Learning without correspondence

Daniel Hsu

Computer Science Department & Data Science Institute
Columbia University
Introduction
Example #1: unlinked data sources

- Two separate data sources about same entities:

<table>
<thead>
<tr>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>20</td>
<td>180</td>
</tr>
<tr>
<td>F</td>
<td>24</td>
<td>162.5</td>
</tr>
<tr>
<td>F</td>
<td>22</td>
<td>160</td>
</tr>
<tr>
<td>F</td>
<td>23</td>
<td>167.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

- First source contains covariates (sex, age, height, ...).
- Second source contains response variable (disease status).
Example #1: unlinked data sources

- Two separate data sources about same entities:

<table>
<thead>
<tr>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>20</td>
<td>180</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>24</td>
<td>162.5</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>22</td>
<td>160</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>23</td>
<td>167.5</td>
<td>1</td>
</tr>
</tbody>
</table>

- First source contains covariates (sex, age, height, ...).
- Second source contains response variable (disease status).

**To learn:** relationship between response and covariates.

*Record linkage unknown.*
Example #2: flow cytometry

1. Suspended cells in fluid.
2. Cells pass through laser, one at a time; measure emitted light.
Example #2: flow cytometry

1. Suspended cells in fluid.
2. Cells pass through laser, one at a time; measure emitted light.

To learn: relationship between measurements and cell properties.

Order in which cells pass through laser is unknown.
Example #3: unassigned distance geometry

1. Unknown arrangement of $n$ points in Euclidean space.

![Diagram](image)

(Image credit: Billinge, Duxbury, Gonçalves, Lavor, & Mucherino, 2016)

2. Measure distribution of *pairwise distances* among the $n$ points (using high-energy X-rays).
Example #3: unassigned distance geometry

1. Unknown arrangement of $n$ points in Euclidean space.

2. Measure distribution of pairwise distances among the $n$ points (using high-energy X-rays).

To learn: original arrangement of the $n$ points.

Assignment of distances to pairs of points is unknown.
Learning without correspondence

Observation:
Correspondence information is missing in many natural settings.
Observation:
Correspondence information is missing in many natural settings.

Question:
How does this affect machine learning / statistical estimation?
Learning without correspondence

Observation:
Correspondence information is missing in many natural settings.

Question:
How does this affect machine learning / statistical estimation?

We give a theoretical treatment in context of two simple problems:

1. Linear regression without correspondence
   (Joint work with Kevin Shi and Xiaorui Sun; NIPS 2017.)

2. Correspondence retrieval (generalization of phase retrieval)
   (Joint work with Alexandr Andoni, Kevin Shi, and Xiaorui Sun; COLT 2017.)
Our contributions

1. **Linear regression without correspondence**
   - Strong NP-hardness of least squares problem.
   - Polynomial-time approximation scheme in constant dimensions.
   - Information-theoretic signal-to-noise lower bounds.
   - Polynomial-time algorithm in noise-free average case setting.

2. **Correspondence retrieval**
   - Measurement-optimal recovery algorithm in noise-free setting.
   - Robust recovery algorithm in noisy setting.
Linear regression without correspondence
### Linear regression without correspondence

#### Feature vectors:
\[ x_1, x_2, \ldots, x_n \in \mathbb{R}^d \]

#### Labels:
\[ y_1, y_2, \ldots, y_n \in \mathbb{R} \]
Linear regression without correspondence

Classical linear regression:

\[ y_i = x_i^T \beta^* + \varepsilon_i, \quad i = 1, \ldots, n. \]
Linear regression without correspondence:

\[ y_i = \mathbf{x}_{\pi^*(i)}^\top \beta^* + \varepsilon_i, \quad i = 1, \ldots, n. \]
Model for linear regression without correspondence

Unnikrishnan, Haghighatshoar, & Vetterli, 2015; Pananjady, Wainwright, & Courtade 2016; Elhami, Scholefield, Haro, & Vetterli, 2017; Abid, Poon, & Zou, 2017; ...

- **Feature vectors**: $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$
- **Labels**: $y_1, y_2, \ldots, y_n \in \mathbb{R}$
- **Model**:
  \[
  y_i = x_{\pi^*(i)}^\top \beta^* + \varepsilon_i, \quad i = 1, \ldots, n.
  \]
  - Linear function: $\beta^* \in \mathbb{R}^d$
  - Permutation: $\pi^* \in S_n$
  - Errors: $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in \mathbb{R}$. 

Goal: "learn" $\beta^*$.

Correspondence between $(x_i)_{i=1}^n$ and $(y_i)_{i=1}^n$ is unknown.
Model for linear regression without correspondence

Unnikrishnan, Haghighatshoar, & Vetterli, 2015; Pananjady, Wainwright, & Courtade 2016; Elhami, Scholefield, Haro, & Vetterli, 2017; Abid, Poon, & Zou, 2017; …

- **Feature vectors**: \( x_1, x_2, \ldots, x_n \in \mathbb{R}^d \)
- **Labels**: \( y_1, y_2, \ldots, y_n \in \mathbb{R} \)
- **Model**:
  \[
  y_i = x_{\pi^*(i)}^T \beta^* + \varepsilon_i, \quad i = 1, \ldots, n.
  \]
  - Linear function: \( \beta^* \in \mathbb{R}^d \)
  - Permutation: \( \pi^* \in S_n \)
  - Errors: \( \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in \mathbb{R} \).
- **Goal**: “learn” \( \beta^* \).
Model for linear regression without correspondence

Unnikrishnan, Haghighatshoar, & Vetterli, 2015; Pananjady, Wainwright, & Courtade 2016; Elhami, Scholefield, Haro, & Vetterli, 2017; Abid, Poon, & Zou, 2017; ...

