$:

$y | (x_i)_{i=1}^{n} \sim \mathcal{N}(u_n^{-1}, \sigma^2 I_n)$

where $u_1^{-1} = (u_1(1), u_2(2), \ldots, u_n(n))$ and $u_n^{-1} = (u_n(n), u_{n-1}(n-1), \ldots, u_1(1))$.

Data processing: Lose information by going from $y$ to $H_y$.
Proof sketch (continued)

Generative process for $P_w$:

1. Draw $(\boldsymbol{x}_i)_{i=1}^n \overset{iid}{\sim} \text{Uniform}([-1, 1]^d)$, $(\varepsilon_i)_{i=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$. 

Data processing: Lose information by going from $y$ to $H_{\text{fix}}$. 

30
Proof sketch (continued)

Generative process for $P_w$:

1. Draw $(\mathbf{x}_i)_{i=1}^{n} \overset{iid}{\sim} \text{Uniform}([-1, 1]^d)$, $(\varepsilon_i)_{i=1}^{n} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$.
2. Set $u_i := \mathbf{w}^\top \mathbf{x}_i$ for $i \in [n]$. 
Proof sketch (continued)

Generative process for $P_{\bar{w}}$:

1. Draw $(\mathbf{x}_i)_{i=1}^n \overset{iid}{\sim} \text{Uniform}([-1, 1]^d)$, $(\varepsilon_i)_{i=1}^n \overset{iid}{\sim} \text{N}(0, \sigma^2)$.

2. Set $u_i := \bar{w}^\top \mathbf{x}_i$ for $i \in [n]$.

3. Set $y_i := u_i + \varepsilon_i$ for $i \in [n]$, where $u(1) \leq u(2) \leq \cdots \leq u(n)$. 
Proof sketch (continued)

Generative process for $P_{\bar{w}}$:

1. Draw $(x_i)_{i=1}^n \overset{iid}{\sim} \text{Uniform}([-1, 1]^d)$, $(\varepsilon_i)_{i=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$.
2. Set $u_i := \bar{w}^\top x_i$ for $i \in [n]$.
3. Set $y_i := u(i) + \varepsilon_i$ for $i \in [n]$, where $u(1) \leq u(2) \leq \cdots \leq u(n)$.

Conditional distribution of $y = (y_1, y_2, \ldots, y_n)$ given $(x_i)_{i=1}^n$:

Under $P_{e_1}$: $y \mid (x_i)_{i=1}^n \sim \mathcal{N}(u^\uparrow, \sigma^2 I_n)$

Under $P_{-e_1}$: $y \mid (x_i)_{i=1}^n \sim \mathcal{N}(-u^\downarrow, \sigma^2 I_n)$

where $u^\uparrow = (u(1), u(2), \ldots, u(n))$ and $u^\downarrow = (u(n), u(n-1), \ldots, u(1))$. 
Proof sketch (continued)

Generative process for $P_{\bar{w}}$:

1. Draw $(x_i)_{i=1}^n \overset{iid}{\sim} \text{Uniform}([-1, 1]^d)$, $(\varepsilon_i)_{i=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$.
2. Set $u_i := \bar{w}^\top x_i$ for $i \in [n]$.
3. Set $y_i := u(i) + \varepsilon_i$ for $i \in [n]$, where $u(1) \leq u(2) \leq \cdots \leq u(n)$.

Conditional distribution of $y = (y_1, y_2, \ldots, y_n)$ given $(x_i)_{i=1}^n$:

Under $P_{e_1}$: $y \mid (x_i)_{i=1}^n \sim \mathcal{N}(u^\uparrow, \sigma^2 I_n)$
Under $P_{-e_1}$: $y \mid (x_i)_{i=1}^n \sim \mathcal{N}(-u^\downarrow, \sigma^2 I_n)$

where $u^\uparrow = (u(1), u(2), \ldots, u(n))$ and $u^\downarrow = (u(n), u(n-1), \ldots, u(1))$.

Data processing: Lose information by going from $y$ to $(y_i)_{i=1}^n$. 
Proof sketch (continued)

By data processing inequality,

\[
\text{KL} \left( P_{e_1} \left( \cdot \ \mid (x_i)_{i=1}^n \right), P_{-e_1} \left( \cdot \ \mid (x_i)_{i=1}^n \right) \right) \\
\leq \text{KL} \left( N(u^\uparrow, \sigma^2 I_n), N(-u^\downarrow, \sigma^2 I_n) \right)
\]
Proof sketch (continued)

