Linear Regression without Correspondences

through a computational lens

Liangzu Peng
School of Information Science and Technology
ShanghaiTech University
penglz@shanghaitech.edu.cn

November 9, 2019

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

When learning a d-dimensional subspace $\mathcal{V}$ from corrupted data, we have routinely encountered data corruptions such as:

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

When learning a d-dimensional subspace $\mathcal{V}$ from corrupted data, we have routinely encountered data corruptions such as:

dense noise ($\tilde{X} = X + \mathcal{E}$, X clean data, $\mathcal{E}$ noise)

$$\min_{\mathcal{V}} \left\| \tilde{X} - \mathbb{P}_{\mathcal{V}} \tilde{X} \right\|_{\mathsf{F}}, \mathbb{P}_{\mathcal{V}} \text{ is a projection on } \mathcal{V}. \tag{1}$$

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

When learning a d-dimensional subspace $\mathcal{V}$ from corrupted data, we have routinely encountered data corruptions such as:

dense noise

sparse corruption ($\tilde{X} = L + E$, L low-rank, E sparse)

$$\min_{L,E} \|L\|_* + \tau \|E\|_1, \text{ s.t. } \tilde{X} = L + E. \tag{2}$$

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

When learning a d-dimensional subspace $\mathcal{V}$ from corrupted data, we have routinely encountered data corruptions such as:

dense noise

sparse corruption

outliers ($\tilde{X} = [XO]\Gamma$, O outliers, Γ permutation)

$$\min_{L,E} \|L\|_* + \lambda \|E\|_{2,1}, \text{ s.t. } \tilde{X} = L + E. \tag{3}$$

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

When learning a d-dimensional subspace $\mathcal{V}$ from corrupted data, we have routinely encountered data corruptions such as:

dense noise

sparse corruption

outliers

missing entries (Matrix Completion)

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

When learning a d-dimensional subspace $\mathcal{V}$ from corrupted data, we have routinely encountered data corruptions such as:

dense noise

sparse corruption

outliers

missing entries

Question:

are these formulations feasible when corruptions **arbitrarily large**?

e.g., when no entries observed (Matrix Completion).

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

When learning a d-dimensional subspace $\mathcal{V}$ from corrupted data, we have routinely encountered data corruptions such as:

dense noise

sparse corruption

outliers

missing entries

Question:

are these formulations feasible when corruptions **arbitrarily large**?

no, e.g., when no entries observed (Matrix Completion).

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

When learning a d-dimensional subspace $\mathcal{V}$ from corrupted data, we have routinely encountered data corruptions such as:

dense noise

sparse corruption

outliers

missing entries

Question:

are these formulations feasible when corruptions **arbitrarily large**?

no, e.g., when no entries observed (Matrix Completion).

yes, with some special corruption (e.g., without correspondences).

Linear Regression without Correspondences

Motivation: a Subspace Learning perspective

Principle Component Analysis without Correspondences?

$\rightsquigarrow$ current research

$$\Downarrow$$

Linear Regression without Correspondences

$\rightsquigarrow$ this lecture

Linear Regression without Correspondences

Linear Regression.

$y = Ax + \epsilon$, $y \in \mathbb{R}^m, x \in \mathbb{R}^n, \epsilon$ noise

find x from y, A.

Example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = Ax + \epsilon.$$

Linear Regression without Correspondences

Linear Regression.

$y = Ax + \epsilon$, $y \in \mathbb{R}^m, x \in \mathbb{R}^n, \epsilon$ noise

find x from y, A.

Example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = Ax + \epsilon.$$

without Correspondences?

unknown correspondences between y's entries and A's rows

Example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = Ax + \epsilon? \quad \begin{bmatrix} y_2 \\ y_1 \\ y_3 \end{bmatrix} = Ax + \epsilon? \quad \begin{bmatrix} y_2 \\ y_3 \\ y_1 \end{bmatrix} = Ax + \epsilon?$$

Linear Regression without Correspondences

Linear Regression.

$y = Ax + \epsilon$, $y \in \mathbb{R}^m, x \in \mathbb{R}^n, \epsilon$ noise

find x from y, A.

Example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = Ax + \epsilon.$$

without Correspondences?

unknown correspondences between y's entries and A's rows

Example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = Ax + \epsilon? \quad \begin{bmatrix} y_2 \\ y_1 \\ y_3 \end{bmatrix} = Ax + \epsilon? \quad \begin{bmatrix} y_2 \\ y_3 \\ y_1 \end{bmatrix} = Ax + \epsilon?$$

$\iff$ $\Pi y = Ax + \epsilon$, Π unknown $m \times m$ permutation matrix.

Linear Regression without Correspondences

Model:

$$y = \Pi A x^* + \epsilon, \text{ where } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \tag{4}$$

and Π belongs to the set $\mathcal{P}$ of $m \times m$ permutation matrices.

