Bilinear Regression via Convex Programming without Lifting

Sohail Bahmani
Georgia Tech

iTWIST'18
CIRM, Marseille

Nov 23, 2018

Bilinear regression

Given $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \in \mathbb{R}^{d_1}$ and $\boldsymbol{a}'_1, \ldots, \boldsymbol{a}'_n \in \mathbb{R}^{d_2}$ we observe noisy *bilinear* measurements of unknown vectors $\boldsymbol{x}_\star \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}'_\star \in \mathbb{R}^{d_2}$ as

$$
\begin{aligned}
y_1 &= \boldsymbol{a}_1^\top \boldsymbol{x}_\star \, \boldsymbol{x}'^\top_\star \boldsymbol{a}'_1 + \xi_1 \\
y_2 &= \boldsymbol{a}_2^\top \boldsymbol{x}_\star \, \boldsymbol{x}'^\top_\star \boldsymbol{a}'_2 + \xi_2 \\
&\;\;\vdots \qquad\quad \vdots \\
y_n &= \boldsymbol{a}_n^\top \boldsymbol{x}_\star \, \boldsymbol{x}'^\top_\star \boldsymbol{a}'_n + \underbrace{\xi_n}_{\text{noise}} .
\end{aligned}
$$

Problem

Estimate $\boldsymbol{x}_\star$ and $\boldsymbol{x}'_\star$ accurately, using the observations $(\boldsymbol{a}_i, \boldsymbol{a}'_i, y_i)$, $i = 1, \ldots, n$.

A much more general **nonlinear regression** will be addressed towards the end.

Some reminders

Scaling ambiguity:
$(\boldsymbol{x}, \boldsymbol{x}')$ is an accurate solution $\implies$ $(t\boldsymbol{x}, t^{-1}\boldsymbol{x}')$ is an accurate solution $\forall t \neq 0$.

Computation:
There are computationally hard instances. Randomness helps to avoid them.

Examples

In matrix notation,

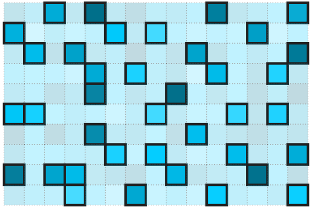
$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_\star \circ \boldsymbol{A}'\boldsymbol{x}'_\star + \boldsymbol{\xi}\,,$$

where $\boldsymbol{A} = [\boldsymbol{a}_1 \;\cdots\; \boldsymbol{a}_n]^\top$ and $\boldsymbol{A}' = [\boldsymbol{a}'_1 \;\cdots\; \boldsymbol{a}'_n]^\top$.

Matrix completion:

$\boldsymbol{a}_i = \boldsymbol{e}_{r_i}$ and $\boldsymbol{a}'_i = \boldsymbol{e}_{c_i}$ are random coordinate indicator vectors

$X = x_\star x'^{\top}_\star$



$y_i = X_{r_i, c_i} = a_{r_i}^\top X a'_{c_i}$

Examples

In matrix notation,

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_\star \circ \boldsymbol{A}'\boldsymbol{x}'_\star + \boldsymbol{\xi}\,,$$

where $\boldsymbol{A} = [\boldsymbol{a}_1 \;\cdots\; \boldsymbol{a}_n]^\top$ and $\boldsymbol{A}' = [\boldsymbol{a}'_1 \;\cdots\; \boldsymbol{a}'_n]^\top$.

Blind deconvolution:

$\boldsymbol{A}$ and $\boldsymbol{A}'$ interpreted as subspaces in the *Fourier* domain (ignoring $\mathbb{R}$ vs. $\mathbb{C}$)

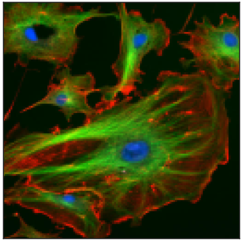
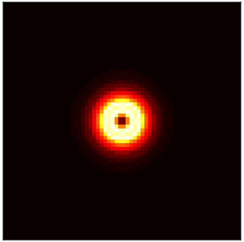
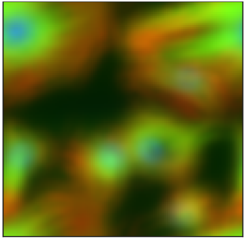


image: $\boldsymbol{F}^{-1}\boldsymbol{A}\boldsymbol{x}_\star$



PSF: $\boldsymbol{F}^{-1}\boldsymbol{A}'\boldsymbol{x}'_\star$



blurred image: $\boldsymbol{F}^{-1}\boldsymbol{y}$

Related work ...

