High Dimensional Linear Regression Without Sparsity: A Lattice-Based Approach

Ilias Zadik, joint work with David Gamarnik

Massachusetts Institute of Technology

MSR ML Ideas Lunch

November 13, 2018
Linear Regression

Let (unknown) $\beta^* \in \mathbb{R}^p$.
For measurement matrix $X \in \mathbb{R}^{n \times p}$, and noise vector $W \in \mathbb{R}^n$, we observe $n$ noisy linear samples of $\beta^*$, $Y = X\beta^* + W$.

Goal: Given $(Y, X)$, recover $\beta^*$. 
Linear Regression

Let (unknown) $\beta^* \in \mathbb{R}^p$.

For measurement matrix $X \in \mathbb{R}^{n \times p}$, and noise vector $W \in \mathbb{R}^n$, we observe $n$ noisy linear samples of $\beta^*$, $Y = X\beta^* + W$.

**Goal:** Given $(Y, X)$, recover $\beta^*$.

**Notation:** $n$ number of samples, $p$ number of features.
Linear Regression

Let (unknown) $\beta^* \in \mathbb{R}^p$.

For measurement matrix $X \in \mathbb{R}^{n \times p}$, and noise vector $W \in \mathbb{R}^n$, we observe $n$ noisy linear samples of $\beta^*$, $Y = X\beta^* + W$.

**Goal:** Given $(Y, X)$, recover $\beta^*$.

**Notation:** $n$ number of samples, $p$ number of features.

**Distributional ass.:** $X$ has iid $N(0, 1)$ entries and $W$ iid $N(0, \sigma^2)$.
Main Question

Question:
What is the minimum $n$ (as a function of the rest of the parameters) so that efficient and accurate recovery $\beta^*$ is possible?

An immediate answer under full generality: at least $p$.

Reason: Even if $W = 0$, we have $Y = X\beta^*$, a linear system with $p$ unknowns and $n$ equations! To solve it, we need at least $p$ equations, i.e. $n \geq p$. 

Gamarnik, Zadik (MIT) Regression with LLL algorithm November 13, 2018 3 / 30
Question:

What is the **minimum** $n$ (as a function of the rest of the parameters) so that *efficient and accurate* recovery $\beta^*$ is possible?

An immediate answer under full generality: at least $p$. 
Question: What is the minimum $n$ (as a function of the rest of the parameters) so that efficient and accurate recovery $\beta^*$ is possible?

An immediate answer under full generality: at least $p$.

Reason: Even if $W = 0$, we have $Y = X\beta^*$, a linear system with $p$ unknowns and $n$ equations! To solve it, we need at least $p$ equations, i.e. $n \geq p$. 
In many real-life applications of Linear Regression (e.g. natural language processing, genomics, pricing, MRI etc) we observe much more features than samples (i.e. $n \ll p, p \to +\infty$).
In many *real-life applications* of Linear Regression (e.g. natural language processing, genomics, pricing, MRI etc) we observe **much more** features than samples (i.e. $n \ll p, p \to +\infty$).

**Question**

Is there a way to make the inference of $\beta^*$ well-posed in such setting?
Structural Assumptions on $\beta^*$

- Sparsity: $k \leq p$ non zero coordinates.
  - Vast literature
    - $n > k \log(p/k)$: many known efficient algorithms (Lasso [Wainwright '09], OMP [Fletcher et al '11] etc)
    - $n < k \log(p/k)$: possible but evidence of computational hardness [Gamarnik, Z '17a], [Gamarnik, Z '17b].

- Other assumptions:
  - Block-sparsity [Baron et al',05], Tree-Sparsity [He et al '09], Output of a Generative Model [Bora et al '17]
  - Similar picture: can achieve some $n < p$ but not always small.

Question: Can we make other (natural) assumptions which works for smaller $n$? (e.g. constant $n$)
Structural Assumptions on $\beta^*$

- **Sparsity!** $k \leq p$ non zero coordinates.
  - Vast literature

- Other assumptions:
  - Block-sparsity [Baron et al',05], Tree-Sparsity [He et al '09], Output of a Generative Model [Bora et al '17]
  - Similar picture: can achieve some $n < p$ but not always small.

Question: Can we make other (natural) assumptions which works for smaller $n$? (e.g. constant $n$)
Structural Assumptions on $\beta^*$

- **Sparsity!** $k \leq p$ non-zero coordinates.
  - Vast literature
  - $n > k \log \left( \frac{p}{k} \right)$: many known **efficient** algorithms (Lasso [Wainwright '09], OMP [Fletcher et al '11] etc)

Other assumptions:
- Block-sparsity [Baron et al', 05]
- Tree-Sparsity [He et al '09]
- Output of a Generative Model [Bora et al '17]

Similar picture: can achieve some $n < p$ but not always small.