Linear Regression without Correspondences

Model:

$$y = \Pi A x^* + \epsilon, \text{ where } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \tag{4}$$

and Π belongs to the set $\mathcal{P}$ of $m \times m$ permutation matrices.
such a problem is also known as

Linear Regression with Permuted Data

Linear Regression with Shuffled Labels

Linear Regression with (Partially) Mismatched Data

Shuffled Linear Regression

Linear Regression without Correspondences

Model:

$$y = \Pi A x^* + \epsilon, \text{ where } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \tag{4}$$

and Π belongs to the set $\mathcal{P}$ of $m \times m$ permutation matrices.
such a problem is also known as

Linear Regression with Permuted Data

Linear Regression with Shuffled Labels

Linear Regression with (Partially) Mismatched Data

Shuffled Linear Regression

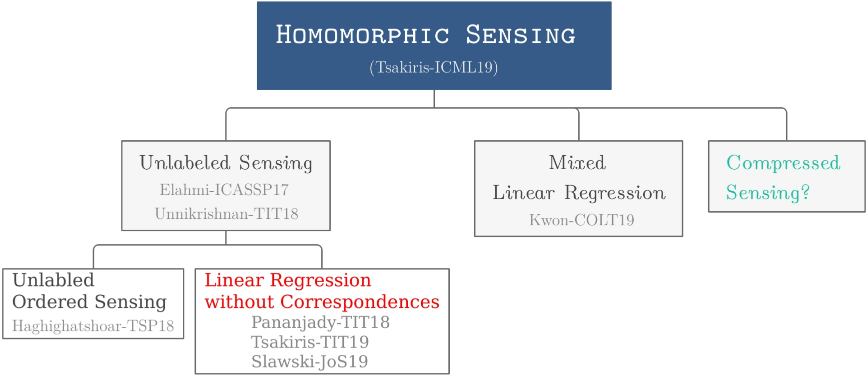
Thematic Question:

can linear regression be robust to permutation corruptions?

Linear Regression without Correspondences

Context

(we are not alone)



$$y = \Pi A x^* + \epsilon, \text{ where } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n. \tag{5}$$

Is (5) identifiable? ($\epsilon = 0$).

not identifiable if two different signals $x_1, x_2 \in \mathbb{R}^n$ cause the same observations, i.e.,

$$x_1 \neq x_2 \Rightarrow \Pi_1 A x_1 = \Pi_2 A x_2. \tag{6}$$

unique recovery of x_1 or x_2 is impossible if (6) happens.

Linear Regression without Correspondences

The model

$$y = \Pi A x^* + \epsilon, \text{ where } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n. \tag{5}$$

Is (5) identifiable? ($\epsilon = 0$).

not identifiable if two different signals $x_1, x_2 \in \mathbb{R}^n$ cause the same observations, i.e.,

$$x_1 \neq x_2 \Rightarrow \Pi_1 A x_1 = \Pi_2 A x_2. \tag{6}$$

unique recovery of x_1 or x_2 is impossible if (6) happens.

Hence we ask, under what conditions the following holds:

$$\forall x_1, x_2, \Pi_1, \Pi_2 : x_1 \neq x_2 \Rightarrow \Pi_1 A x_1 \neq \Pi_2 A x_2? \tag{7}$$

unique recovery of x^* is possible if (7) holds.

in Linear Regression (7) holds if $m \geq n$ & A full rank.

$$y = \Pi A x^* + \epsilon, \text{ where } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n. \tag{5}$$

Is (5) identifiable? ($\epsilon = 0$).

not identifiable if two different signals $x_1, x_2 \in \mathbb{R}^n$ cause the same observations, i.e.,

$$x_1 \neq x_2 \Rightarrow \Pi_1 A x_1 = \Pi_2 A x_2. \tag{6}$$

unique recovery of x_1 or x_2 is impossible if (6) happens.

Hence we ask, under what conditions the following holds:

$$\forall x_1, x_2, \Pi_1, \Pi_2 : x_1 \neq x_2 \Rightarrow \Pi_1 A x_1 \neq \Pi_2 A x_2? \tag{7}$$
$$\text{or } x_1 = x_2 \Leftarrow \Pi_1 A x_1 = \Pi_2 A x_2?$$

unique recovery of x^* is possible if (7) holds.

in Linear Regression (7) holds if $m \geq n$ & A full rank.

Linear Regression without Correspondences

The model

$$y = \Pi A x^* + \epsilon, \text{ where } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n. \tag{5}$$

Is (5) identifiable? ($\epsilon = 0$).

not identifiable if two different signals $x_1, x_2 \in \mathbb{R}^n$ cause the same observations, i.e.,

$$x_1 \neq x_2 \Rightarrow \Pi_1 A x_1 = \Pi_2 A x_2. \tag{6}$$

unique recovery of x_1 or x_2 is impossible if (6) happens.