SDP-relaxation[*,†]: The conversion $\boldsymbol{x}\boldsymbol{x}'^{\top} \mapsto \boldsymbol{X}$ maps the bilinear regression to a *matrix linear regression*

$$y_i = \boldsymbol{a}_i^{\top} \boldsymbol{x}_\star \boldsymbol{x}_\star'^{\top} \boldsymbol{a}_i' + \xi_i = \boldsymbol{a}_i^{\top} \boldsymbol{X}_\star \boldsymbol{a}_i' + \xi_i \,.$$

Estimate the rank one solution through *nuclear norm* minimization

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{X}} \; \|\boldsymbol{X}\|_* \\ &\text{subject to } \boldsymbol{a}_i^{\top} \boldsymbol{X} \boldsymbol{a}_i' = y_i \,, \quad i = 1, \dots, n \,. \end{aligned}$$

Not scalable, due to the prohibitive cost of SDP.

[*] Ahmed et al, "Blind deconvolution using convex programming," *IEEE Trans. Info. Theory*, 2014

[†] Cai & Zhang, "ROP: Matrix recovery via rank-one projections," Annals of Statistics, 2015

Related work ...

SDP-relaxation[*,†]: The conversion $\boldsymbol{x}\boldsymbol{x}'^{\top} \mapsto \boldsymbol{X}$ maps the bilinear regression to a *matrix linear regression*

$$y_i = \boldsymbol{a}_i^{\top} \boldsymbol{x}_{\star} \boldsymbol{x}_{\star}'^{\top} \boldsymbol{a}_i' + \xi_i = \boldsymbol{a}_i^{\top} \boldsymbol{X}_{\star} \boldsymbol{a}_i' + \xi_i \,.$$

Estimate the rank one solution through *nuclear norm* minimization

$$\begin{aligned} &\underset{\boldsymbol{X}}{\operatorname{argmin}} \ \|\boldsymbol{X}\|_* \\ &\text{subject to } \boldsymbol{a}_i^{\top} \boldsymbol{X} \boldsymbol{a}_i' = y_i \,, \quad i = 1, \dots, n \,. \end{aligned}$$

Not scalable, due to the prohibitive cost of SDP.

[*] Ahmed et al, "Blind deconvolution using convex programming," *IEEE Trans. Info. Theory*, 2014
[†] Cai & Zhang, "ROP: Matrix recovery via rank-one projections," Annals of Statistics, 2015

Related work ...

Nonconvex gradient descent[*,†]: Similar to the *Wirtinger flow algorithm* for phase retrieval, run gradient descent on the residual error

$$f(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{n} \sum_{i=1}^{n} \left| \boldsymbol{a}_i^* \boldsymbol{x} \boldsymbol{x}'^* \boldsymbol{a}_i' - y_i \right|^2 .$$

Regularized variants are proposed for blind deconvolution, blind calibration, & matrix completion.

Light-tailed distribution is needed for iteration analysis.

Analyses are often lengthy.

[*] Cambareri, Jacques, "Through the haze: A non-convex approach to blind gain calibration for linear random sensing models," Information & Inference, 2018.

[†] Li et al, "Rapid, robust, and reliable blind deconvolution via nonconvex optimization," *App. & Comp. Harmonic Analysis*, 2018.

Related work ...

Nonconvex gradient descent[*,†]: Similar to the *Wirtinger flow algorithm* for phase retrieval, run gradient descent on the residual error

$$f(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{n} \sum_{i=1}^{n} |\boldsymbol{a}_i^* \boldsymbol{x} \boldsymbol{x}'^* \boldsymbol{a}_i' - y_i|^2 .$$

Regularized variants are proposed for blind deconvolution, blind calibration, & matrix completion.

Light-tailed distribution is needed for iteration analysis.

Analyses are often lengthy.

[*] Cambareri, Jacques, "Through the haze: A non-convex approach to blind gain calibration for linear random sensing models," Information & Inference, 2018.

[†] Li et al, "Rapid, robust, and reliable blind deconvolution via nonconvex optimization," *App. & Comp. Harmonic Analysis*, 2018.

Related work

BranchHull*: With $\boldsymbol{s} = \operatorname{sgn}(\boldsymbol{A}\boldsymbol{x}_\star)$ given, solve the **convex program**

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^{d_1},\boldsymbol{x}'\in\mathbb{R}^{d_2}}{\operatorname{argmin}} \quad & \|\boldsymbol{A}\boldsymbol{x}\|_2^2 + \|\boldsymbol{A}'\boldsymbol{x}'\|_2^2 \\ \text{subject to } & \boldsymbol{s} \circ (\boldsymbol{A}\boldsymbol{x}) \circ (\boldsymbol{A}'\boldsymbol{x}') \geq |\boldsymbol{y}| \\ & \boldsymbol{s} \circ (\boldsymbol{A}\boldsymbol{x}) \geq 0 \,. \end{aligned}$$

The sample complexity $n \gtrsim d_1 + d_2$ is shown for Gaussian $\boldsymbol{A}$ and $\boldsymbol{A}'$

Knowing $\boldsymbol{s}$ is a **restrictive assumption**.

