The Projected Power Method: An Efficient Algorithm for Joint Alignment from Pairwise Differences

Yuxin Chen   Emmanuel Candès

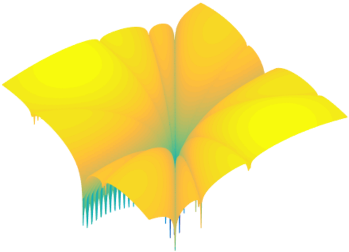
Department of Statistics, Stanford University, *Sep. 2016*

Nonconvex optimization is everywhere

For instance, maximum likelihood estimation is nonconvex in numerous problems

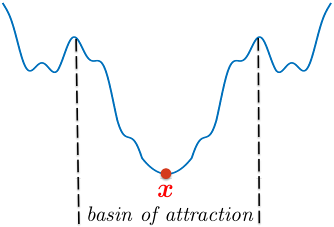
$$\begin{array}{ll} \text{maximize}_{\boldsymbol{x}} & \ell(\boldsymbol{x};\boldsymbol{y}) \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S} \end{array}$$

- matrix completion
- phase retrieval
- dictionary learning
- blind deconvolution
- robust PCA
- ...



Recent flurry of research in nonconvex procedures

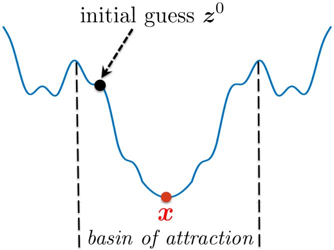
Nice geometry within a neighborhood around x (basin of attraction)



Keshavan et al'08, Netrapalli et al'13, Candès et al'14, Soltanolkotabi'14, Jain et al'14, Sun et al'14, Chen et al'15, Cai et al'15, Tu et al'15, Sun et al'15, White et al'15, Li et al'16, Yi et al'16, Zhang et al'16, Wang et al'16, ...

Recent flurry of research in nonconvex procedures

Nice geometry within a neighborhood around $\boldsymbol{x}$ (basin of attraction)



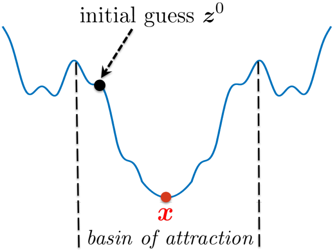
Suggests two-stage paradigms

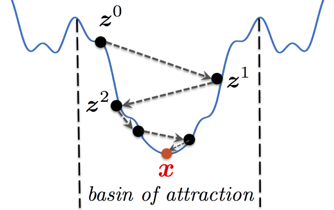
1. Start from an appropriate initial point

Keshavan et al'08, Netrapalli et al'13, Candès et al'14, Soltanolkotabi'14, Jain et al'14, Sun et al'14, Chen et al'15, Cai et al'15, Tu et al'15, Sun et al'15, White et al'15, Li et al'16, Yi et al'16, Zhang et al'16, Wang et al'16, ...

Recent flurry of research in nonconvex procedures

Nice geometry within a neighborhood around $\boldsymbol{x}$ (basin of attraction)





Suggests two-stage paradigms

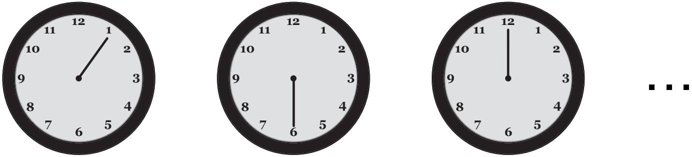
1. Start from an appropriate initial point
2. Proceed via some iterative updates

Keshavan et al'08, Netrapalli et al'13, Candès et al'14, Soltanolkotabi'14, Jain et al'14, Sun et al'14, Chen et al'15, Cai et al'15, Tu et al'15, Sun et al'15, White et al'15, Li et al'16, Yi et al'16, Zhang et al'16, Wang et al'16, ...