Hence we ask, under what conditions the following holds:

$$\forall x_1, x_2, \Pi_1, \Pi_2 : x_1 \neq x_2 \Rightarrow \Pi_1 A x_1 \neq \Pi_2 A x_2? \tag{7}$$

$$\text{or } x_1 = x_2 \Leftarrow \Pi_1 A x_1 = \Pi_2 A x_2?$$

unique recovery of x^* is possible if (7) holds.

in Linear Regression (7) holds if $m \geq n$ & A full rank.

Theorem (Unnikrishan-TIT18)

(7) holds if $m \geq 2n$ & A random.

Theorem (Tsakiris-TIT19)

(7) holds if $m > n$ & A, x random.

Linear Regression without Correspondences

From unique recovery conditions to computation

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \Pi \in \mathcal{P}. \tag{8}$$

	recovery conditions	$\lvert\mathcal{P}\rvert$
Linear Regression	$m \geq n$ & A full rank	1
Linear Regression without Corr.	$m \geq 2n$ & A random	$m!$
Linear Regression without Corr.	$m > n$ & A, x random	$m!$

Good news: m does not increase exponentially as $\lvert\mathcal{P}\rvert$.

Linear Regression without Correspondences

From unique recovery conditions to computation

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \Pi \in \mathcal{P}. \tag{8}$$

	recovery conditions	$\lvert\mathcal{P}\rvert$
Linear Regression	$m \geq n$ & A full rank	1
Linear Regression without Corr.	$m \geq 2n$ & A random	$m!$
Linear Regression without Corr.	$m > n$ & A, x random	$m!$

Good news: m does not increase exponentially as $\lvert\mathcal{P}\rvert$.

question: an intuitive justification for this phenomenon?

Linear Regression without Correspondences

From unique recovery conditions to computation

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \Pi \in \mathcal{P}. \tag{8}$$

	recovery conditions	$\lvert\mathcal{P}\rvert$
Linear Regression	$m \geq n$ & A full rank	1
Linear Regression without Corr.	$m \geq 2n$ & A random	$m!$
Linear Regression without Corr.	$m > n$ & A, x random	$m!$

Good news: m does not increase exponentially as $\lvert\mathcal{P}\rvert$.

question: an intuitive justification for this phenomenon?

But: does it imply computational easiness?

Linear Regression without Correspondences

From unique recovery conditions to computation

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \Pi \in \mathcal{P}. \tag{8}$$

	recovery conditions	$\lvert\mathcal{P}\rvert$
Linear Regression	$m \geq n$ & A full rank	1
Linear Regression without Corr.	$m \geq 2n$ & A random	$m!$
Linear Regression without Corr.	$m > n$ & A, x random	$m!$

Good news: m does not increase exponentially as $\lvert\mathcal{P}\rvert$.

question: an intuitive justification for this phenomenon?

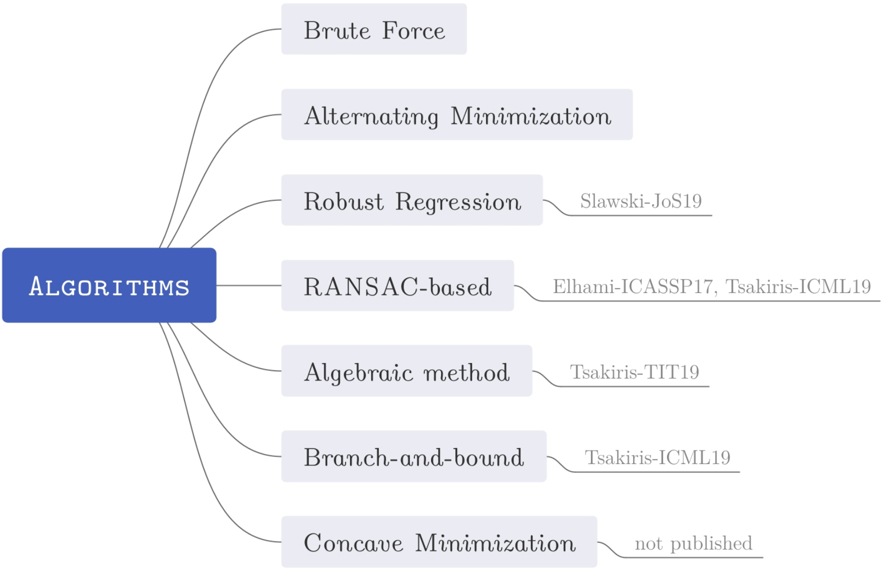
But: does it imply computational easiness?