The relaxation seem **very sensitive to noise**.

*Aghasi et al, "BranchHull: Convex bilinear inversion from the entrywise product of signals with known signs," arXiv:1702.04342, 2017

Related work

BranchHull*: With $\boldsymbol{s} = \operatorname{sgn}(\boldsymbol{A}\boldsymbol{x}_\star)$ given, solve the **convex program**

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^{d_1},\boldsymbol{x}'\in\mathbb{R}^{d_2}}{\operatorname{argmin}} \quad & \|\boldsymbol{A}\boldsymbol{x}\|_2^2 + \|\boldsymbol{A}'\boldsymbol{x}'\|_2^2 \\ \text{subject to } & \boldsymbol{s}\circ(\boldsymbol{A}\boldsymbol{x})\circ(\boldsymbol{A}'\boldsymbol{x}') \geq |\boldsymbol{y}| \\ & \boldsymbol{s}\circ(\boldsymbol{A}\boldsymbol{x}) \geq 0\,. \end{aligned}$$

The sample complexity $n \gtrsim d_1 + d_2$ is shown for Gaussian $\boldsymbol{A}$ and $\boldsymbol{A}'$

Knowing $\boldsymbol{s}$ is a **restrictive assumption**.
The relaxation seem **very sensitive to noise**.

*Aghasi et al, "BranchHull: Convex bilinear inversion from the entrywise product of signals with known signs," `arXiv:1702.04342`, 2017

Our proposed convex relaxation

Given $\boldsymbol{x}_0 \approx \boldsymbol{x}_\star$ and $\boldsymbol{x}'_0 \approx \boldsymbol{x}'_\star$ such that for a sufficiently small $\epsilon \geq 0$,

$$\left\| \begin{pmatrix} \boldsymbol{x}_0 - \boldsymbol{x}_\star \\ \boldsymbol{x}'_0 - \boldsymbol{x}'_\star \end{pmatrix} \right\|_2 \leq \epsilon \left\| \begin{pmatrix} \boldsymbol{x}_\star \\ \boldsymbol{x}'_\star \end{pmatrix} \right\|_2 ,$$

we estimate $\boldsymbol{x}_\star$ and $\boldsymbol{x}'_\star$ through the convex program:

$$(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{x}}') = \underset{\boldsymbol{x}, \boldsymbol{x}'}{\operatorname{argmax}}\ \boldsymbol{x}_0^\top \boldsymbol{x} + \boldsymbol{x}_0'^\top \boldsymbol{x}' - \frac{2}{n} \sum_{i=1}^{n} \max\left\{ \frac{1}{4} (\boldsymbol{a}_i^\top \boldsymbol{x} + \boldsymbol{a}_i'^\top \boldsymbol{x}')^2 - y_i, \frac{1}{4} (\boldsymbol{a}_i^\top \boldsymbol{x} - \boldsymbol{a}_i'^\top \boldsymbol{x}')^2 \right\}$$

Our proposed convex relaxation

Given $\boldsymbol{x}_0 \approx \boldsymbol{x}_\star$ and $\boldsymbol{x}'_0 \approx \boldsymbol{x}'_\star$ such that for a sufficiently small $\epsilon \geq 0$,

$$\left\| \begin{pmatrix} \boldsymbol{x}_0 - \boldsymbol{x}_\star \\ \boldsymbol{x}'_0 - \boldsymbol{x}'_\star \end{pmatrix} \right\|_2 \leq \epsilon \left\| \begin{pmatrix} \boldsymbol{x}_\star \\ \boldsymbol{x}'_\star \end{pmatrix} \right\|_2 ,$$

we estimate $\boldsymbol{x}_\star$ and $\boldsymbol{x}'_\star$ through the convex program:

$$(\widehat{\boldsymbol{x}},\widehat{\boldsymbol{x}}') = \operatorname*{argmax}_{\boldsymbol{x},\boldsymbol{x}'} \boldsymbol{x}_0^\mathsf{T}\boldsymbol{x} + \boldsymbol{x}_0'^\mathsf{T}\boldsymbol{x}' - \frac{2}{n}\sum_{i=1}^{n} \max\left\{\frac{1}{4}(\boldsymbol{a}_i^\mathsf{T}\boldsymbol{x} + \boldsymbol{a}_i'^\mathsf{T}\boldsymbol{x}')^2 - y_i, \frac{1}{4}(\boldsymbol{a}_i^\mathsf{T}\boldsymbol{x} - \boldsymbol{a}_i'^\mathsf{T}\boldsymbol{x}')^2\right\}$$