Question: Can we make other (natural) assumptions which works for smaller $n$? (e.g. constant $n$)
Structural Assumptions on $\beta^*$

- **Sparsity!** $k \leq p$ non zero coordinates.
  - Vast literature
  - $n > k \log \left( \frac{p}{k} \right)$: many known efficient algorithms (Lasso [Wainwright ’09], OMP [Fletcher et al ’11] etc)
  - $n < k \log \left( \frac{p}{k} \right)$: possible but evidence of computational hardness [Gamarnik, Z ’17a], [Gamarnik, Z ’17b].

- Other assumptions: Block-sparsity [Baron et al’05], Tree-Sparsity [He et al ’09], Output of a Generative Model [Bora et al ’17] etc

Similar picture: can achieve some $n < p$ but not always small.

Question: Can we make other (natural) assumptions which works for smaller $n$? (e.g. constant $n$)

Gamarnik, Zadik (MIT)
Structural Assumptions on $\beta^*$

- **Sparsity!** $k \leq p$ non zero coordinates.
  - Vast literature
  - $n > k \log \left( \frac{p}{k} \right)$: many known **efficient** algorithms (Lasso [Wainwright ’09], OMP [Fletcher et al ’11] etc)
  - $n < k \log \left( \frac{p}{k} \right)$: possible but evidence of **computational hardness** [Gamarnik, Z ’17a], [Gamarnik, Z ’17b].

- **Other assumptions:**
  Block-sparsity [Baron et al’,05], Tree-Sparsity [He et al ’09], Output of a Generative Model [Bora et al ’17] etc

Similar picture: can achieve some $n < p$ but not always small.
Structural Assumptions on $\beta^*$

- **Sparsity!** $k \leq p$ non zero coordinates.
  - Vast literature
  - $n > k \log \left(\frac{p}{k}\right)$: many known **efficient** algorithms
    (Lasso [Wainwright ’09], OMP [Fletcher et al ’11] etc)
  - $n < k \log \left(\frac{p}{k}\right)$: possible but evidence of **computational hardness**
    [Gamarnik, Z ’17a], [Gamarnik, Z ’17b].

- **Other assumptions:**
  Block-sparsity [Baron et al’,05], Tree-Sparsity [He et al ’09], Output of a Generative Model [Bora et al ’17] etc

  Similar picture: can achieve some $n < p$ but not always small.

**Question**

Can we make other (natural) assumptions which works for smaller $n$? (e.g. **constant** $n$)
New efficient algorithm for recovering $\beta^*$ from $(Y, X)$ under a new generic structural assumption (Q-rationality assumption) based on a connection with lattice-based algorithms.
New efficient algorithm for recovering $\beta^*$ from $(Y, X)$ under a new generic structural assumption (Q-rationality assumption) based on a connection with lattice-based algorithms.

Guarantees:
works for any $n$ (even $n = 1$) given sufficiently small noise!
Outline of the talk

1. Q-rationality assumption and definition of the model
2. Main results
3. Description of the successful algorithm
4. (Preliminary) experiments
5. Summary and future work
Outline of the talk

1. Q-rationality assumption and definition of the model
2. Main results
3. Description of the successful algorithm
4. (Preliminary) experiments
5. Summary and future work
### The Assumption

Every entry of $\beta^*$ is a **rational number** with denominator $Q$.

*(Alternatively: For $Q = 2^M$, log $Q$ bits after zero position per entry)*
The Assumption

Every entry of $\beta^*$ is a **rational number** with denominator $Q$.

(Alternatively: For $Q = 2^M$, log $Q$ bits after zero position per entry)

Motivation:
Q-rationality assumption

The Assumption

Every entry of $\beta^*$ is a **rational number** with denominator $Q$.

*(Alternatively: For $Q = 2^M$, log $Q$ bits after zero position per entry)*

Motivation:

1. Large $Q$: **Large** but **finite domain** for the coefficients.
Q-rationality assumption

The Assumption

Every entry of $\beta^*$ is a **rational number** with denominator $Q$.

*Alternatively:* For $Q = 2^M$, log $Q$ bits after zero position per entry

Motivation:

1. **Large** $Q$: **Large** but **finite domain** for the coefficients.
   
2. **Small** $Q$: Standard in wireless communication.
   
   E.g. linear models for GPS (e.g. [Boyd, Hassibi ’98]),
   
   **Physics laws** imply for half the coordinates $\beta^*_i \in \mathbb{Z} [Q = 1]$. 
The Assumption

Every entry of $\beta^*$ is a **rational number** with denominator $Q$.

(Alternatively: For $Q = 2^M$, log $Q$ bits after zero position per entry)

Motivation:

1. **Large $Q$:** Large but **finite domain** for the coefficients.