Theorem (Pananjady-TIT18)

$$\min_{x \in \mathbb{R}^n, \Pi \in \mathcal{P}} \|y - \Pi A x\|_2 \,. \tag{9}$$

There is an $\mathcal{O}(m \log m)$ algorithm (Exercise: find it) to compute (9) when $n = 1$, otherwise (9) is NP-hard to compute.

Linear Regression without Correspondences



$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \tag{10}$$

Brute Force. for each possible permutation $\Pi \in \mathcal{P}$, solve for x the least-squares problem

$$\min_{x \in \mathbb{R}^n} \|y - \Pi A x\|_2, \text{ and?} \tag{11}$$

Linear Regression without Correspondences

Brute Force

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \tag{10}$$

Brute Force. for each possible permutation $\Pi \in \mathcal{P}$, solve for x the least-squares problem

$$\min_{x \in \mathbb{R}^n} \|y - \Pi A x\|_2, \text{ and} \tag{11}$$

take as the output the solution $\hat{x}$ with the smallest error

$$\min_{\Pi \in \mathcal{P}} \|y - \Pi A \hat{x}\|_2 . \tag{12}$$

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n, \tag{10}$$

Brute Force. for each possible permutation $\Pi \in \mathcal{P}$, solve for x the least-squares problem

$$\min_{x \in \mathbb{R}^n} \|y - \Pi A x\|_2, \text{ and} \tag{11}$$

take as the output the solution $\hat{x}$ with the smallest error

$$\min_{\Pi \in \mathcal{P}} \|y - \Pi A \hat{x}\|_2. \tag{12}$$

Exercise: design an algorithm to solve (12) (note that $\hat{x}$ is given), prove its correctness. When is the solution unique?

Linear Regression without Correspondences

Alternating Minimization (EM)

Now we know that:

given x, the optimal Π_x can be computed by your algorithm:

$$\underset{\Pi \in \mathcal{P}}{\operatorname{argmin}} \left\| y - \Pi A x \right\|_2 . \tag{13}$$

given Π, the optimal x_Π can be obtained via least-squares:

$$\underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\| y - \Pi A x \right\|_2 . \tag{14}$$

Now we know that:

given x, the optimal Π_x can be computed by your algorithm:

$$\operatorname*{argmin}_{\Pi \in \mathcal{P}} \|y - \Pi A x\|_2 \,. \tag{13}$$

given Π, the optimal x_Π can be obtained via least-squares:

$$\operatorname*{argmin}_{x \in \mathbb{R}^n} \|y - \Pi A x\|_2 \,. \tag{14}$$

Alternating Minimization. with an initialization $x^{(0)}$, recursively:

solve (13) for $\Pi^{(k)}$ with $x^{(k)}$, and
solve (14) for $x^{(k+1)}$ with $\Pi^{(k)}$, until convergence.

Now we know that:

given x, the optimal Π_x can be computed by your algorithm:

$$\operatorname*{argmin}_{\Pi \in \mathcal{P}} \|y - \Pi A x\|_2 . \tag{13}$$

given Π, the optimal x_Π can be obtained via least-squares:

$$\operatorname*{argmin}_{x \in \mathbb{R}^n} \|y - \Pi A x\|_2 . \tag{14}$$

Alternating Minimization. with an initialization $x^{(0)}$, recursively:

solve (13) for $\Pi^{(k)}$ with $x^{(k)}$, and
solve (14) for $x^{(k+1)}$ with $\Pi^{(k)}$, until convergence.

Pros: (a) low-complexity? (b) easy to implement? (Exercise)

Linear Regression without Correspondences

Alternating Minimization (EM)

Now we know that:

given x, the optimal Π_x can be computed by your algorithm:

$$\operatorname*{argmin}_{\Pi \in \mathcal{P}} \|y - \Pi A x\|_2 . \tag{13}$$

given Π, the optimal x_Π can be obtained via least-squares:

$$\operatorname*{argmin}_{x \in \mathbb{R}^n} \|y - \Pi A x\|_2 . \tag{14}$$

Alternating Minimization. with an initialization $x^{(0)}$, recursively:

solve (13) for $\Pi^{(k)}$ with $x^{(k)}$, and
solve (14) for $x^{(k+1)}$ with $\Pi^{(k)}$, until convergence.

Pros: (a) low-complexity? (b) easy to implement? (Exercise)
Challenges: (a) reliable initialization? (b) theoretical guarantees?