- **Feature vectors**: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$
- **Labels**: $y_1, y_2, \ldots, y_n \in \mathbb{R}$
- **Model**: 
  
  $$y_i = \mathbf{x}_{\pi^*(i)}^\top \beta^* + \varepsilon_i, \quad i = 1, \ldots, n.$$ 

  - Linear function: $\beta^* \in \mathbb{R}^d$
  - Permutation: $\pi^* \in S_n$
  - Errors: $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n \in \mathbb{R}$.

- **Goal**: “learn” $\beta^*$.

Correspondence between $(\mathbf{x}_i)_{i=1}^n$ and $(y_i)_{i=1}^n$ is unknown.
1. Can we determine if there is a good linear fit to the data? (Least squares approximation.)
1. Can we determine if there is a good linear fit to the data? (Least squares approximation.)
2. When is it possible to recover the “correct” $\beta^*$? (When is the “best” linear fit actually meaningful?)
Least squares approximation
Least squares problem

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

\[
F(\beta, \pi) := \sum_{i=1}^{n} \left( x_i \beta - y_{\pi(i)} \right)^2.
\]
Least squares problem

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

\[
F(\beta, \pi) := \sum_{i=1}^n \left( x_i^\top \beta - y_{\pi(i)} \right)^2.
\]

- \(d = 1\): \(O(n \log n)\)-time algorithm.
  
  (Observed by Pananjady, Wainwright, & Courtade, 2016.)
Least squares problem

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

\[
F(\beta, \pi) := \sum_{i=1}^n (x_i^\top \beta - y_{\pi(i)})^2.
\]

- \(d = 1\): \(O(n \log n)\)-time algorithm.
  (Observed by Pananjady, Wainwright, & Courtade, 2016.)

- \(d = \Omega(n)\): (strongly) NP-hard to decide if \(\min F = 0\).
  Reduction from 3-PARTITION (H., Shi, & Sun, 2017).
Least squares problem

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

\[
F(\beta, \pi) := \sum_{i=1}^{n} \left( x_i^\top \beta - y_{\pi(i)} \right)^2.
\]

- \(d = 1\): \(O(n \log n)\)-time algorithm.
  (Observed by Pananjady, Wainwright, & Courtade, 2016.)

- \(d = \Omega(n)\): (strongly) NP-hard to decide if \(\min F = 0\).
  Reduction from 3-PARTITION (H., Shi, & Sun, 2017).

Naïve brute-force search: \(\Omega(|S_n|) = \Omega(n!)\).
Least squares problem

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

\[
F(\beta, \pi) := \sum_{i=1}^{n} \left( x_i^\top \beta - y_{\pi(i)} \right)^2.
\]

- \(d = 1\): \(O(n \log n)\)-time algorithm.
  (Observed by Pananjady, Wainwright, & Courtade, 2016.)

- \(d = \Omega(n)\): (strongly) NP-hard to decide if \(\min F = 0\).
  Reduction from 3-PARTITION (H., Shi, & Sun, 2017).

Naïve brute-force search: \(\Omega(|S_n|) = \Omega(n!)\).

Least squares with known correspondence: \(O(nd^2)\) time.
Given \((x_i)_{i=1}^{n}\) and \((y_i)_{i=1}^{n}\) from \(\mathbb{R}\), minimize

\[
F(\beta, \pi) := \sum_{i=1}^{n} \left( x_i \beta - y_{\pi(i)} \right)^2.
\]

<table>
<thead>
<tr>
<th>(x_1)</th>
<th>(y_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_2)</td>
<td>(y_2)</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
</tr>
<tr>
<td>(x_n)</td>
<td>(y_n)</td>
</tr>
</tbody>
</table>
Least squares problem \((d = 1)\)

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

\[
F(\beta, \pi) := \sum_{i=1}^n (x_i \beta - y_{\pi(i)})^2.
\]

\[
\begin{align*}
(x_1 \beta - y_1)^2 \\
(x_2 \beta - y_2)^2 \\
\vdots \\
(x_n \beta - y_n)^2
\end{align*}
\]

Cost with \(\pi(i) = i\) for all \(i = 1, \ldots, n\).
Least squares problem \((d = 1)\)

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

\[
F(\beta, \pi) := \sum_{i=1}^n \left( x_i \beta - y_{\pi(i)} \right)^2.
\]

Cost with \(\pi(i) = i\) for all \(i = 1, \ldots, n\).
Least squares problem \((d = 1)\)

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

\[
F(\beta, \pi) := \sum_{i=1}^n \left( x_i \beta - y_{\pi(i)} \right)^2.
\]

If \(\beta > 0\), then can improve cost with \(\pi(1) = 2\) and \(\pi(2) = 1\).
Least squares problem \((d = 1)\)

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

\[
F(\beta, \pi) := \sum_{i=1}^n (x_i \beta - y_{\pi(i)})^2.
\]

If \(\beta > 0\), then can improve cost with \(\pi(1) = 2\) and \(\pi(2) = 1\).

\[
25\beta^2 - 20\beta + 5 + \cdots > 25\beta^2 - 22\beta + 5 + \cdots
\]
Algorithm for least squares problem ($d = 1$) [PWC’16]

1. “Guess” sign of optimal $\beta$. (Only two possibilities.)
Algorithm for least squares problem \((d = 1)\) [PWC’16]

1. “Guess” sign of optimal \(\beta\). (Only two possibilities.)
2. Assuming WLOG that \(x_1 \beta \leq x_2 \beta \leq \cdots \leq x_n \beta\),
   find optimal \(\pi\) such that \(y_{\pi(1)} \leq y_{\pi(2)} \leq \cdots \leq y_{\pi(n)}\)
   (via sorting).
3. Solve classical least squares problem
   \[
   \min_{\beta \in \mathbb{R}^n} \sum_{i=1}^{n} (x_i \beta - y_{\pi(i)})^2
   \]
   to get optimal \(\beta\).

Overall running time: \(O(n \log n)\).
Algorithm for least squares problem \((d = 1)\) [PWC’16]

1. “Guess” sign of optimal \(\beta\). (Only two possibilities.)