Intuition: Using the identities

$$2\max\{u, v\} = |u - v| + u + v\,, \qquad \begin{aligned}(u+v)^2 - (u-v)^2 &= 4uv \\ (u+v)^2 + (u-v)^2 &= 2(u^2+v^2)\end{aligned}\,,$$

the objective can be written as

$$\boldsymbol{x}_0^\mathsf{T}\boldsymbol{x} + \boldsymbol{x}_0'^\mathsf{T}\boldsymbol{x}' - \frac{1}{2n}\sum_{i=1}^{n}(\boldsymbol{a}_i^\mathsf{T}\boldsymbol{x})^2 + (\boldsymbol{a}_i'^\mathsf{T}\boldsymbol{x}')^2 - \frac{1}{n}\sum_{i=1}^{n}\left|\boldsymbol{a}_i^\mathsf{T}\boldsymbol{x}\,\boldsymbol{x}'^\mathsf{T}\boldsymbol{a}_i' - y_i\right|$$

Our proposed convex relaxation

Given $\boldsymbol{x}_0 \approx \boldsymbol{x}_\star$ and $\boldsymbol{x}'_0 \approx \boldsymbol{x}'_\star$ such that for a sufficiently small $\epsilon \geq 0$,

$$\left\| \begin{pmatrix} \boldsymbol{x}_0 - \boldsymbol{x}_\star \\ \boldsymbol{x}'_0 - \boldsymbol{x}'_\star \end{pmatrix} \right\|_2 \leq \epsilon \left\| \begin{pmatrix} \boldsymbol{x}_\star \\ \boldsymbol{x}'_\star \end{pmatrix} \right\|_2 ,$$

we estimate $\boldsymbol{x}_\star$ and $\boldsymbol{x}'_\star$ through the convex program:

$$(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{x}}') = \underset{\boldsymbol{x}, \boldsymbol{x}'}{\operatorname{argmax}}\ \boldsymbol{x}_0^\top \boldsymbol{x} + \boldsymbol{x}_0'^\top \boldsymbol{x}' - \frac{2}{n} \sum_{i=1}^{n} \max\left\{ \frac{1}{4} (\boldsymbol{a}_i^\top \boldsymbol{x} + \boldsymbol{a}_i'^\top \boldsymbol{x}')^2 - y_i , \frac{1}{4} (\boldsymbol{a}_i^\top \boldsymbol{x} - \boldsymbol{a}_i'^\top \boldsymbol{x}')^2 \right\}$$

Equivalent folmulation: Quadratically Constrained Linear Minimization

$$\begin{aligned} \underset{\boldsymbol{x}, \boldsymbol{x}'}{\operatorname{argmax}}\ \max_{\boldsymbol{w}}\ & \boldsymbol{x}_0^\top \boldsymbol{x} + \boldsymbol{x}_0'^\top \boldsymbol{x}' - \frac{2}{n} \mathbf{1}^\top \boldsymbol{w} \\ \text{subject to}\ & \frac{1}{4} (\boldsymbol{a}_i^\top \boldsymbol{x} + \boldsymbol{a}_i'^\top \boldsymbol{x}')^2 - y_i \leq w_i , && i = 1, \ldots, n \\ & \frac{1}{4} (\boldsymbol{a}_i^\top \boldsymbol{x} - \boldsymbol{a}_i'^\top \boldsymbol{x}')^2 \leq w_i , && i = 1, \ldots, n . \end{aligned}$$

Our proposed convex relaxation

Given $\mathbf{x}_0 \approx \mathbf{x}_\star$ and $\mathbf{x}'_0 \approx \mathbf{x}'_\star$ such that for a sufficiently small $\epsilon \geq 0$,

$$\left\| \begin{pmatrix} \mathbf{x}_0 - \mathbf{x}_\star \\ \mathbf{x}'_0 - \mathbf{x}'_\star \end{pmatrix} \right\|_2 \leq \epsilon \left\| \begin{pmatrix} \mathbf{x}_\star \\ \mathbf{x}'_\star \end{pmatrix} \right\|_2 ,$$

we estimate $\mathbf{x}_\star$ and $\mathbf{x}'_\star$ through the convex program:

$$(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}') = \underset{\mathbf{x},\mathbf{x}'}{\operatorname{argmax}}\ \mathbf{x}_0^\top \mathbf{x} + \mathbf{x}_0'^\top \mathbf{x}' - \frac{2}{n} \sum_{i=1}^{n} \max\left\{ \frac{1}{4}(\mathbf{a}_i^\top \mathbf{x} + \mathbf{a}_i'^\top \mathbf{x}')^2 - y_i, \frac{1}{4}(\mathbf{a}_i^\top \mathbf{x} - \mathbf{a}_i'^\top \mathbf{x}')^2 \right\}$$