Linear Regression without Correspondences

Robust Regression

difficult to solve in the presence of exponential cardinality of $\mathcal{P}$

$$\min_{x \in \mathbb{R}^n, \Pi \in \mathcal{P}} \|y - \Pi A x\|_2 \,. \tag{15}$$

difficult to solve in the presence of exponential cardinality of $\mathcal{P}$

$$\min_{x\in\mathbb{R}^n, \Pi\in\mathcal{P}} \|y - \Pi A x\|_2 . \tag{15}$$

resort to the subset $\mathcal{S}_r$ of $\mathcal{P}$, $r \leq m$ (partially shuffled data):

$$\mathcal{S}_r := \{\Pi \in \mathcal{P} : d_H(\Pi, I_m) \leq r\}, \tag{16}$$

where $d_H(\Pi, I_m) := |\{i : \Pi_{ii} = 0\}|$.

resort to the subset $\mathcal{S}_r$ of $\mathcal{P}$, $r \leq m$ (partially shuffled data):

$$\mathcal{S}_r := \{\Pi \in \mathcal{P} : d_H(\Pi, I_m) \leq r\}, \tag{15}$$

where $d_H(\Pi, I_m) := |\{i : \Pi_{ii} = 0\}|$.

Important observations:

(a) $\frac{|\mathcal{S}_r|}{|\mathcal{P}|} =$? (Exercise)

resort to the subset $\mathcal{S}_r$ of $\mathcal{P}$, $r \leq m$ (partially shuffled data):

$$\mathcal{S}_r := \{\Pi \in \mathcal{P} : d_H(\Pi, I_m) \leq r\}, \tag{15}$$

where $d_H(\Pi, I_m) := |\{i : \Pi_{ii} = 0\}|$.

Important observations:

(a) $\frac{|\mathcal{S}_r|}{|\mathcal{P}|} =$? (Exercise)

(b) if $\Pi \in \mathcal{S}_r$, $\leq r$ rows and entries of A, y are mismatched

treat mismatches as outliers and the rest as inliers

$\rightsquigarrow$ robust ℓ_0/ℓ_1 norm minimization, e.g., when noiseless:

$$\rightsquigarrow \min_{x \in \mathbb{R}^n} \|y - Ax\|_0 \tag{16}$$

$$\rightsquigarrow \min_{x \in \mathbb{R}^n} \|y - Ax\|_1 . \tag{17}$$

resort to the subset $\mathcal{S}_r$ of $\mathcal{P}$, $r \leq m$ (partially shuffled data):

$$\mathcal{S}_r := \{\Pi \in \mathcal{P} : d_H(\Pi, I_m) \leq r\}, \tag{15}$$

where $d_H(\Pi, I_m) := |\{i : \Pi_{ii} = 0\}|$.

Important observations:

(a) $\frac{|\mathcal{S}_r|}{|\mathcal{P}|} =$? (Exercise)

(b) if $\Pi \in \mathcal{S}_r$, $\leq r$ rows and entries of A, y are mismatched

treat mismatches as outliers and the rest as inliers

$\rightsquigarrow$ robust ℓ_0/ℓ_1 norm minimization, e.g., when noiseless:

$$\rightsquigarrow \min_{x \in \mathbb{R}^n} \|y - Ax\|_0 \tag{16}$$

$$\rightsquigarrow \min_{x \in \mathbb{R}^n} \|y - Ax\|_1 . \tag{17}$$

question: why this formulation is a good choice?

Linear Regression without Correspondences

Robust Regression

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{18}$$

$$\mathcal{S}_r = \{\Pi \in \mathcal{P} : d_H(\Pi, I_m) \leq r\} \tag{19}$$

ℓ_1 norm minimization (noiseless): (Candes-TIT05)

$$\min_{x \in \mathbb{R}^n} \|y - Ax\|_1 \tag{20}$$

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{18}$$

$$\mathcal{S}_r = \{\Pi \in \mathcal{P} : d_H(\Pi, I_m) \leq r\} \tag{19}$$

ℓ_1 norm minimization (noiseless): (Candes-TIT05)

$$\min_{x \in \mathbb{R}^n} \|y - Ax\|_1 \tag{20}$$

ℓ_1 norm minimization (noisy):

$$(\hat{x}, \hat{e}) = \operatorname*{argmin}_{x \in \mathbb{R}^n, e \in \mathbb{R}^m} \frac{1}{m} \left\| y - Ax - \sqrt{m} e \right\|_2^2 + \lambda \|e\|_1, \lambda > 0. \tag{21}$$

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{18}$$

$$\mathcal{S}_r = \{\Pi \in \mathcal{P} : d_H(\Pi, I_m) \leq r\} \tag{19}$$

ℓ_1 norm minimization (noiseless): (Candes-TIT05)

$$\min_{x \in \mathbb{R}^n} \|y - Ax\|_1 \tag{20}$$

ℓ_1 norm minimization (noisy):

$$(\hat{x}, \hat{e}) = \operatorname*{argmin}_{x \in \mathbb{R}^n, e \in \mathbb{R}^m} \frac{1}{m} \left\| y - Ax - \sqrt{m} e \right\|_2^2 + \lambda \|e\|_1, \lambda > 0. \tag{21}$$

Theorem (Slawski-JoS19, informal)