2. **Small $Q$:** Standard in wireless communication. E.g. linear models for GPS (e.g. [Boyd, Hassibi ’98]), **Physics laws** imply for half the coordinates $\beta^*_i \in \mathbb{Z} [Q = 1]$. 
Q-rationality assumption

The Assumption

Every entry of $\beta^*$ is a rational number in $[0, 1]$ with denominator Q. 
(Alternatively: For $Q = 2^M$, log $Q$ bits after zero position per entry)

Motivation:

(1) Large $Q$: Large but finite domain for the coefficients.

(2) Small $Q$: Standard in wireless communication. 
E.g. linear models for GPS (e.g. [Boyd, Hassibi ’98]), 
Physics laws imply for half the coordinates $\beta^*_i \in \mathbb{Z} [Q = 1]$. 

Gamarnik, Zadik (MIT)  Regression with LLL algorithm  November 13, 2018  8 / 30
The New Model

Setup: Let \( \beta^* \) be a Q-rational vector. For

- \( X \in \mathbb{R}^{n \times p} \) consisting of entries i.i.d \( N(0, 1) \) random variables
- \( W \in \mathbb{R}^n \) consisting of entries i.i.d. \( N(0, \sigma^2) \) random variables

we get \( n \) noisy linear samples of \( \beta^* \), \( Y \in \mathbb{R}^n \), given by,

\[
Y := X\beta^* + W.
\]
The New Model

Setup: Let $\beta^*$ be a Q-rational vector. For

- $X \in \mathbb{R}^{n \times p}$ consisting of entries i.i.d $N(0, 1)$ random variables
- $W \in \mathbb{R}^n$ consisting of entries i.i.d. $N(0, \sigma^2)$ random variables

we get $n$ noisy linear samples of $\beta^*$, $Y \in \mathbb{R}^n$, given by,

$$Y := X\beta^* + W.$$ 

Goal: Given $(Y, X)$, recover efficiently and exactly $\beta^*$ with $n$ as small as possible.
The recovery should happen with probability tending to 1 as $p$ tend to infinity (w.h.p.).
Outline of the talk

(1) Q-rationality assumption and definition of the model
(2) Main results
(3) Description of the successful algorithm
(4) (Preliminary) experiments
(5) Summary and future work
Brute-force with one sample

Any hope for $n = 1$? Recall $y_1 = \langle X_1, \beta^* \rangle + w_1$. 
Brute-force with one sample

Any hope for $n = 1$? Recall $y_1 = \langle X_1, \beta^* \rangle + w_1$.

Yes, if $\sigma = 0$! *Reason:* continuous data but discrete $\beta^*$.
Brute-force with one sample

Any hope for $n = 1$? Recall $y_1 = \langle X_1, \beta^* \rangle + w_1$.

**Yes, if $\sigma = 0$!** *Reason:* continuous data but discrete $\beta^*$.

**Brute Force Algorithm**

Check all $Q$-rational $\beta$ for

$$y_1 = \langle X_1, \beta \rangle.$$
Brute-force with one sample

Any hope for \( n = 1 \)? Recall \( y_1 = \langle X_1, \beta^* \rangle + w_1 \).

**Yes, if \( \sigma = 0 \)! Reason:** continuous data but discrete \( \beta^* \).

**Brute Force Algorithm**

Check all Q-rational \( \beta \) for

\[
y_1 = \langle X_1, \beta \rangle.
\]

**Termination Time**

\[
(Q + 1)(Q + 1) \ldots (Q + 1) = (Q + 1)^p - \text{not efficient!}
\]

\( p \) terms
Lemma (Brute-force works!)

Suppose $\beta^*$ Q-rational and $\sigma^2 = 0$, i.e. $y_1 = \langle X_1, \beta^* \rangle$.
There is no Q-rational $\beta \neq \beta^*$ with $y_1 = \langle X_1, \beta \rangle$, almost surely.
Lemma (Brute-force works!)

Suppose $\beta^*$ is $Q$-rational and $\sigma^2 = 0$, i.e. $y_1 = \langle X_1, \beta^* \rangle$. There is no $Q$-rational $\beta \neq \beta^*$ with $y_1 = \langle X_1, \beta \rangle$, almost surely.

Proof Sketch:

For any $\beta \neq \beta^*$,

$$P(y_1 = \langle X_1, \beta \rangle) = P(\langle X_1, \beta^* \rangle = \langle X_1, \beta \rangle) = P(\langle X_1, \beta^* - \beta \rangle = 0) = 0,$$

since $\langle X_1, \beta^* - \beta \rangle \sim N(0, \|\beta^* - \beta\|_2^2)$.