Accelerated first-order methods: We can use a *smoothed* formulation,

$$\underset{\mathbf{x},\mathbf{x}'}{\operatorname{argmax}}\ \mathbf{x}_0^\top \mathbf{x} + \mathbf{x}_0'^\top \mathbf{x}' - \frac{2}{\mu n} \sum_{i=1}^{n} \log\left(e^{\frac{\mu}{4}\left(\mathbf{a}_i^\top \mathbf{x} + \mathbf{a}_i'^\top \mathbf{x}'\right)^2 - \mu y_i} + e^{\frac{\mu}{4}\left(\mathbf{a}_i^\top \mathbf{x} - \mathbf{a}_i'^\top \mathbf{x}'\right)^2} \right)$$

Theoretical guarantee*

We consider $\boldsymbol{a}_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(\mathbf{0}, \boldsymbol{I}_{d_1 \times d_1})$ and $\boldsymbol{a}'_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(\mathbf{0}, \boldsymbol{I}_{d_2 \times d_2})$, and $\boldsymbol{\xi} = \mathbf{0}$.

*Bahmani, "Estimation from non-linear observations via convex programming with application to bilinear regression," `arXiv:1806.07307`, 2018.

Theoretical guarantee*

We consider $\boldsymbol{a}_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(\mathbf{0}, \boldsymbol{I}_{d_1 \times d_1})$ and $\boldsymbol{a}'_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(\mathbf{0}, \boldsymbol{I}_{d_2 \times d_2})$, and $\boldsymbol{\xi} = \mathbf{0}$.

"Spectral" method of choosing $\boldsymbol{x}_0$ and $\boldsymbol{x}'_0$:

Let $\boldsymbol{S}_n = n^{-1} \sum_{i=1}^{n} y_i \boldsymbol{a}_i \boldsymbol{a}_i'^{\top}$ which satisfies $\mathbb{E}\boldsymbol{S}_n = \boldsymbol{x}_\star \boldsymbol{x}_\star'^{\top}$. Then we choose,

$$\boldsymbol{x}_0 = \|\boldsymbol{S}_n\|^{1/2} \boldsymbol{u}_{\max}(\boldsymbol{S}_n) \,, \qquad \boldsymbol{x}'_0 = \|\boldsymbol{S}_n\|^{1/2} \boldsymbol{v}_{\max}(\boldsymbol{S}_n) \,.$$

that meet the required ϵ relative error if $n \overset{\epsilon}{\gtrsim} (d_1 + d_2) \log (d_1 + d_2)$.

*Bahmani, "Estimation from non-linear observations via convex programming with application to bilinear regression," `arXiv:1806.07307`, 2018.

Theoretical guarantee*

We consider $\boldsymbol{a}_i \overset{\text{i.i.d.}}{\sim} \mathrm{Normal}(\mathbf{0}, \boldsymbol{I}_{d_1 \times d_1})$ and $\boldsymbol{a}'_i \overset{\text{i.i.d.}}{\sim} \mathrm{Normal}(\mathbf{0}, \boldsymbol{I}_{d_2 \times d_2})$, and $\boldsymbol{\xi} = \mathbf{0}$.

Theorem

For a sufficiently large absolute constant $C > 0$, if

$$n \geq C \max\left\{d_1 + d_2, \log\left(8/\delta\right)\right\} \ ,$$

then with probability $\geq 1 - \delta$ the estimates are exact up to the scaling ambiguity, i.e., for some $t \neq 0$, we have $\widehat{\boldsymbol{x}} = t\boldsymbol{x}_\star$ and $\widehat{\boldsymbol{x}}' = t^{-1}\boldsymbol{x}'_\star$.

*Bahmani, "Estimation from non-linear observations via convex programming with application to bilinear regression," `arXiv:1806.07307`, 2018.

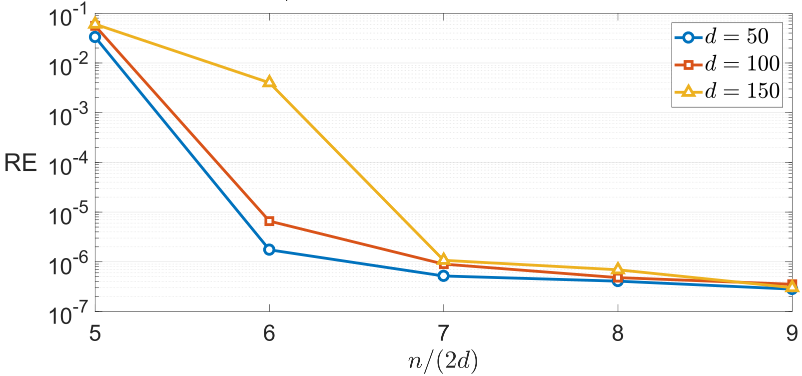
Simulation

Setup: $d_1 = d_2 = d$. Measurement vectors $\boldsymbol{a}_i, \boldsymbol{a}'_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(\boldsymbol{0}, \boldsymbol{I}_{d\times d})$.