2. Assuming WLOG that \(x_1\beta \leq x_2\beta \cdots \leq x_n\beta\), find optimal \(\pi\) such that \(y_{\pi(1)} \leq y_{\pi(2)} \leq \cdots \leq y_{\pi(n)}\) (via sorting).

3. Solve classical least squares problem

\[
\min_{\beta \in \mathbb{R}} \sum_{i=1}^{n} (x_i\beta - y_{\pi(i)})^2
\]

...to get optimal \(\beta\).
Algorithm for least squares problem \((d = 1)\) [PWC’16]

1. “Guess” sign of optimal \(\beta\). (Only two possibilities.)

2. Assuming WLOG that \(x_1\beta \leq x_2\beta \cdots \leq x_n\beta\),
   find optimal \(\pi\) such that \(y_{\pi(1)} \leq y_{\pi(2)} \leq \cdots \leq y_{\pi(n)}\)
   (via sorting).

3. Solve classical least squares problem

\[
\min_{\beta \in \mathbb{R}} \sum_{i=1}^{n} (x_i\beta - y_{\pi(i)})^2
\]

   to get optimal \(\beta\).

**Overall running time:** \(O(n \log n)\).
Algorithm for least squares problem \((d = 1)\) [PWC’16]

1. “Guess” sign of optimal \(\beta\). (Only two possibilities.)

2. Assuming WLOG that \(x_1\beta \leq x_2\beta \cdots \leq x_n\beta\), find optimal \(\pi\) such that \(y_{\pi(1)} \leq y_{\pi(2)} \leq \cdots \leq y_{\pi(n)}\) (via sorting).

3. Solve classical least squares problem

\[
\min_{\beta \in \mathbb{R}} \sum_{i=1}^{n} (x_i\beta - y_{\pi(i)})^2
\]

to get optimal \(\beta\).

**Overall running time:** \(O(n \log n)\).

**What about \(d > 1\)?**
Alternating minimization

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

\[
\hat{\pi} \leftarrow \text{arg min}_{\pi \in S_n} \sum_{i=1}^{n} \left( \mathbf{x}_i^\top \hat{\beta} - y_{\pi(i)} \right)^2 .
\]

\[
\hat{\beta} \leftarrow \text{arg min}_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} \left( \mathbf{x}_i^\top \beta - y_{\hat{\pi}(i)} \right)^2 .
\]

• Each loop-iteration efficiently computable.
• But can get stuck in local minima.

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

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

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

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

- Each loop-iteration efficiently computable.
Alternating minimization

Pick initial $\hat{\beta} \in \mathbb{R}^d$ (e.g., randomly).

Loop until convergence:

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

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

- Each loop-iteration efficiently computable.
- But can get stuck in local minima.
Alternating minimization

- Each loop-iteration efficiently computable.
- But can get stuck in local minima.

(Image credit: Wolfram|Alpha)
Alternating minimization

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

(Open: How many restarts? How many iterations?)
Theorem (H., Shi, & Sun, 2017)

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

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

Theorem (H., Shi, & Sun, 2017)

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

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

Recall: Brute-force solution needs \(\Omega(n!)\) time.

(No other previous algorithm with approximation guarantee.)
Statistical recovery of $\beta^*$: algorithms and lower-bounds
When does the best fit model shed light on the “truth” ($\pi^* \& \beta^*$)?
Motivation

When does the best fit model shed light on the “truth” \((\pi^* \& \beta^*)\)?

**Approach:** Study question in context of statistical model for data.
Motivation

When does the best fit model shed light on the “truth” \((\pi^* \& \beta^*)\)?

**Approach**: Study question in context of statistical model for data.

1. Understand information-theoretic limits on recovering truth.
2. Natural “average-case” setting for algorithms.
Assume \((x_i)_{i=1}^n\) iid from \(P\) and \((\varepsilon_i)_{i=1}^n\) iid from \(N(0, \sigma^2)\).
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{\pi^*}^\top(1) \\ \mathbf{x}_{\pi^*}^\top(2) \\ \vdots \\ \mathbf{x}_{\pi^*}^\top(n) \end{bmatrix} \begin{bmatrix} \beta^* \\ \vdots \\ \beta^* \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Assume $$(\mathbf{x}_i)^n_{i=1} \text{ iid from } \mathbb{P}$$ and $$(\varepsilon_i)^n_{i=1} \text{ iid from } \mathcal{N}(0, \sigma^2)$$.

Recoverability of $$\beta^*$$ depends on signal-to-noise ratio:

$$\text{SNR} := \frac{\|\beta^*\|_2^2}{\sigma^2}.$$
Assume \((x_i)_{i=1}^n\) iid from \(P\) and \((\epsilon_i)_{i=1}^n\) iid from \(N(0, \sigma^2)\).

Recoverability of \(\beta^*\) depends on **signal-to-noise ratio**: 

\[
\text{SNR} := \frac{\|\beta^*\|_2^2}{\sigma^2}.
\]

**Classical setting (where \(\pi^*\) is known):**

Just need \(\text{SNR} \gtrsim d/n\) to approximately recover \(\beta^*\).
Suppose $\beta^*$ is either $e_1 = (1, 0, 0, \ldots, 0)$ or $e_2 = (0, 1, 0, \ldots, 0)$. $
abla \pi^*(1)$, $\nabla \pi^*(2)$, $\nabla \pi^*(n)$, $\beta^*$, $\varepsilon_1$, $\varepsilon_2$, $\varepsilon_n$.
High-level intuition

Suppose $\beta^*$ is either $e_1 = (1, 0, 0, \ldots, 0)$ or $e_2 = (0, 1, 0, \ldots, 0)$. 

\[
\begin{array}{cccc}
 y_1 & y_2 & \cdots & y_n \\
 x_{\pi^*}^\top(1) & x_{\pi^*}^\top(2) & \cdots & x_{\pi^*}^\top(n) \\
 \beta^* & + & \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_n
\end{array}
\]
Suppose $\beta^*$ is either $e_1 = (1, 0, 0, \ldots, 0)$ or $e_2 = (0, 1, 0, \ldots, 0)$.