Union bound over $Q$-rational $\beta$: $P(\exists \beta \text{ Q-rational : } y_1 = \langle X_1, \beta \rangle) = 0.$
(1) Consider a **standard** recovery mechanism: $\min ||\beta||_\infty$, s.t. $Y = X\beta$. For $\beta^* \in \{-1, 1\}^p$, $\sigma = 0$ works iff $n > p/2$, i.e. needs linear samples ([Chandrasekaran et al '10], [Amelunxen et al '13]).
(1) Consider a **standard** recovery mechanism: \( \min \| \beta \|_\infty \), s.t. \( Y = X \beta \).

For \( \beta^* \in \{-1, 1\}^p \), \( \sigma = 0 \) works iff \( n > p/2 \), i.e. needs linear samples ([Chandrasekaran et al '10], [Amelunxen et al '13])

(2) **Physics-based** algorithm works for \( n = o(p) \) and any \( Q \) but

(i) we don't know more than \( n = o(p) \) and

(ii) needs delicate \( X \) (not iid!)

([Donoho et al' '11])
(1) Consider a **standard** recovery mechanism: \( \min \| \beta \|_\infty \), s.t. \( Y = X\beta \). For \( \beta^* \in \{-1, 1\}^p \), \( \sigma = 0 \) works iff \( n > p/2 \), i.e. needs linear samples ([Chandrasekaran et al ’10], [Amelunxen et al ’13])

(2) **Physics-based** algorithm works for \( n = o(p) \) and any \( Q \) but
   (i) we don’t know more than \( n = o(p) \) and
   (ii) needs delicate \( X \) (not iid!)
   ([Donoho et al’ ’11])

**Question**
Is there a computationally efficient algorithm for constant sample size?
Theorem (informal, (Gamarnik, Z. NIPS ’18))

Suppose you have \( n = o(p) \) samples and \( 0 \leq \sigma \leq \exp \left(\frac{-p \max\{p, \log Q\}}{n}\right) \).

Then there exists a **polynomial-in-\( n, p, \log Q \)** time algorithm which with input \((Y, X)\) it outputs \( \beta^* \) \textit{w.h.p.} as \( p \to +\infty \).
Theorem (informal, (Gamarnik, Z. NIPS ’18))

Suppose you have \( n = o(p) \) samples and \( 0 \leq \sigma \leq \exp\left(-\frac{p \max\{p, \log Q\}}{n}\right) \).

Then there exists a polynomial-in-\( n, p, \log Q \) time algorithm which with input \( (Y, X) \) it outputs \( \beta^* \) w.h.p. as \( p \to +\infty \).

1. An efficient algorithm which works for any sample size \( n = o(p) \), also \( n = 1! \).
2. For \( n = 1 \), it works in time poly in \( p, \log Q \), an exponential decrease from brute-force \( (Q + 1)^p \).
3. The algorithm works with any \( X \) with iid well-behaved continuous entries (work even for uniform iid integer in a large domain)!
Theorem (informal, (Gamarnik, Z. NIPS '18))

Suppose you have $n = o(p)$ samples and $0 \leq \sigma \leq \exp\left(-\frac{p \max\{p, \log Q\}}{n}\right)$. Then there exists a polynomial-in-$n, p, \log Q$ time algorithm which with input $(Y, X)$ it outputs $\beta^*$ w.h.p. as $p \to +\infty$.

Call $\sigma_0 = \exp\left(-\frac{p \max\{p, \log Q\}}{n}\right)$. Is this the optimal amount of noise?
Theorem (informal, (Gamarnik, Z. NIPS '18))

Suppose you have \( n = o(p) \) samples and \( 0 \leq \sigma \leq \exp\left(-\frac{p \max\{p, \log Q\}}{n}\right) \). Then there exists a polynomial-in-\( n, p, \log Q \) time algorithm which with input \((Y, X)\) it ouputs \( \beta^* \) w.h.p. as \( p \to +\infty \).

Call \( \sigma_0 = \exp\left(-\frac{p \max\{p, \log Q\}}{n}\right) \). Is this the optimal amount of noise?

Theorem (informal, (Gamarnik, Z. NIPS '18))

Suppose you have \( n = o(p) \) samples and \( \sigma > \exp\left(-\frac{p \log Q}{n}\right) \). Then its impossible to w.h.p. recover correctly \( \beta^* \) with any algorithm with only access to \((Y, X)\).
Theorem (informal, (Gamarnik, Z. NIPS ’18))

Suppose you have \( n = o(p) \) samples and \( 0 \leq \sigma \leq \exp\left(-\frac{p \max\{p, \log Q\}}{n}\right) \).
Then there exists a polynomial-in-\( n, p, \log Q \) time algorithm which with input \((Y, X)\) it outputs \( \beta^* \) w.h.p. as \( p \to +\infty \).

Theorem (informal, (Gamarnik, Z. NIPS ’18))

Suppose you have \( n = o(p) \) samples and \( \sigma > \exp\left(-\frac{p \log Q}{n}\right) \).
Then it’s impossible to w.h.p. recover correctly \( \beta^* \) with any algorithm with only access to \((Y, X)\).