$\|\hat{x} - x^*\|_2 \leq c_1 + r c_2$ *w.h.p. for some* λ *if* $r \leq c_3 \frac{m-n}{\log(m/r)}$. c_1, c_2 *are well-behaved data-dependent quantities and* c_3 *constant.*

Linear Regression without Correspondences

Robust Regression

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{18}$$

$$\mathcal{S}_r = \{\Pi \in \mathcal{P} : d_H(\Pi, I_m) \le r\} \tag{19}$$

ℓ_1 norm minimization (noiseless): (Candes-TIT05)

$$\min_{x \in \mathbb{R}^n} \|y - Ax\|_1 \tag{20}$$

ℓ_1 norm minimization (noisy):

$$(\hat{x}, \hat{e}) = \operatorname*{argmin}_{x \in \mathbb{R}^n, e \in \mathbb{R}^m} \frac{1}{m} \left\|y - Ax - \sqrt{m} e\right\|_2^2 + \lambda \|e\|_1, \lambda > 0. \tag{21}$$

Theorem (Slawski-JoS19, informal)

$\|\hat{x} - x^*\|_2 \le c_1 + r c_2$ *w.h.p. for some* λ *if* $r \le c_3 \frac{m-n}{\log(m/r)}$. c_1, c_2 *are well-behaved data-dependent quantities and* c_3 *constant.*

Experimentally this method fails if $r/m > 0.5$

$\rightsquigarrow$ algorithms for more than 50% shuffled data?

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n. \tag{22}$$

RANSAC?

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n. \tag{22}$$

RANSAC?

(a) Take a subvector $\bar{y} \in \mathbb{R}^n$ of y, and compute

$$x_i = \operatorname*{argmin}_x \|\bar{y} - A_i x\|_2 \text{ (by least squares)} \tag{23}$$

for all possible A_i, where A_i is a $n \times n$ submatrix of A.

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n. \quad (22)$$

RANSAC?

(a) Take a subvector $\bar{y} \in \mathbb{R}^n$ of y, and compute

$$x_i = \operatorname*{argmin}_x \|\bar{y} - A_i x\|_2 \text{ (by least squares)} \quad (23)$$

for all possible A_i, where A_i is a $n \times n$ submatrix of A.

(b) Take as the output the solution x_i with the smallest error

$$\min_{\Pi \in \mathcal{P}} \|y - \Pi A x_i\|_2 \text{ (by your algorithm)}. \quad (24)$$

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n. \tag{22}$$

RANSAC?

(a) Take a subvector $\bar{y} \in \mathbb{R}^n$ of y, and compute

$$x_i = \operatorname*{argmin}_x \|\bar{y} - A_i x\|_2 \text{ (by least squares)} \tag{23}$$

for all possible A_i, where A_i is a $n \times n$ submatrix of A.

(b) Take as the output the solution x_i with the smallest error

$$\min_{\Pi \in \mathcal{P}} \|y - \Pi A x_i\|_2 \text{ (by your algorithm)}. \tag{24}$$

Exercises: (a) the complexity of this algorithm? (b) compare it to **Brute Force** (c) describe this algorithm using the language of RANSAC (d) implement this algorithm.

Linear Regression without Correspondences

The algebraic method

We assume $m > 1$.

Let $q(z) := q(z_1, \ldots, z_m) = \sum_{i=1}^{m} z_i^i$ be a polynomial.

Question: under what conditions on Π we have for any $z \in \mathbb{R}^n$

$$q(z) = q(\Pi z)? \tag{25}$$

Linear Regression without Correspondences

The algebraic method

We assume $m > 1$.

Let $q(z) := q(z_1, \ldots, z_m) = \sum_{i=1}^{m} z_i^i$ be a polynomial.

Question: under what conditions on Π we have for any $z \in \mathbb{R}^n$

$$q(z) = q(\Pi z)? \tag{25}$$

Answer: the only possibility is $\Pi = I_m$.

We assume $m > 1$.

Let $q(z) := q(z_1, \ldots, z_m) = \sum_{i=1}^{m} z_i^i$ be a polynomial.

Let $p_k(z) = \sum_{i=1}^{m} z_i^k$ be a polynomial.

Question: under what conditions on Π we have for any $y \in \mathbb{R}^n$

$$p_k(z) = p_k(\Pi z)? \tag{26}$$

We assume $m > 1$.

Let $q(z) := q(z_1, \ldots, z_m) = \sum_{i=1}^{m} z_i^i$ be a polynomial.

Let $p_k(z) = \sum_{i=1}^{m} z_i^k$ be a polynomial.

Question: under what conditions on Π we have for any $y \in \mathbb{R}^n$

$$p_k(z) = p_k(\Pi z)? \tag{26}$$

Answer: Π can be any permutation.

Linear Regression without Correspondences

The algebraic method

We assume $m > 1$.