By data processing inequality,

\[
\begin{align*}
&\quad \text{KL} \left( P_{e_1} (\cdot \mid (x_i)_{i=1}^n), P_{-e_1} (\cdot \mid (x_i)_{i=1}^n) \right) \\
&\leq \text{KL} \left( \mathcal{N}(u^\uparrow, \sigma^2 I_n), \mathcal{N}(-u^\downarrow, \sigma^2 I_n) \right) \\
&= \frac{\|u^\uparrow - (-u^\downarrow)\|_2^2}{2\sigma^2}
\end{align*}
\]
Proof sketch (continued)

By data processing inequality,

\[
\text{KL} \left( P_{e_1} (\cdot | (x_i)_{i=1}^n), P_{-e_1} (\cdot | (x_i)_{i=1}^n) \right) \\
\leq \text{KL} \left( N(u^\uparrow, \sigma^2 I_n), N(-u^\downarrow, \sigma^2 I_n) \right) \\
= \frac{\|u^\uparrow - (-u^\downarrow)\|^2}{2\sigma^2} = \frac{\text{SNR}}{2} \cdot \|u^\uparrow + u^\downarrow\|^2_2.
\]
Proof sketch (continued)

By data processing inequality,

\[
\text{KL} \left( P_{e_1} (\cdot \mid (x_i)_{i=1}^n), P_{-e_1} (\cdot \mid (x_i)_{i=1}^n) \right) \\
\leq \text{KL} \left( N(u^\uparrow, \sigma^2 I_n), N(-u^\downarrow, \sigma^2 I_n) \right) \\
= \frac{\| u^\uparrow - (-u^\downarrow) \|_2^2}{2\sigma^2} = \frac{\text{SNR}}{2} \cdot \| u^\uparrow + u^\downarrow \|_2^2.
\]

Some computations show that

\[
\text{med} \| u^\uparrow + u^\downarrow \|_2^2 \leq 4.
\]
Proof sketch (continued)

By data processing inequality,

\[
\text{KL} \left( P_{e_1} \cdot | (x_i)_{i=1}^n, P_{-e_1} \cdot | (x_i)_{i=1}^n \right) \\
\leq \text{KL} \left( N(u^\uparrow, \sigma^2 I_n), N(-u^\downarrow, \sigma^2 I_n) \right) \\
= \frac{\| u^\uparrow - (-u^\downarrow) \|^2_2}{2\sigma^2} = \frac{\text{SNR}}{2} \cdot \| u^\uparrow + u^\downarrow \|^2_2.
\]

Some computations show that

\[
\text{med} \| u^\uparrow + u^\downarrow \|^2_2 \leq 4.
\]

By conditioning + Pinsker’s inequality,

\[
\| P_{e_1} - P_{-e_1} \|_{tv} \leq \frac{1}{2} + \frac{1}{2} \text{med} \sqrt{\frac{\text{SNR}}{4} \cdot \| u^\uparrow + u^\downarrow \|^2_2}
\]
Proof sketch (continued)

By data processing inequality,

\[
\begin{align*}
\text{KL} \left( P_{e_1} (\cdot \mid (x_i)_{i=1}^n), P_{-e_1} (\cdot \mid (x_i)_{i=1}^n) \right) \\
\leq \text{KL} \left( \mathcal{N}(u^\uparrow, \sigma^2 I_n), \mathcal{N}(-u^\downarrow, \sigma^2 I_n) \right) \\
= \frac{\|u^\uparrow - (-u^\downarrow)\|_2^2}{2\sigma^2} = \frac{\text{SNR}}{2} \cdot \|u^\uparrow + u^\downarrow\|_2^2.
\end{align*}
\]

Some computations show that

\[
\text{med} \|u^\uparrow + u^\downarrow\|_2^2 \leq 4.
\]

By conditioning + Pinsker’s inequality,

\[
\|P_{e_1} - P_{-e_1}\|_{tv} \leq \frac{1}{2} + \frac{1}{2} \text{med} \sqrt{\frac{\text{SNR}}{4} \cdot \|u^\uparrow + u^\downarrow\|_2^2}
\leq \frac{1}{2} + \frac{1}{2} \sqrt{\text{SNR}}.
\]
Theorem

If \((x_i)_{i=1}^n\) are iid draws from \(N(0, I_d)\), \((y_i)_{i=1}^n\) follow the linear model with \(N(0, \sigma^2)\) noise, and

\[
\text{SNR} \leq C \cdot \frac{d}{\log \log(n)}
\]

for some absolute constant \(C' > 0\), then for any estimator \(\hat{w}\), there exists \(\bar{w} \in \mathbb{R}^d\) such that

\[
\mathbb{E} \left[ \|\hat{w} - \bar{w}\|_2 \right] \geq C' \|\bar{w}\|_2
\]

for some other absolute constant \(C' > 0\).
Gaussian case

**Theorem**

If \((x_i)_{i=1}^n\) are iid draws from \(N(0, I_d)\), \((y_i)_{i=1}^n\) follow the linear model with \(N(0, \sigma^2)\) noise, and

\[
\text{SNR} \leq C \cdot \frac{d}{\log \log(n)}
\]

for some absolute constant \(C > 0\), then for any estimator \(\hat{w}\), there exists \(\bar{w} \in \mathbb{R}^d\) such that

\[
\mathbb{E} \left[ \|\hat{w} - \bar{w}\|_2 \right] \geq C' \|\bar{w}\|_2
\]

for some other absolute constant \(C' > 0\).