$\pi^*$ known: distinguishability of $e_1$ and $e_2$ can improve with $n$. 
Suppose $\beta^*$ is either $e_1 = (1, 0, 0, \ldots, 0)$ or $e_2 = (0, 1, 0, \ldots, 0)$.

$\pi^*$ known: distinguishability of $e_1$ and $e_2$ can improve with $n$.

$\pi^*$ unknown: distinguishability is less clear.

$\mathcal{L} y_i \mathcal{S}_{i=1}^n = \begin{cases} \mathcal{L} x_{i,1} \mathcal{S}_{i=1}^n + N(0, \sigma^2) & \text{if } \beta^* = e_1, \\ \mathcal{L} x_{i,2} \mathcal{S}_{i=1}^n + N(0, \sigma^2) & \text{if } \beta^* = e_2. \end{cases}$

($\mathcal{L} \cdot \mathcal{S}$ denotes unordered multi-set.)
Effect of noise

Without noise \( (P = N(0, I_d)) \)

\[ \sum_{i=1}^{n} x_{i,1} \]

\[ \sum_{i=1}^{n} x_{i,2} \]
Effect of noise

Without noise \((\mathbb{P} = \mathcal{N}(0, I_d))\)

\[ \sum_{i=1}^{n} x_{i,1} \]

\[ \sum_{i=1}^{n} x_{i,2} \]

With noise

\[ ??? + \mathcal{N}(0, \sigma^2) \]
Theorem (H., Shi, & Sun, 2017)

For $\mathbb{P} = \mathcal{N}(0, I_d)$, no estimator $\hat{\beta}$ can guarantee

$$
\mathbb{E} \left[ \| \hat{\beta} - \beta^* \|_2 \right] \leq \frac{\| \beta^* \|_2}{3}
$$

unless

$$
\text{SNR} \geq C \cdot \frac{d}{\log \log(n)}.
$$
Lower bound on SNR

**Theorem (H., Shi, & Sun, 2017)**

For $\mathbb{P} = \mathcal{N}(0, I_d)$, no estimator $\hat{\beta}$ can guarantee

$$
\mathbb{E} \left[ \| \hat{\beta} - \beta^* \|_2 \right] \leq \frac{\| \beta^* \|_2}{3}
$$

unless

$$
\text{SNR} \geq C \cdot \frac{d}{\log \log(n)}.
$$

“Known correspondence” setting: $\text{SNR} \gtrsim d/n$ suffices.
Lower bound on SNR

Theorem (H., Shi, & Sun, 2017)

For $\mathbb{P} = \mathcal{N}(0, I_d)$, no estimator $\hat{\beta}$ can guarantee

$$\mathbb{E} \left[ \| \hat{\beta} - \beta^* \|_2 \right] \leq \frac{\| \beta^* \|_2}{3}$$

unless

$$\text{SNR} \geq C \cdot \frac{d}{\log \log(n)}.$$

“Known correspondence” setting: $\text{SNR} \gtrsim d/n$ suffices.

Another theorem: for $\mathbb{P} = \text{Uniform}([-1, 1]^d)$, must have $\text{SNR} \geq 1/9$, even as $n \to \infty$.  

High SNR regime

Previous works (Unnikrishnan, Haghighatshoar, & Vetterli, 2015; Pananjady, Wainwright, & Courtade, 2016):

If $\text{SNR} \gg \text{poly}(n)$, then can recover $\pi^*$ (and $\beta^*$, approximately) using Maximum Likelihood Estimation, i.e., least squares.

Related ($d = 1$): broken random sample (DeGroot and Goel, 1980).

Estimate sign of correlation between $x_i$ and $y_i$.

Have estimator for $\text{sign}(\beta^*)$ that is correct w.p. $1 - \tilde{O}(\text{SNR}^{-1/4})$.

Does high SNR also permit efficient algorithms? (Recall: our approximate MLE algorithm has running time $n\tilde{O}(d)$.)
High SNR regime

Previous works (Unnikrishnan, Haghhighatshoar, & Vetterli, 2015; Pananjady, Wainwright, & Courtade, 2016):

If $\text{SNR} \gg \text{poly}(n)$, then can recover $\pi^*$ (and $\beta^*$, approximately) using Maximum Likelihood Estimation, i.e., least squares.
High SNR regime

Previous works (Unnikrishnan, Haghighatshoar, & Vetterli, 2015; Pananjady, Wainwright, & Courtade, 2016):

If $\text{SNR} \gg \text{poly}(n)$, then can recover $\pi^*$ (and $\beta^*$, approximately) using Maximum Likelihood Estimation, i.e., least squares.

Related ($d = 1$): broken random sample (DeGroot and Goel, 1980).

Estimate sign of correlation between $x_i$ and $y_i$.

Have estimator for $\text{sign}(\beta^*)$ that is correct w.p. $1 - \tilde{O}(\text{SNR}^{-1/4})$. 
High SNR regime

**Previous works** (Unnikrishnan, Haghighatshoar, & Vetterli, 2015; Pananjady, Wainwright, & Courtade, 2016):

If $\text{SNR} \gg \text{poly}(n)$, then can recover $\pi^*$ (and $\beta^*$, approximately) using Maximum Likelihood Estimation, i.e., least squares.

**Related** ($d = 1$): broken random sample (DeGroot and Goel, 1980).
Estimate sign of correlation between $x_i$ and $y_i$.

Have estimator for $\text{sign}(\beta^*)$ that is correct w.p. $1 - \tilde{O}(\text{SNR}^{-1/4})$.