Let $q(z) := q(z_1, \ldots, z_m) = \sum_{i=1}^{m} z_i^i$ be a polynomial.

Let $p_k(z) = \sum_{i=1}^{m} z_i^k$ be a polynomial.

Definition (symmetric polynomials)

A polynomial f in m variables satisfying $f(z) = f(\Pi z)$ for any $z \in \mathbb{R}^m$ and $\Pi \in \mathcal{P}$ is called *symmetric*. Note that p_k is symmetric.

Linear Regression without Correspondences

The algebraic method

We assume $m > 1$.

Let $q(z) := q(z_1, \ldots, z_m) = \sum_{i=1}^{m} z_i^i$ be a polynomial.

Let $p_k(z) = \sum_{i=1}^{m} z_i^k$ be a polynomial.

Definition (symmetric polynomials)

A polynomial f in m variables satisfying $f(z) = f(\Pi z)$ for any $z \in \mathbb{R}^m$ and $\Pi \in \mathcal{P}$ is called *symmetric*. Note that p_k is symmetric.

Remark: the polynomial p_k is symmetric for any $k > 0$, but q not.

Linear Regression without Correspondences

The algebraic method: the noiseless case

$$y = \Pi A x^*, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{27}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{28}$$

Given a symmetric polynomial f in m variables, we have

$$f(Ax^*) = f(\Pi A x^*) = f(y). \tag{29}$$

Linear Regression without Correspondences

The algebraic method: the noiseless case

$$y = \Pi A x^*, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{27}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \dots, n\} =: [n] \tag{28}$$

Given a symmetric polynomial f in m variables, we have

$$f(Ax^*) = f(\Pi A x^*) = f(y). \tag{29}$$

Hence the parameter vector x^* satisfies:

$$p_k(Ax^*) - p_k(y) = 0, k \in [n]. \tag{30}$$

Linear Regression without Correspondences

The algebraic method: the noiseless case

$$y = \Pi A x^*, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{27}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{28}$$

Given a symmetric polynomial f in m variables, we have

$$f(Ax^*) = f(\Pi A x^*) = f(y). \tag{29}$$

Hence the parameter vector x^* satisfies:

$$p_k(Ax^*) - p_k(y) = 0, k \in [n]. \tag{30}$$

It suggests to solve for x the polynomial system:

$$p_k(Ax) - p_k(y) = 0, k \in [n], \tag{31}$$

to which x^* has to be a solution.

Linear Regression without Correspondences

The algebraic method: the noiseless case

$$y = \Pi A x^*, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{27}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{28}$$

Given a symmetric polynomial f in m variables, we have

$$f(Ax^*) = f(\Pi A x^*) = f(y). \tag{29}$$

Hence the parameter vector x^* satisfies:

$$p_k(Ax^*) - p_k(y) = 0, k \in [n]. \tag{30}$$

It suggests to solve for x the polynomial system:

$$p_k(Ax) - p_k(y) = 0, k \in [n], \tag{31}$$

to which x^* has to be a solution.

Theorem (Tsakiris-TIT19)

With the model (27), the polynomial system (31) has l complex solutions with $1 \leq l \leq n!$ if A is random.

Linear Regression without Correspondences

The algebraic method: the noisy case

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{32}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{33}$$

ideally we would like to solve for x the polynomial system

$$p_k(Ax) - p_k(y - \epsilon) = 0, k \in [n], \tag{34}$$

that x^* satisfies, but we have no access to ϵ.

Linear Regression without Correspondences

The algebraic method: the noisy case

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{32}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{33}$$

ideally we would like to solve for x the polynomial system

$$p_k(Ax) - p_k(y - \epsilon) = 0, k \in [n], \tag{34}$$

that x^* satisfies, but we have no access to ϵ.
Instead, for finding an suitable estimate, we dare to solve

$$p_k(Ax) - p_k(y) = 0, k \in [n], \tag{35}$$

Linear Regression without Correspondences

The algebraic method: the noisy case

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x^* \in \mathbb{R}^n \tag{32}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{33}$$

ideally we would like to solve for x the polynomial system

$$p_k(Ax) - p_k(y - \epsilon) = 0, k \in [n], \tag{34}$$

that x^* satisfies, but we have no access to ϵ.
Instead, for finding an suitable estimate, we dare to solve

$$p_k(Ax) - p_k(y) = 0, k \in [n], \text{ because} \tag{35}$$

Theorem (Tsakiris-TIT19)

With the model (36), the polynomial system has l complex solutions with $1 \leq l \leq n!$ if A is random.

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x \in \mathbb{R}^n \tag{36}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{37}$$

The Algorithm.

(a) solve the polynomial system

$$p_k(Ax) - p_k(y) = 0, k \in [n], \tag{38}$$

which gives l complex roots $x_1, \ldots, x_l$ $(1 \leq l \leq n!)$.