Solver: Gurobi for QCLM formulation

Plots show the **median** of the relative error over **100 trials**.

Relative Error: $\sqrt{\dfrac{\left\|\sqrt{\|\widehat{\boldsymbol{x}}'\|_2/\|\widehat{\boldsymbol{x}}\|_2}\,\widehat{\boldsymbol{x}}-\boldsymbol{x}_\star\right\|_2^2+\left\|\sqrt{\|\widehat{\boldsymbol{x}}\|_2/\|\widehat{\boldsymbol{x}}'\|_2}\,\widehat{\boldsymbol{x}}'-\boldsymbol{x}'_\star\right\|_2^2}{\|\boldsymbol{x}_\star\|_2^2+\|\boldsymbol{x}'_\star\|_2^2}}$



Nonlinear parametric regression*

Difference of convex regression:

For a random data point $\boldsymbol{a}$, the observation function is **given** in the DC form as

$$f_{\boldsymbol{a}}(\cdot) = f_{\boldsymbol{a}}^{+}(\cdot) - f_{\boldsymbol{a}}^{-}(\cdot)\,,$$

where the functions $f_{\boldsymbol{a}}^{+}$ and $f_{\boldsymbol{a}}^{-}$ are both convex.

*Bahmani, "Estimation from non-linear observations via convex programming with application to bilinear regression," `arXiv:1806.07307`, 2018.

Nonlinear parametric regression*

Difference of convex regression:

For a random data point $\boldsymbol{a}$, the observation function is **given** in the DC form as

$$f_{\boldsymbol{a}}(\cdot) = f_{\boldsymbol{a}}^{+}(\cdot) - f_{\boldsymbol{a}}^{-}(\cdot)\,,$$

where the functions $f_{\boldsymbol{a}}^{+}$ and $f_{\boldsymbol{a}}^{-}$ are both convex.

Estimate the parameter $\boldsymbol{x}_\star$, from observations at i.i.d. data points $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$, i.e.,

$$y_i = f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star) - f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star) + \xi_i\,, \qquad i = 1, \dots, n\,.$$

*Bahmani, "Estimation from non-linear observations via convex programming with application to bilinear regression," `arXiv:1806.07307`, 2018.

Nonlinear parametric regression*

Difference of convex regression:

For a random data point $\boldsymbol{a}$, the observation function is **given** in the DC form as

$$f_{\boldsymbol{a}}(\cdot) = f_{\boldsymbol{a}}^{+}(\cdot) - f_{\boldsymbol{a}}^{-}(\cdot)\,,$$

where the functions $f_{\boldsymbol{a}}^{+}$ and $f_{\boldsymbol{a}}^{-}$ are both convex.

Estimate the parameter $\boldsymbol{x}_{\star}$, from observations at i.i.d. data points $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$, i.e.,

$$y_i = f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_{\star}) - f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_{\star}) + \xi_i\,, \qquad\qquad i = 1, \ldots, n\,.$$

Estimator

Given $\boldsymbol{x}_0 \approx \frac{1}{2n}\sum_{i=1}^{n} \nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_{\star}) + \nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_{\star})$, we formulate the estimator as

$$\widehat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}}\ \boldsymbol{x}_0^{\top}\boldsymbol{x} - \frac{1}{n}\sum_{i=1}^{n} \max\{f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}) - y_i\,, f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x})\}$$

*Bahmani, "Estimation from non-linear observations via convex programming with application to bilinear regression," `arXiv:1806.07307`, 2018.

Nonlinear parametric regression*

Difference of convex regression:

For a random data point $\boldsymbol{a}$, the observation function is **given** in the DC form as

$$f_{\boldsymbol{a}}(\cdot) = f_{\boldsymbol{a}}^{+}(\cdot) - f_{\boldsymbol{a}}^{-}(\cdot)\,,$$

where the functions $f_{\boldsymbol{a}}^{+}$ and $f_{\boldsymbol{a}}^{-}$ are both convex.

Theorem (simplified)

Let $\Lambda = \sup_{\boldsymbol{h}\in\mathbb{S}^{d-1}} \mathbb{E}\left|\boldsymbol{h}^{\top}\nabla f_{\boldsymbol{a}}(\boldsymbol{x}_{\star})\right|$ *and* $\lambda = \inf_{\boldsymbol{h}\in\mathbb{S}^{d-1}} \mathbb{E}\left|\boldsymbol{h}^{\top}\nabla f_{\boldsymbol{a}}(\boldsymbol{x}_{\star})\right|$. *Then, for a sufficiently accurate* $\boldsymbol{x}_0$, *with probability* $\geq 1-\delta$, *having*

$$n \gtrsim \max\left\{\frac{\Lambda^2}{\lambda^2}\log\left(\frac{2}{\delta}\right), \frac{\Lambda^3}{\lambda^3}d\right\}\,,$$