**Does high SNR also permit efficient algorithms?**

(Recall: our approximate MLE algorithm has running time $n^{O(d)}$.)
Average-case recovery with very high SNR
Noise-free setting \((\text{SNR} = \infty)\)

\[ \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{\pi^*(0)}^\top \\ x_{\pi^*(1)}^\top \\ \vdots \\ x_{\pi^*(n)}^\top \end{bmatrix} \beta^* \]

Assume \((x_i)_{i=0}^n\) iid from \(\mathcal{N}(0, I_d)\).
Noise-free setting \((\text{SNR} = \infty)\)

\[
\begin{bmatrix}
y_0 \\
y_1 \\
\vdots \\
y_n
\end{bmatrix}
= 
\begin{bmatrix}
x_0^\top \\
x_{\pi^*(1)}^\top \\
\vdots \\
x_{\pi^*(n)}^\top
\end{bmatrix}
\beta^*
\]

Assume \( (x_i)_{i=0}^n \) iid from \( \mathcal{N}(0, I_d) \).
Also assume \( \pi^*(0) = 0 \).
Assume \((x_i)_{i=0}^{n} \) iid from \(N(0, I_d)\).

Also assume \(\pi^*(0) = 0\).

If \(n + 1 \geq d\), then recovery of \(\pi^*\) gives exact recovery of \(\beta^*\) (a.s.).
Noise-free setting \( (\text{SNR} = \infty) \)

\[
\begin{align*}
\begin{array}{c}
  y_0 \\
  y_1 \\
  \vdots \\
  y_n
\end{array}
  & =
\begin{array}{c}
  \mathbf{x}_0^\top \\
  \mathbf{x}_{\pi^*(1)}^\top \\
  \vdots \\
  \mathbf{x}_{\pi^*(n)}^\top
\end{array}
  \begin{array}{c}
  \beta^* \\
  \pi^*(1) \\
  \vdots \\
  \pi^*(n)
\end{array}
\end{align*}
\]

Assume \( (\mathbf{x}_i)_{i=0}^n \) iid from \( \mathcal{N}(0, \mathbf{I}_d) \).

Also assume \( \pi^*(0) = 0 \).

If \( n + 1 \geq d \), then recovery of \( \pi^* \) gives exact recovery of \( \beta^* \) (a.s.).

We’ll assume \( n + 1 \geq d + 1 \) (i.e., \( n \geq d \)).
Noise-free setting ($\text{SNR} = \infty$)

Assume $(x_i)_{i=0}^n$ iid from $\mathcal{N}(0, I_d)$.
Also assume $\pi^*(0) = 0$.

If $n + 1 \geq d$, then recovery of $\pi^*$ gives exact recovery of $\beta^*$ (a.s.).

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

**Claim:** $n \geq d$ suffices to recover $\pi^*$ with high probability.
Theorem (H., Shi, & Sun, 2017)

In the noise-free setting, there is a \(\text{poly}(n, d)\)-time* algorithm that returns \(\pi^*\) and \(\beta^*\) with high probability.

*Assuming problem is appropriately discretized.
Main idea: hidden subset

Measurements:

\[ y_0 = x_0^\top \beta^* ; \quad y_i = x_{\pi^*(i)}^\top \beta^* , \quad i = 1, \ldots, n . \]
Main idea: hidden subset

Measurements:

\[ y_0 = x_0^\top \beta^*; \quad y_i = x_{\pi^*(i)}^\top \beta^*, \quad i = 1, \ldots, n. \]

For simplicity: assume \( n = d \), and \( x_i = e_i \) for \( i = 1, \ldots, d \), so

\[ \{y_1, \ldots, y_d\} = \{\beta_1^*, \ldots, \beta_d^*\}. \]
Main idea: hidden subset

Measurements:

\[ y_0 = x_0^\top \beta^* ; \quad y_i = x_{\pi^*(i)}^\top \beta^* , \quad i = 1, \ldots, n . \]

For simplicity: assume \( n = d \), and \( x_i = e_i \) for \( i = 1, \ldots, d \), so

\[ \{ y_1, \ldots, y_d \} = \{ \beta_1^*, \ldots, \beta_d^* \} . \]

We also know:

\[ y_0 = x_0^\top \beta^* = \sum_{j=1}^{d} x_{0,j} \beta_j^* . \]
\[ y_0 = \mathbf{x}_0^\top \mathbf{\beta}^* = \sum_{j=1}^{d} x_{0,j} \beta_j^* \]

\[ = \sum_{i=1}^{d} \sum_{j=1}^{d} x_{0,j} y_i \cdot 1\{\pi^*(i) = j\} \]

\[
\begin{array}{c|c|c}
\hline
x_{0,1} & y_1 \\
\hline
x_{0,2} & y_2 \\
\vdots & \vdots \\
x_{0,n} & y_n \\
\hline
\end{array}
\]
\[ y_0 = x_0^\top \beta^* = \sum_{j=1}^{d} x_{0,j} \beta_j^* \]

\[ = \sum_{i=1}^{d} \sum_{j=1}^{d} x_{0,j} y_i \cdot 1\{\pi^*(i) = j\} \]
Reduction to Subset Sum

\[ y_0 = x_0^\top \beta^* = \sum_{j=1}^{d} x_{0,j} \beta_j^* \]

\[ = \sum_{i=1}^{d} \sum_{j=1}^{d} x_{0,j} y_i \cdot 1\{\pi^*(i) = j\} \]

- \(d^2\) “source” numbers \(c_{i,j} := x_{0,j} y_i\), “target” sum \(y_0\).

The subset \(\{c_{i,j} : \pi^*(i) = j\}\) adds up to \(y_0\).
Reduction to Subset Sum

\[ y_0 = x_0^\top \beta^* = \sum_{j=1}^{d} x_{0,j} \beta_j^* \]

\[ = \sum_{i=1}^{d} \sum_{j=1}^{d} x_{0,j} y_i \cdot 1\{\pi^*(i) = j\} \]

- \(d^2\) “source” numbers \(c_{i,j} := x_{0,j} y_i\), “target” sum \(y_0\).

The subset \(\{c_{i,j} : \pi^*(i) = j\}\) adds up to \(y_0\).