If \( Q > 2^p \): our algorithm has optimal noise-tolerance!
Outline of the talk

(1) Q-rationality assumption and definition of the model
(2) Main results
(3) **Description of the successful algorithm**
(4) (Preliminary) experiments
(5) Summary and future work
The Algorithm: Connecting HDLR with Lattices

Key Influence:
The LLL algorithm ['82] and its application on solving randomly generated subset-sum problems [Lagarias, Odlyzko '83], [Frieze '84].
The Algorithm: Connecting HDLR with Lattices

Key Influence:
The LLL algorithm ['82] and its application on solving randomly generated subset-sum problems [Lagarias, Odlyzko '83], [Frieze '84].

Plan
The Algorithm: Connecting HDLR with Lattices

**Key Influence:**
The LLL algorithm ['82] and its application on solving randomly generated subset-sum problems [Lagarias, Odlyzko '83], [Frieze '84].

**Plan**

1. Describe LLL guarantees and general strategy

Gamarnik, Zadik (MIT)  Regression with LLL algorithm  November 13, 2018  16 / 30
Key Influence:
The LLL algorithm ['82] and its application on solving randomly generated subset-sum problems [Lagarias, Odlyzko ’83], [Frieze ’84].

Plan
(1) Describe LLL guarantees and general strategy
(2) Details for using LLL to find $\beta^*$ in some restrictive case
Key Influence:
The LLL algorithm [’82] and its application on solving randomly generated subset-sum problems [Lagarias, Odlyzko ’83], [Frieze ’84].

Plan

(1) Describe LLL guarantees and general strategy
(2) Details for using LLL to find $\beta^*$ in some restrictive case
(3) Describe the general algorithm
Let $b_1, \ldots, b_m \in \mathbb{Z}^p$ linearly independent vectors.

**Definition**

The lattice $\mathcal{L}$ spanned by $b_1, \ldots, b_m$ is the set of all integer combinations of the $m$ vectors.
The LLL algorithm

“The Shortest Vector Problem” $\min_{x \in \mathcal{L} \setminus \{0\}} \|x\|_2$

Shortest Vector Problem (SVP): given a lattice, find a shortest (nonzero) vector
The LLL algorithm

“The Shortest Vector Problem”  \( \min_{x \in \mathcal{L} \setminus \{0\}} \|x\|_2 \)

- NP-hard but,
- A famous algorithm proposed by Lenstra-Lenstra-Lovasz (LLL) efficiently *approximates* it; find \( \hat{x} \in \mathcal{L} \setminus \{0\} \) with

\[
\|\hat{x}\|_2 \leq 2^{p/2} \min_{x \in \mathcal{L} \setminus \{0\}} \|x\|_2.
\]

Time poly in \( p, \log \max_i \{\|b_i\|_\infty\} \)
Use LLL for Finding $\beta^*$

Main Steps:

• Create a lattice $L = L(Y, X)$ such that "approximately" shortest vectors of $L$ are multiples of $\beta^*$

• Use LLL and recover a multiple of $\beta^*$.

• Recover $\beta^*$ from a multiple of $\beta^*$ (needs some structure).
Use LLL for Finding $\beta^*$

Main Steps:

- Create a lattice $\mathcal{L} = \mathcal{L}(Y, X)$ such that “approximately” shortest vectors of $\mathcal{L}$ ↔ multiples of $\beta^*$
- Use LLL and recover a multiple of $\beta^*$
- Recover $\beta^*$ from a multiple of $\beta^*$ (needs some structure)
Use LLL for Finding $\beta^*$

Main Steps:

- Create a lattice $\mathcal{L} = \mathcal{L}(Y, X)$ such that “approximately” shortest vectors of $\mathcal{L} \leftrightarrow$ multiples of $\beta^*$

- Use LLL and recover a multiple of $\beta^*$. 
Use LLL for Finding $\beta^*$

Main Steps:

- Create a lattice $\mathcal{L} = \mathcal{L}(Y, X)$ such that "approximately" shortest vectors of $\mathcal{L} \leftrightarrow$ multiples of $\beta^*$
- Use LLL and recover a multiple of $\beta^*$.
- Recover $\beta^*$ from a multiple of $\beta^*$ (needs some structure).
The Algorithm (special case, [F ’84])

Assume

- $n = 1$, $\sigma = 0$, $\beta^*$ binary: $y = \langle X_1, \beta^* \rangle$.
- $X_1 \in \mathbb{Z}^p$ with iid uniform in $[2^N]$ entries for large $N$ (say $N = p^2$).
The Algorithm (special case, [F ’84])

Assume

• \( n = 1, \sigma = 0, \beta^* \) binary: \( y = \langle X_1, \beta^* \rangle \).
• \( X_1 \in \mathbb{Z}^p \) with iid uniform in \( [2^N] \) entries for large \( N \) (say \( N = p^2 \)).