guarantees

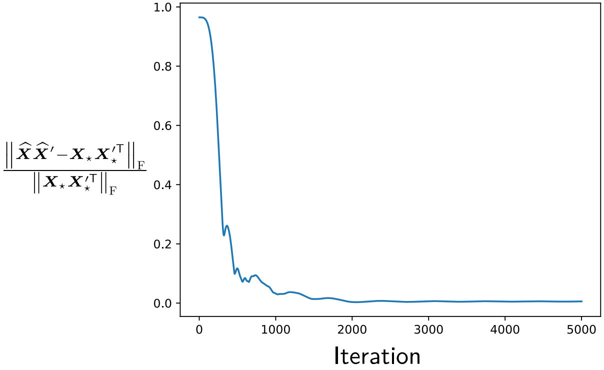
$$\left\|\widehat{\boldsymbol{x}} - \boldsymbol{x}_{\star}\right\|_2 \lesssim \frac{\left\|\boldsymbol{\xi}\right\|_1}{\lambda n}\,.$$

*Bahmani, "Estimation from non-linear observations via convex programming with application to bilinear regression," arXiv:1806.07307, 2018.

Simulation: rank > 1

Setup: For $d = 128$, $r = 3$ the signal is $\boldsymbol{X}_\star \boldsymbol{X}_\star'^{\top}$ with $\boldsymbol{X}_\star, \boldsymbol{X}_\star' \in \mathbb{R}^{d\times r}$ and the measurement functions are $f^{\pm}_{\boldsymbol{a}_i,\boldsymbol{a}_i'}(\boldsymbol{X},\boldsymbol{X}') = \frac{1}{4}\|\boldsymbol{X}^\top \boldsymbol{a}_i \pm \boldsymbol{X}'^{\top}\boldsymbol{a}_i'\|_{\mathrm{F}}^2$ for vectors $\boldsymbol{a}_i$, $\boldsymbol{a}_i' \overset{\text{i.i.d.}}{\sim} \mathrm{Normal}(\boldsymbol{0}, \boldsymbol{I}_{d\times d})$. The observations are then $y_i = \boldsymbol{a}_i^\top \boldsymbol{X}\,\boldsymbol{X}'^{\top}\boldsymbol{a}_i'$

Solver: Nesterov's accelerated gradient method for the smoothed variant



Proof sketch

By convexity of $f_{\boldsymbol{a}_i}^{\pm}$ we have

$$\begin{aligned} &\max\{f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star+\boldsymbol{h})-y_i, f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star+\boldsymbol{h})\} \\ &\geq \max\{\boldsymbol{h}^\top\nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star), \boldsymbol{h}^\top\nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)\}+f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)-(\xi_i)_+ \,. \end{aligned}$$

It suffices to show that

$$\frac{1}{2n}\sum_{i=1}^{n}\left|\left(\nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star)-\nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)\right)^\top\boldsymbol{h}\right| \geq \left\|\boldsymbol{x}_0-\frac{1}{2n}\sum_{i=1}^{n}\nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star)+\nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)\right\|_2\|\boldsymbol{h}\|_2 \,.$$

Using a *PAC-Bayesian* argument (à la Catoni*), we show

$$\frac{1}{2n}\sum_{i=1}^{n}\left|\left(\nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star)-\nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)\right)^\top\boldsymbol{h}\right| \gtrsim \lambda\|\boldsymbol{h}\|_2 \,,$$

with high probability.

*Catoni and Giulini, "Dimension-free PAC-Bayesian bounds for matrices, vectors, and linear least squares regression," arXiv:1712.02747.

Proof sketch

By convexity of $f_{\boldsymbol{a}_i}^{\pm}$ we have

$$\begin{aligned}&\max\{f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star+\boldsymbol{h})-y_i, f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star+\boldsymbol{h})\}\\ &\geq \max\{\boldsymbol{h}^\mathsf{T}\nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star), \boldsymbol{h}^\mathsf{T}\nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)\}+f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)-(\xi_i)_+ .\end{aligned}$$

It suffices to show that

$$\frac{1}{2n}\sum_{i=1}^{n}\left|\left(\nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star)-\nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)\right)^\mathsf{T}\boldsymbol{h}\right| \geq \left\|\boldsymbol{x}_0-\frac{1}{2n}\sum_{i=1}^{n}\nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star)+\nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)\right\|_2\|\boldsymbol{h}\|_2 .$$

Using a *PAC-Bayesian* argument (à la Catoni*), we show

$$\frac{1}{2n}\sum_{i=1}^{n}\left|\left(\nabla f_{\boldsymbol{a}_i}^{+}(\boldsymbol{x}_\star)-\nabla f_{\boldsymbol{a}_i}^{-}(\boldsymbol{x}_\star)\right)^\mathsf{T}\boldsymbol{h}\right| \gtrsim \lambda\|\boldsymbol{h}\|_2 ,$$

with high probability.