Subset Sum problem.
NP-Completeness of Subset Sum (a.k.a. "Knapsack")

Richard M. Karp

University of California at Berkeley

18. KNAPSACK

INPUT: \((a_1, a_2, \ldots, a_r, b) \in \mathbb{Z}^{n+1}\)

PROPERTY: \(\sum a_j x_j = b\) has a 0-1 solution.

(Karp, 1972)
Easiness of Subset Sum

- But Subset Sum is only “weakly” NP-hard (efficient algorithm exists for unary-encoded inputs).

- Lagarias & Odlyzko (1983): solving certain random instances can be reduced to solving Approximate Shortest Vector Problem in lattices.

- Lenstra, Lenstra, & Lovász (1982): efficient algorithm to solve Approximate SVP.

- Our algorithm is based on similar reduction but requires a somewhat different analysis.
But Subset Sum is only “weakly” NP-hard (efficient algorithm exists for unary-encoded inputs).

Lagarias & Odlyzko (1983): solving certain random instances can be reduced to solving Approximate Shortest Vector Problem in lattices.
But Subset Sum is only “weakly” NP-hard (efficient algorithm exists for unary-encoded inputs).

Lagarias & Odlyzko (1983): solving certain random instances can be reduced to solving Approximate Shortest Vector Problem in lattices.

Lenstra, Lenstra, & Lovász (1982): efficient algorithm to solve Approximate SVP.
But Subset Sum is only “weakly” NP-hard (efficient algorithm exists for unary-encoded inputs).

Lagarias & Odlyzko (1983): solving certain random instances can be reduced to solving Approximate Shortest Vector Problem in lattices.

Lenstra, Lenstra, & Lovász (1982): efficient algorithm to solve Approximate SVP.

Our algorithm is based on similar reduction but requires a somewhat different analysis.
Lagarias & Odlyzko (1983): random instances of Subset Sum efficiently solvable when $N$ source numbers $c_1, \ldots, c_N$ 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.

$\begin{bmatrix} b_0 & b_1 & \cdots & b_N \end{bmatrix} := \begin{bmatrix} 0 & I & -M & c_1 & \cdots & -M & c_N \end{bmatrix}$ for sufficiently large $M > 0$. 

Reducing subset sum to shortest vector problem

Lagarias & Odlyzko (1983): random instances of Subset Sum efficiently solvable when $N$ source numbers $c_1, \ldots, c_N$ 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 $c_1, \ldots, c_N$ 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 $v_\star$;
- any other lattice vector $\not\propto v_\star$ is more than $2^{N/2}$-times longer.

\[
\begin{bmatrix}
    b_0 & b_1 & \cdots & b_N \\
\end{bmatrix} := \begin{bmatrix}
    0 \\
    MT \\
    -Mc_1 \\
    \vdots \\
    -Mc_N
\end{bmatrix}
\]

for sufficiently large $M > 0$. 


Our random subset sum instance

**Catch:** Our source numbers $c_{i,j} = y_i \mathbf{x}_j^\top \mathbf{x}_0$ are **not independent**, and **not uniformly distributed** on some wide interval of $\mathbb{Z}$. 
Our random subset sum instance

**Catch:** Our source numbers $c_{i,j} = y_i \mathbf{x}_j \mathbf{x}_0$ 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 $c_{i,j} = y_i \mathbf{x}_j^\top \mathbf{x}_0$ 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, use Gaussian anti-concentration for quadratic and quartic forms.
**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, use 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 $\pi^*$,

$$\left| y_0 - \sum_{i,j} Z_{i,j} \cdot c_{i,j} \right| \geq \frac{1}{2^{\text{poly}(d)}} \cdot \| \beta^* \|_2 .$$
Some remarks

- In general, $x_1, \ldots, x_n$ are not $e_1, \ldots, e_d$, but similar reduction works via Moore-Penrose pseudoinverse.
Some remarks

- In general, \( x_1, \ldots, x_n \) are not \( e_1, \ldots, e_d \), but similar reduction works via Moore-Penrose pseudoinverse.

- Algorithm strongly exploits assumption of noise-free measurements. **Unlikely to tolerate much noise.**

**Open problem:**

*robust* efficient algorithm in high SNR setting.
Correspondence retrieval
Correspondence retrieval problem

**Goal:** recover $k$ unknown “signals” $\beta_1^*, \ldots, \beta_k^* \in \mathbb{R}^d$. 
Correspondence retrieval problem

**Goal:** recover \( k \) unknown “signals” \( \beta_1^*, \ldots, \beta_k^* \in \mathbb{R}^d \).

**Measurements:** \((x_i, Y_i)\) for \( i = 1, \ldots, n \), where

- \((x_i)\) iid from \( N(0, I_d) \);
- \(Y_i = \{x_i^\top \beta_1^* + \epsilon_{i,1}, \ldots, x_i^\top \beta_k^* + \epsilon_{i,k}\}\) as unordered multi-set;
- \((\epsilon_{i,j})\) iid from \( N(0, \sigma^2) \).

Correspondence across measurements is lost.
**Correspondence retrieval problem**

**Goal:** recover \( k \) unknown “signals” \( \beta_1^*, \ldots, \beta_k^* \in \mathbb{R}^d \).

**Measurements:** \((x_i, Y_i)\) for \( i = 1, \ldots, n \), where

- \((x_i)\) iid from \( \mathcal{N}(0, I_d) \);
- \( Y_i = \{ x_i^\top \beta_1^* + \varepsilon_{i,1}, \ldots, x_i^\top \beta_k^* + \varepsilon_{i,k} \} \) as unordered multi-set;
- \((\varepsilon_{i,j})\) iid from \( \mathcal{N}(0, \sigma^2) \).