(1) For \( M \) sufficiently large enough set \( \mathcal{L}_M(y_1, X_1) \) produced by the columns of

\[
A_M := \begin{bmatrix}
MX_1 & -My_1 \\
l_{p \times p} & 0
\end{bmatrix}
\]
The Algorithm (special case, [F ’84])

Assume

- \( n = 1, \sigma = 0, \beta^* \) binary: \( y = \langle X_1, \beta^* \rangle \).
- \( X_1 \in \mathbb{Z}^p \) with iid uniform in \([2^N]\) entries for large \( N \) (say \( N = p^2 \)).

(1) For \( M \) sufficiently large enough set \( \mathcal{L}_M(y_1, X_1) \) produced by the columns of

\[
A_M := \begin{bmatrix}
MX_1 & -My_1 \\
I_{p \times p} & 0
\end{bmatrix}
\]

**Lemma:** Each \( z \in \mathcal{L}_M, \|z\|_2 < M \) is a multiple of \( \begin{bmatrix} 0 \\ \beta^* \end{bmatrix} \), w.h.p. (\( N \) large)
Assume

- \( n = 1, \sigma = 0, \beta^* \) binary: \( y = \langle X_1, \beta^* \rangle \).
- \( X_1 \in \mathbb{Z}_p \) with iid uniform in \( [2^N] \) entries for large \( N \) (say \( N = p^2 \)).

(1) For \( M \) sufficiently large enough set \( L_M(y_1, X_1) \) produced by the columns of

\[
A_M := \begin{bmatrix}
MX_1 & -My_1 \\
l_{p \times p} & 0
\end{bmatrix}
\]

**Lemma:** Each \( z \in L_M, \|z\|_2 < M \) is a multiple of \( \begin{bmatrix} 0 \\ \beta^* \end{bmatrix} \), w.h.p. (\( N \) large)

**Intuition:**

\[
z = A_M \begin{bmatrix}
\beta \\
\lambda
\end{bmatrix} = \begin{bmatrix}
M\langle X_1, \beta \rangle - M\lambda y_1 \\
\beta
\end{bmatrix} = \begin{bmatrix}
M\langle X_1, \beta - \lambda \beta^* \rangle \\
\beta
\end{bmatrix},
\]

\[
P(\text{Lemma is false}) \leq P(\exists \beta \neq \lambda \beta^* : \|\beta\|_2 < M, \langle X_1, \beta - \lambda \beta^* \rangle = 0) \to 0.
\]
Choose $M$ appropriately so that LLL outputs a multiple of $\beta^*$. 

(2) Choose $M$ appropriately so that LLL outputs a multiple of $\beta^*$. 

We know $A_M[\beta^*] = [0 \beta^*] \in L$. 

LLL outputs $\hat{x}$ with norm at most $2p^2 \| [0 \beta^*] \|_2 \leq 2p^2 \sqrt{p} < M$. 

Using the lemma we are done!
(2) Choose $M$ appropriately so that LLL outputs a multiple of $\beta^*$. 

- Choose $M = 2^\frac{p}{2} \sqrt{p} + 1$. 
- We know $A_M \begin{bmatrix} \beta^* \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \beta^* \end{bmatrix} \in \mathcal{L}$. 
- LLL outputs $\hat{x}$ with norm at most $2^\frac{p}{2} \| \begin{bmatrix} 0 \\ \beta^* \end{bmatrix} \|_2 \leq 2^\frac{p}{2} \sqrt{p} < M$. 
- Using the lemma we are done!
(2) Choose $M$ appropriately so that LLL outputs a multiple of $\beta^*$.  

- Choose $M = 2^{\frac{p}{2}} \sqrt{p} + 1$.  
- We know $A_M \begin{bmatrix} \beta^* \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \beta^* \end{bmatrix} \in \mathcal{L}$.  
- LLL outputs $\hat{x}$ with norm at most $2^{\frac{p}{2}} \left\| \begin{bmatrix} 0 \\ \beta^* \end{bmatrix} \right\|_2 \leq 2^{\frac{p}{2}} \sqrt{p} < M$.  
- Using the lemma we are done!

(3) Rescale to get $\beta^*$.  