*Catoni and Giulini, "Dimension-free PAC-Bayesian bounds for matrices, vectors, and linear least squares regression," arXiv:1712.02747.

Proof sketch

By convexity of $f^{\pm}_{\boldsymbol{a}_i}$ we have

$$\begin{aligned}&\max\{f^{+}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star+\boldsymbol{h})-y_i, f^{-}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star+\boldsymbol{h})\}\\ &\geq \max\{\boldsymbol{h}^\top\nabla f^{+}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star), \boldsymbol{h}^\top\nabla f^{-}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star)\}+f^{-}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star)-(\xi_i)_+ .\end{aligned}$$

It suffices to show that

$$\frac{1}{2n}\sum_{i=1}^{n}\left|\left(\nabla f^{+}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star)-\nabla f^{-}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star)\right)^\top\boldsymbol{h}\right| \geq \left\|\boldsymbol{x}_0-\frac{1}{2n}\sum_{i=1}^{n}\nabla f^{+}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star)+\nabla f^{-}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star)\right\|_2\|\boldsymbol{h}\|_2 .$$

Using a *PAC-Bayesian* argument (à la Catoni*), we show

$$\frac{1}{2n}\sum_{i=1}^{n}\left|\left(\nabla f^{+}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star)-\nabla f^{-}_{\boldsymbol{a}_i}(\boldsymbol{x}_\star)\right)^\top\boldsymbol{h}\right| \gtrsim \lambda\|\boldsymbol{h}\|_2 ,$$

with high probability.

*Catoni and Giulini, "Dimension-free PAC-Bayesian bounds for matrices, vectors, and linear least squares regression," arXiv:1712.02747.

Final remarks

Apply the general result in special cases (e.g., matrix completion, blind deconvolution, ...). Adding regularization could be necessary.

Choice of the *anchor* vector $\boldsymbol{x}_0$ in special problems

Other applications where the approach applies (e.g, machine learning)

Analyzing the *iterated* method: $\cdots \rightarrow \boldsymbol{x}_0^{(t)} \rightarrow \widehat{\boldsymbol{x}}^{(t)} \rightarrow \boldsymbol{x}_0^{(t+1)} \rightarrow \widehat{\boldsymbol{x}}^{(t+1)} \rightarrow \cdots$

Final remarks

Apply the general result in special cases (e.g., matrix completion, blind deconvolution, …). Adding regularization could be necessary.

Choice of the *anchor* vector $\boldsymbol{x}_0$ in special problems

Other applications where the approach applies (e.g, machine learning)

Analyzing the *iterated* method: $\cdots \longrightarrow \boldsymbol{x}_0^{(t)} \longrightarrow \widehat{\boldsymbol{x}}^{(t)} \longrightarrow \boldsymbol{x}_0^{(t+1)} \longrightarrow \widehat{\boldsymbol{x}}^{(t+1)} \longrightarrow \cdots$

Final remarks

- Apply the general result in special cases (e.g., matrix completion, blind deconvolution, …). Adding regularization could be necessary.
- Choice of the *anchor* vector $\boldsymbol{x}_0$ in special problems
- Other applications where the approach applies (e.g, machine learning)
- Analyzing the *iterated* method: $\cdots \to \boldsymbol{x}_0^{(t)} \to \widehat{\boldsymbol{x}}^{(t)} \to \boldsymbol{x}_0^{(t+1)} \to \widehat{\boldsymbol{x}}^{(t+1)} \to \cdots$

Final remarks

- Apply the general result in special cases (e.g., matrix completion, blind deconvolution, ...). Adding regularization could be necessary.
- Choice of the *anchor* vector $\boldsymbol{x}_0$ in special problems
- Other applications where the approach applies (e.g, machine learning)
- Analyzing the *iterated* method: $\cdots \to \boldsymbol{x}_0^{(t)} \to \widehat{\boldsymbol{x}}^{(t)} \to \boldsymbol{x}_0^{(t+1)} \to \widehat{\boldsymbol{x}}^{(t+1)} \to \cdots$

Final remarks

- Apply the general result in special cases (e.g., matrix completion, blind deconvolution, …). Adding regularization could be necessary.
- Choice of the *anchor* vector $\boldsymbol{x}_0$ in special problems
- Other applications where the approach applies (e.g, machine learning)
- Analyzing the *iterated* method: $\cdots \to \boldsymbol{x}_0^{(t)} \to \widehat{\boldsymbol{x}}^{(t)} \to \boldsymbol{x}_0^{(t+1)} \to \widehat{\boldsymbol{x}}^{(t+1)} \to \cdots$

Thank you.

preprint @ `arXiv:1806.07307`