Correspondence across measurements is lost.
Correspondence retrieval problem

**Goal**: recover $k$ unknown “signals” $\beta_1^*, \ldots, \beta_k^* \in \mathbb{R}^d$.

**Measurements**: $(x_i, Y_i)$ for $i = 1, \ldots, n$, where

- $(x_i)$ iid from $N(0, I_d)$;
- $Y_i = \{ x_i^\top \beta_1^* + \varepsilon_{i,1}, \ldots, x_i^\top \beta_k^* + \varepsilon_{i,k} \}$ as unordered multi-set;
- $(\varepsilon_{i,j})$ iid from $N(0, \sigma^2)$.

Correspondence across measurements is lost.
Special cases

- $k = 1$: classical linear regression model.
Special cases

- \( k = 1 \): classical linear regression regression model.
- \( k = 2 \) and \( \beta_1^* = -\beta_2^* \): (real variant of) phase retrieval.

Note that \( \langle x_i^\top \beta^*, -x_i^\top \beta^* \rangle \) has same information as \( |x_i^\top \beta^*| \).

Existing methods require \( n > 2d \).
Algorithmic results (Andoni, H., Shi, & Sun, 2017)

- **Noise-free setting** (i.e., $\sigma = 0$):
  Algorithm based on reduction to Subset Sum that requires $n \geq d + 1$, which is optimal.
Algorithmic results (Andoni, H., Shi, & Sun, 2017)

- **Noise-free setting** (i.e., $\sigma = 0$): Algorithm based on reduction to Subset Sum that requires $n \geq d + 1$, which is optimal.

- **General setting:** Method-of-moments algorithm that requires $n \geq d \cdot \text{poly}(k)$. 

Questions: SNR limits? Sub-optimality of “method-of-moments”? 

Algorithmic results (Andoni, H., Shi, & Sun, 2017)

- **Noise-free setting** (i.e., $\sigma = 0$):
  Algorithm based on reduction to Subset Sum that requires $n \geq d + 1$, which is optimal.

- **General setting**:
  Method-of-moments algorithm that requires $n \geq d \cdot \text{poly}(k)$. I.e., based on forming averages over the data, like:

\[
\frac{1}{n} \sum_{i=1}^{n} \left( \sum_{y_j \in \mathcal{Y}_i} y_j^2 \right) x_i x_i^\top.
\]

Questions: SNR limits? Sub-optimality of "method-of-moments"?
Algorithmic results (Andoni, H., Shi, & Sun, 2017)

- **Noise-free setting** (i.e., $\sigma = 0$): Algorithm based on reduction to Subset Sum that requires $n \geq d + 1$, which is optimal.

- **General setting**:
  Method-of-moments algorithm that requires $n \geq d \cdot \text{poly}(k)$. I.e., based on forming averages over the data, like:

\[
\frac{1}{n} \sum_{i=1}^{n} \left( \sum_{y_j \in \mathcal{Y}_i} y_j^2 \right) x_i x_i^\top.
\]

**Questions:** SNR limits? Sub-optimality of “method-of-moments”?
Closing remarks and open problems
Learning without correspondence is challenging for computation and statistics.

- Computational and information-theoretic hardness show striking contrast to "known correspondence" settings.
- New (and unexpected?) algorithmic techniques in worst-case and average-case settings.
- Open problems: Close gap between SNR lower and upper bounds? Lower bounds for correspondence retrieval? Faster/more robust algorithms? (Smoothed) analysis of alternating minimization?
Closing remarks and open problems

Learning without correspondence is challenging for computation and statistics.

- Computational and information-theoretic hardness show striking contrast to “known correspondence” settings.

- Open problems:
  - Close gap between SNR lower and upper bounds?
  - Lower bounds for correspondence retrieval?
  - Faster/more robust algorithms?
  - (Smoothed) analysis of alternating minimization?
Learning without correspondence is challenging for computation and statistics.

- **Computational and information-theoretic hardness** show striking contrast to “known correspondence” settings.

- **New (and unexpected?) algorithmic techniques** in worst-case and average-case settings.

### Open problems

- Close gap between SNR lower and upper bounds?
- Lower bounds for correspondence retrieval?
- Faster/more robust algorithms?
- (Smoothed) analysis of alternating minimization?
Learning without correspondence is challenging for computation and statistics.

- **Computational and information-theoretic hardness** show striking contrast to “known correspondence” settings.

- **New (and unexpected?) algorithmic techniques** in worst-case and average-case settings.

- **Open problems:**
  - Close gap between SNR lower and upper bounds?
  - Lower bounds for correspondence retrieval?
  - Faster/more robust algorithms?
  - (Smoothed) analysis of alternating minimization?
Acknowledgements

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

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


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

$$y_{\pi_\star}(i) = x_i^\top \beta_\star, \quad i \in [n].$$
“Realizable” case: Suppose there exist $\beta_\star \in \mathbb{R}^d$ and $\pi_\star \in S_n$ s.t.

$$y_{\pi_\star}(i) = x_i^\top \beta_\star, \quad i \in [n].$$

Solution is determined by action of $\pi_\star$ on $d$ points

(assume $\dim(\text{span}(x_i)_{i=1}^d) = d$).
"Realizable" case: Suppose there exist $\beta_* \in \mathbb{R}^d$ and $\pi_* \in S_n$ s.t.

$$y_{\pi_*}(i) = x_i^\top \beta_*, \quad i \in [n].$$

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

**Algorithm:**

- Find subset of $d$ linearly independent points $x_{i_1}, x_{i_2}, \ldots, x_{i_d}$.
- "Guess" values of $\pi_*(i_j) \in [d], j \in [d]$.
- Solve linear system $y_{\pi_*}(i_j) = x_{i_j}^\top \beta, j \in [d]$, for $\beta \in \mathbb{R}^d$.
- To check correctness of $\hat{\beta}$: compute $\hat{y}_i := x_i^\top \hat{\beta}, i \in [n]$, and check if $\min_{\pi \in S_n} \sum_{i=1}^n (y_{\pi}(i) - \hat{y}_i)^2 = 0$. 
"Realizable" case: Suppose there exist $\beta_\star \in \mathbb{R}^d$ and $\pi_\star \in S_n$ s.t.