Gamarnik, Zadik (MIT)  
Regression with LLL algorithm  
November 13, 2018  
21 / 30
Restricted case:

(1) One sample $n = 1$.
(2) Noiseless $\sigma = 0$
(3) Integer $X$
(4) Binary coefficients $\beta^*$
Restricted case $\rightarrow$ General case:

1. One sample $n = 1 \rightarrow$ many samples $n > 1$.
2. Noiseless $\sigma = 0 \rightarrow$ noisy $\sigma > 0$
3. Integer $Y, X \rightarrow$ real $Y, X$
4. Binary coefficients $\beta^* \rightarrow$ Q-rational $\beta^*$
Towards the General Algorithm

**Restricted case → General case:**

1. One sample $n = 1 \rightarrow$ many samples $n > 1$.
2. Noiseless $\sigma = 0 \rightarrow$ noisy $\sigma > 0$
3. Integer $Y, X \rightarrow$ real $Y, X$
4. Binary coefficients $\beta^* \rightarrow$ Q-rational $\beta^*$

**Methods:**

1),(2) follow by redesigning the lattice basis.
3),(4) follow by reductions to the integer case using the linear structure.
Step 1: $n > 1$ and $\sigma > 0$

Key Idea

Change the lattice basis to the columns of the $(2n + p) \times (2n + p)$ matrix

$$A_M := \begin{bmatrix}
  MX & -MDiag_{n \times n} (Y) & MI_{n \times n} \\
  I_{p \times p} & O_{p \times n} & 0_{p \times n} \\
  O_{n \times p} & 0_{n \times n} & I_{n \times n}
\end{bmatrix}$$
Step 1: \( n > 1 \) and \( \sigma > 0 \)

Key Idea

Change the lattice basis to the columns of the \((2n + p) \times (2n + p)\) matrix

\[
A_M := \begin{bmatrix}
MX & -M\text{Diag}_{n \times n}(Y) & MI_{n \times n} \\
I_{p \times p} & 0_{p \times n} & 0_{p \times n} \\
0_{n \times p} & 0_{n \times n} & I_{n \times n}
\end{bmatrix}
\]

- Extra columns to use more observations and handle noise
- Similar proof
Step 2: Real-valued $(Y, X)$ and Q-rational $\beta^*$

**Real-valued $(Y, X)$**

Truncate $(Y, X)$ to the first $N$ coordinates and reduce it to integer $(Y, X)$. 

$$y = \langle x, \beta^* \rangle + w \rightarrow 2^N y = \langle 2^N x, \beta^* \rangle + 2^N w \rightarrow \lfloor 2^N y \rfloor = \langle \lfloor 2^N x \rfloor, \beta^* \rangle + w'.$$

**Q rational $\beta^*$**

Reduce to integer $\beta^*$ with $\gcd(\beta^*) = 1$, so that it is recovered from any multiple of $\beta^*$!

First, $y = \langle x, \beta^* \rangle + w$ implies $Qy = \langle x, Q\beta^* \rangle + Qw$.

Second, for random $Z$, $y + \langle x, Z \rangle = \langle x, \beta^* + Z \rangle + w$ and $\gcd(\beta^* + Z) = 1$, w.h.p. (analytic number theory argument)
Step 2: Real-valued \((Y, X)\) and Q-rational \(\beta^*\)

**Real-valued \((Y, X)\)**

Truncate \((Y, X)\) to the first \(N\) coordinates and reduce it to integer \((Y, X)\).

\[
y = \langle x, \beta^* \rangle + w \rightarrow 2^Ny = \langle 2^Nx, \beta^* \rangle + 2^Nw \rightarrow [2^Ny] = \langle [2^Nx], \beta^* \rangle + w'.
\]
Step 2: Real-valued \((Y, X)\) and Q-rational \(\beta^*\)

### Real-valued \((Y, X)\)

Truncate \((Y, X)\) to the first \(N\) coordinates and reduce it to integer \((Y, X)\).

\[
y = \langle x, \beta^* \rangle + w \rightarrow 2^N y = \langle 2^N x, \beta^* \rangle + 2^N w \rightarrow \lfloor 2^N y \rfloor = \langle \lfloor 2^N x \rfloor, \beta^* \rangle + w'.
\]

### Q rational \(\beta^*\)

Reduce to integer \(\beta^*\) with \(\gcd(\beta^*) = 1\), so that it is recovered from any multiple of \(\beta^*\)!
Step 2: Real-valued \((Y, X)\) and Q-rational \(\beta^*\)

Real-valued \((Y, X)\)

Truncate \((Y, X)\) to the first \(N\) coordinates and reduce it to integer \((Y, X)\).

\[
y = \langle x, \beta^* \rangle + w \rightarrow 2^N y = \langle 2^N x, \beta^* \rangle + 2^N w \rightarrow \lfloor 2^N y \rfloor = \langle \lfloor 2^N x \rfloor, \beta^* \rangle + w'.
\]

Q rational \(\beta^*\)

Reduce to integer \(\beta^*\) with \(\gcd(\beta^*) = 1\), so that it is recovered from any multiple of \(\beta^*\)!