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x \in \mathbb{R}^n \tag{36}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{37}$$

The Algorithm.

(a) solve the polynomial system

$$p_k(Ax) - p_k(y) = 0, k \in [n], \tag{38}$$

which gives l complex roots $x_1, \ldots, x_l$ $(1 \leq l \leq n!)$.

(b) Let $x^{(0)}$ be the real part $(x_i)_{\mathbb{R}}$ of x_i that causes the smallest error $\min_{\Pi \in \mathcal{P}} \|y - \Pi A (x_i)_{\mathbb{R}}\|_2$ among all roots.

$$y = \Pi A x^* + \epsilon, \text{ with } y \in \mathbb{R}^m, x \in \mathbb{R}^n \tag{36}$$

$$p_k(z) = \sum_{i=1}^{m} z_i^k, k \in \{1, 2, \ldots, n\} =: [n] \tag{37}$$

The Algorithm.

(a) solve the polynomial system

$$p_k(Ax) - p_k(y) = 0, k \in [n], \tag{38}$$

which gives l complex roots $x_1, \ldots, x_l$ $(1 \leq l \leq n!)$.

(b) Let $x^{(0)}$ be the real part $(x_i)_{\mathbb{R}}$ of x_i that causes the smallest error $\min_{\Pi \in \mathcal{P}} \|y - \Pi A (x_i)_{\mathbb{R}}\|_2$ among all roots.

(c) run the alternating minimization algorithm initialized by $x^{(0)}$ to produce the final estimate.

The branch-and-bound technique can be employed to compute the global solution of the NP-hard problem

$$\min_{x \in \mathbb{R}^n, \Pi \in \mathcal{P}} \|y - \Pi A x\|_2 = \min_{x \in \mathbb{R}^n} g(x), \tag{39}$$

where $g(x) = \min_{\Pi} \|y - \Pi A x\|_2$.

$g(x)$ can be considered as a continuous **black-box** function:

we have only access to the values of g

The branch-and-bound technique can be employed to compute the global solution of the NP-hard problem

$$\min_{x\in\mathbb{R}^n,\Pi\in\mathcal{P}} \|y-\Pi Ax\|_2 = \min_{x\in\mathbb{R}^n} g(x), \tag{39}$$

where $g(x) = \min_\Pi \|y-\Pi Ax\|_2$.

$g(x)$ can be considered as a continuous **black-box** function:

we have only access to the values of g

we study this algorithm on the board.

brute force : only for educational purpose

alt. min. : sensitive to initialization, no theoretical results

ℓ_1 **norm min.** : scalable, only for $\leq 50\%$ shuffled data

RANSAC : limited $n \leq 4$.

algebraic method : limited in $n \leq 6$.

branch-and-bound : limited in $n \leq 4$

concave min. : limited in $n \leq 8$

Linear Regression without Correspondences

Recap.

we start by discussing whether the problem is well-posed or not,

is it solvable? under what conditions if yes?

Linear Regression without Correspondences

Recap.

we start by discussing whether the problem is well-posed or not,

is it solvable? under what conditions if yes?

then the hardness of the problem,

Linear Regression without Correspondences

Recap.

we start by discussing whether the problem is well-posed or not,

is it solvable? <u>under what conditions</u> if yes?

then the hardness of the problem,

and finally the algorithms.

Future Research

Can we do better for this problem?

what do you learn for your research?

(a) **General Idea.**

1. $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$?

   f is a function encountered in your research

(b) **Deep Learning.**

1. Learning permutation invariance via symmetric polynomials?
   computer vision flavor,
   can we produce some permutation-invariant features?

(b) **Deep Learning.**

2. Learning to select samples ($A_i \in \mathbb{R}^{n \times n}$) in RANSAC?
   reinforcement learning flavor

(b) **Deep Learning.**

3. Permutation learning?
   perform least-squares with the learned permutation

(c) **Deep Learning.**

1. Learning to identify shuffled and unshuffled data?

   the simplest case: train a 0-1 classifier

   $\rightsquigarrow$ remove the identified outliers, perform least-squares

(c) **Deep Learning.**

2. Learning to reweight shuffled and unshuffled data?
   attention mechanism
   $\rightsquigarrow$ wish: the learned weights are small for shuffled data
   $\rightsquigarrow$ perform least-squares with the learned weight.

(d) **Algorithmic Ideas.**

Any robust regression algorithms are applicable to our problem

shuffled data as outliers and the rest as inliers

$\rightsquigarrow$ new theoretical guarantees should:

$\rightsquigarrow\rightsquigarrow$ take into account permutation constraints

(e) **Theoretical Development?**

There are a lot.

need some background to introduce them.

(f) **New Topic?**

Can Subspace Learning be robust to permutation corruptions?

It is underway.