C.f. “known correspondence” setting, where SNR \(\gtrsim d/n\) suffices.
4. Closing remarks and open problems
Closing remarks and open problems

Lack of correspondence changes both computational and statistical difficulty of linear regression.
Closing remarks and open problems

Lack of correspondence changes both computational and statistical difficulty of linear regression.

- Algorithms shed light on computational difficulty in worst-case and average-case settings.

- SNR lower bounds show striking contrast to "known correspondence" settings.

- Gap remains between SNR lower and upper bounds.

- E.g., $N(0, I_d)$ case (with $d$ constant):

- Is MLE (near) optimal for recovering $\bar{w}$?

- $N(0, I_d)$ vs Uniform([-1, 1]^d)?

- Faster algorithms? (Smoothed) analysis of alternating minimization?
Closing remarks and open problems

Lack of correspondence changes both computational and statistical difficulty of linear regression.

- Algorithms shed light on computational difficulty in worst-case and average-case settings.

- SNR lower bounds show striking contrast to “known correspondence” settings.

\[ N(0, I_d) \text{ case (with } d \text{ constant):} \]

- Is MLE (near) optimal for recovering \( \bar{w} \)?

\[ N(0, I_d) \text{ vs Uniform([} -1, 1] d) \]

- Faster algorithms? (Smoothed) analysis of alternating minimization?
Closing remarks and open problems

Lack of correspondence changes both computational and statistical difficulty of linear regression.

- Algorithms shed light on computational difficulty in worst-case and average-case settings.

- SNR lower bounds show striking contrast to “known correspondence” settings.

- Gap remains between SNR lower and upper bounds.

  E.g., $\mathcal{N}(0, I_d)$ case (with $d$ constant):

  $O\left(\frac{1}{\log \log n}\right)$: fails $\quad \rightarrow \quad \Omega(n^c)$: succeeds [PWC’16]

  - Is MLE (near) optimal for recovering $\bar{w}$?
  - $\mathcal{N}(0, I_d)$ vs Uniform$([-1, 1]^d)$?
Closing remarks and open problems

Lack of correspondence changes both computational and statistical difficulty of linear regression.

- Algorithms shed light on computational difficulty in worst-case and average-case settings.

- SNR lower bounds show striking contrast to “known correspondence” settings.

- Gap remains between SNR lower and upper bounds.

  E.g., $\mathcal{N}(0, I_d)$ case (with $d$ constant):

  $$O\left(\frac{d}{\log \log n}\right): \text{fails} \quad \rightarrow \quad \omega(1): \text{succeeds (new!)}$$

- Is MLE (near) optimal for recovering $\bar{w}$?

- $\mathcal{N}(0, I_d)$ vs Uniform($[-1, 1]^d$)?
Closing remarks and open problems

Lack of correspondence changes both computational and statistical difficulty of linear regression.

- Algorithms shed light on computational difficulty in worst-case and average-case settings.

- SNR lower bounds show striking contrast to “known correspondence” settings.

- Gap remains between SNR lower and upper bounds.
  E.g., $\mathcal{N}(0, I_d)$ case (with $d$ constant):
  \[
  O\left(\frac{d}{\log \log n}\right): \text{fails} \quad \leftrightarrow \quad \omega(1): \text{succeeds (new!)}
  \]

  - Is MLE (near) optimal for recovering $\bar{w}$?
  - $\mathcal{N}(0, I_d)$ vs Uniform([−1, 1]$^d$)?

- Faster algorithms?
  (Smoothed) analysis of alternating minimization?
Acknowledgements

Collaborators: Kevin Shi (Columbia University), Xiaorui Sun (Microsoft Research).

Discussants: Ashwin Pananjady (UCB), Michał Dereziński (UCSC), Manfred Warmuth (UCSC).

Funding: NSF (DMR-1534910, IIS-1563785), Sloan Research Fellowship, Bloomberg Data Science Research Grant.


See preprint for details & references: arxiv.org/abs/1705.07048

Thank you