$$y_{\pi_\star}(i) = x_i^\top \beta_\star, \quad i \in [n].$$

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

**Algorithm:**

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

"Guess" means "enumerate over $\binom{n}{d}$ choices"; rest is poly($n, d$).
**General case**: solution may not be determined by only $d$ points.
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 $\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^d} \sum_{j=1}^{d} (x_{i_j}^\top \beta - b_{i_j})^2$ satisfies

$$
\sum_{i=1}^{n} (x_{i}^\top \hat{\beta} - b_i)^2 \leq (d + 1) \cdot \min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (x_{i}^\top \beta - b_i)^2.
$$
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 \( \hat{\beta} \in \text{arg min}_{\beta \in \mathbb{R}^d} \sum_{j=1}^{d} (x_{i_j}^\top \beta - b_{i_j})^2 \) satisfies

\[
\sum_{i=1}^{n} (x_{i}^\top \hat{\beta} - b_{i})^2 \leq (d + 1) \cdot \min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (x_{i}^\top \beta - b_{i})^2 .
\]

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

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

\[
\hat{\beta} \in \arg \min_{\beta \in \mathbb{R}^d} \sum_{j=1}^{d} (x_{i_j}^\top \beta - b_{i_j})^2 \text{ satisfies }
\]

\[
\sum_{i=1}^{n} (x_{i_i}^\top \hat{\beta} - b_i)^2 \leq (d + 1) \cdot \min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (x_{i_i}^\top \beta - b_i)^2 .
\]

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

or $n^{\tilde{O}(d/\epsilon)}$-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 $k = \dim(\text{span}(x_i)_{i=1}^{n})$. 
Lower bound proof sketch

We show that no estimator can confidently distinguish between $\beta^* = e_1$ and $\beta^* = -e_1$, where $e_1 = (1, 0, \ldots, 0)^T$. 
Lower bound proof sketch

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

Let $P_{\beta^*}$ be the data distribution with parameter $\beta^* \in \{e_1, -e_1\}$.

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

$$\max_{\beta^* \in \{e_1, -e_1\}} \mathbb{E}_{P_{\beta^*}} \|\hat{\beta} - \beta^*\|_2 \geq 1 - \|P_{e_1} - P_{-e_1}\|_{tv}.$$
Lower bound proof sketch

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

Let $P_{\beta^*}$ be the data distribution with parameter $\beta^* \in \{e_1, -e_1\}$.

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

$$\max_{\beta^* \in \{e_1, -e_1\}} \mathbb{E}_{P_{\beta^*}} \| \hat{\beta} - \beta^* \|_2 \geq 1 - \| P_{e_1} - P_{-e_1} \|_{tv}. $$

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

Generative process for $P_{\beta^*}$:

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 $u_i := x_i^\top \beta^*$ for $i \in [n]$.

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

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

Under $P_{e_1}$:

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

Under $P_{-e_1}$:

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

Data processing: Lose information by going from $y$ to $H y_i I_n$. 


Generative process for $P_{\beta^*}$:

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

Generative process for $P_{\beta^*}$:

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 $u_i := x_i^T \beta^*$ for $i \in [n]$. 

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

Generative process for $P_{\beta^*}$:

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 $u_i := x_i^\top \beta^*$ 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 $P_{\beta^*}$:

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 $u_i := x_i^\top \beta^*$ 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 $H_y i_{n_i=1}^1$. 
Generative process for $P_{\beta^*}$:

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 $u_i := x_i^\top \beta^*$ 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,

\[
\text{KL} \left( \left. P_{e_1} \right| (x_i)_{i=1}^n, \left. P_{-e_1} \right| (x_i)_{i=1}^n \right) \\
\leq \text{KL} \left( N(u^\uparrow, \sigma^2 I_n), N(-u^\downarrow, \sigma^2 I_n) \right)
\]
By data processing inequality,

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

By data processing inequality,

$$\text{KL} \left( P_{e_1} \cdot | (x_i)_{i=1}^n, P_{-e_1} \cdot | (x_i)_{i=1}^n \right) \leq \text{KL} \left( N(u^\uparrow, \sigma^2 I_n), N(-u^\downarrow, \sigma^2 I_n) \right)$$

$$= \frac{\|u^\uparrow - (-u^\downarrow)\|^2_2}{2\sigma^2} = \frac{\text{SNR}}{2} \cdot \|u^\uparrow + u^\downarrow\|^2_2.$$
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,

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

Some computations show that

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

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 (H., Shi, & Sun, 2017)

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

$$y_0 = x_0^\top \beta^*; \quad y_i = x_{\pi^*(i)}^\top \beta^*, \quad i = 1, \ldots, n.$$ 

There is a poly$(n, d)$-time\footnote{Assuming problem is appropriately discretized.} algorithm that, given inputs $(x_i)_{i=0}^n$ and $(y_i)_{i=0}^n$, returns $\pi^*$ and $\beta^*$ with high probability.
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 $\mathbb{Z}$.
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 $v_\star$;
- any other lattice vector $\not\propto v_\star$ is more than $2^{N/2}$-times longer.

$$
\begin{bmatrix}
    b_0 & b_1 & \cdots & b_N \\
\end{bmatrix} := \begin{bmatrix}
    0 & I_N \\
    MT & -Mc_1 & \cdots & -Mc_N \\
\end{bmatrix}
$$

for sufficiently large $M > 0$. 

\[41\]
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 $\mathbb{Z}$. 
**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 $\mathbb{Z}$.

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

**Catch:** Our source numbers $c_{i,j} = y_i x_j^T x_0$ 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 \( c_{i,j} = y_i \mathbf{x}_j^\top \mathbf{x}_0 \) 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.

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

\[
\left| y_0 - \sum_{i,j} Z_{i,j} \cdot c_{i,j} \right| \geq \frac{1}{2^{\text{poly}(d)}} \cdot \| \beta^* \|_2.
\]