First, \(y = \langle x, \beta^* \rangle + w\) implies \(Qy = \langle x, Q\beta^* \rangle + Qw\).
Step 2: Real-valued \((Y, X)\) and \(\mathbb{Q}\)-rational \(\beta^*\)

**Real-valued \((Y, X)\)**

Truncate \((Y, X)\) to the first \(N\) coordinates and reduce it to integer \((Y, X)\).

\[
y = \langle x, \beta^* \rangle + w \rightarrow 2^N y = \langle 2^N x, \beta^* \rangle + 2^N w \rightarrow \lfloor 2^N y \rfloor = \langle \lfloor 2^N x \rfloor, \beta^* \rangle + w'.
\]

**\(\mathbb{Q}\) rational \(\beta^*\)**

Reduce to integer \(\beta^*\) with \(\gcd(\beta^*) = 1\), so that it is recovered from any multiple of \(\beta^*\)!

First, \(y = \langle x, \beta^* \rangle + w\) implies \(Qy = \langle x, Q\beta^* \rangle + Qw\).

Second, for random \(Z\),

\[
y + \langle x, Z \rangle = \langle x, \beta^* + Z \rangle + w\quad \text{and} \quad \gcd(\beta^* + Z) = 1, \text{ w.h.p.}
\]

(*analytic number theory argument*)
Outline of the talk

1. Q-rationality assumption and definition of the model
2. Main results
3. Description of the successful algorithm
4. (Preliminary) experiments
5. Summary and Future Work
Preliminary Experimental Results (small $p$)-Integer Data

Figure: 20 instances per dot: $p = 30$, $n = 1, 10, 30$, $X$ Uniform in $[2^N]$, noiseless, $\alpha \sim 1/N$.

(Work with Patricio Foncea, Andrew Zheng)
Figure: 20 instances per dot: $X, W$ Gaussian, $p = 30$ and $n = 10$.

(Work with Patricio Foncea, Andrew Zheng)
Outline of the talk

(1) Q-rationality assumption and definition of the model
(2) Main results
(3) Description of the successful algorithm
(4) (Preliminary) experiments
(5) **Summary and future work**
(1) **New rationality assumption** to perform high dimensional inference for linear regression.
(1) **New rationality assumption** to perform high dimensional inference for linear regression.

(2) Coefficients \( \beta^* \) are **efficiently recoverable** even when \( n = 1 \) and noise is small. Optimal noise tolerance for large \( Q \).
Summary

(1) **New rationality assumption** to perform high dimensional inference for linear regression.

(2) Coefficients $\beta^*$ are **efficiently recoverable** even when $n = 1$ and noise is small. Optimal noise tolerance for large $Q$.

(3) **Algorithmic connection** between Linear Regression and Shortest Vector Problem.
(1) New rationality assumption to perform high dimensional inference for linear regression.

(2) Coefficients $\beta^*$ are efficiently recoverable even when $n = 1$ and noise is small. Optimal noise tolerance for large $Q$.

(3) Algorithmic connection between Linear Regression and Shortest Vector Problem.

(4) Preliminary Synthetic experiments suggest the algorithm works also for small $p$. 
Future Directions

1. Other discrete prior distributions - ongoing work.

2. Other similar noiseless problems - ongoing work, e.g., Phase Retrieval $y_i = |\langle X_i, \beta^* \rangle|$.

3. Increase the noise level tolerance for small $Q$.

4. Perform more systematic experiments to verify method for small $p$.

Thank you!!
(1) Other discrete prior distributions - ongoing work.

(2) Other similar noiseless problems - ongoing work
   e.g. Phase Retrieval! \( y_i = |\langle X_i, \beta^* \rangle| \).

(3) Increase the noise level tolerance for small \( Q \).
   Also: use rationality and sparsity together as assumptions?

(4) Perform more systematic experiments to verify method for small \( p \).

Thank you!!
Future Directions

1. Other discrete prior distributions - ongoing work.

2. Other similar noiseless problems - ongoing work
   e.g. *Phase Retrieval!* \( y_i = |\langle X_i, \beta^* \rangle| \).

3. Increase the noise level tolerance for small \( Q \).
   Also: use *rationality and sparsity* together as assumptions?
Future Directions

(1) Other discrete prior distributions - ongoing work.

(2) Other similar noiseless problems - ongoing work  
   e.g. Phase Retrieval! \( y_i = |\langle X_i, \beta^* \rangle| \).

(3) Increase the noise level tolerance for small Q.  
   Also: use rationality and sparsity together as assumptions?

(4) Perform more systematic experiments to verify method for small p.
(1) Other discrete prior distributions - ongoing work.

(2) Other similar noiseless problems - ongoing work e.g. Phase Retrieval! \( (y_i = |\langle X_i, \beta^* \rangle|) \).

(3) Increase the noise level tolerance for small Q. Also: use rationality and sparsity together as assumptions?

(4) Perform more systematic experiments to verify method for small p.

Thank you!!