This talk: a discrete nonconvex problem

Joint alignment from pairwise differences

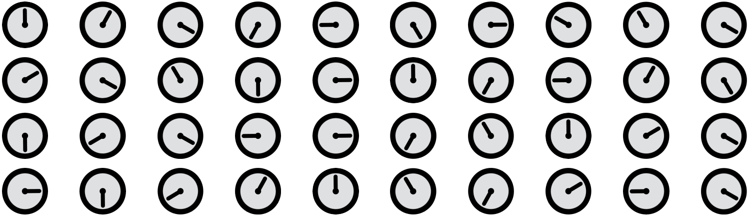
- n unknown variables: $x_1, \cdots, x_n$
- m possible states: $x_i \in \{1, 2, \cdots, m\}$



$x_1 = 1$ $\qquad$ $x_2 = 6$ $\qquad$ $x_3 = 12$

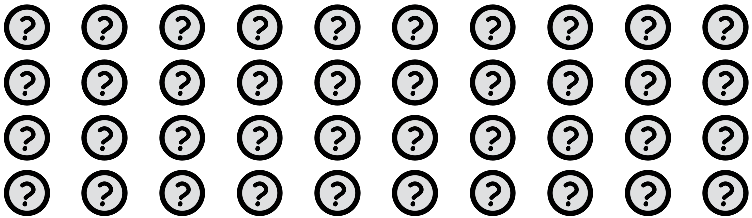
Joint alignment from pairwise differences

- n unknown variables: $x_1, \cdots, x_n$
- m possible states: $x_i \in \{1, 2, \cdots, m\}$



Joint alignment from pairwise differences

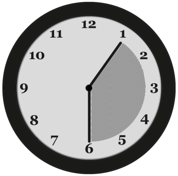
- n unknown variables: $x_1, \cdots, x_n$
- m possible states: $x_i \in \{1, 2, \cdots, m\}$



Joint alignment from pairwise differences

- **Measurements:** pairwise differences

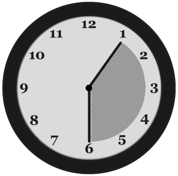
$$y_{i,j} \overset{\text{ind.}}{=} x_i - x_j + \underbrace{\eta_{i,j}}_{\text{noise}} \bmod m, \qquad i \neq j$$



$x_i - x_j \bmod m$

Bandiera, Charikar, Singer, Zhu '13; Chen, Guibas, Huang '14

Joint alignment from pairwise differences

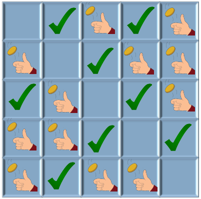


$x_i - x_j \bmod m$

- **Measurements:** pairwise differences

$$y_{i,j} \overset{\text{ind.}}{=} x_i - x_j + \underbrace{\eta_{i,j}}_{\text{noise}} \bmod m, \qquad i \neq j$$

— e.g. random corruption model

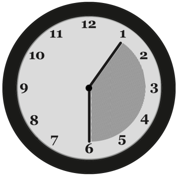


$$y_{i,j} \overset{\text{ind}}{=} \begin{cases} x_i - x_j \text{ mod } m & \text{with prob. } \pi_0 \\ \text{Uniform}\,(m) & \text{else} \end{cases}$$

- π_0: non-corruption rate

Bandiera, Charikar, Singer, Zhu '13; Chen, Guibas, Huang '14

Joint alignment from pairwise differences

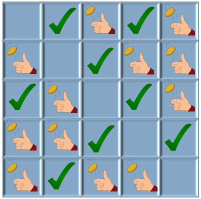


- **Measurements:** pairwise differences

$$y_{i,j} \overset{\text{ind.}}{=} x_i - x_j + \underbrace{\eta_{i,j}}_{\text{noise}} \bmod m, \qquad i \neq j$$

$$x_i - x_j \bmod m$$

— e.g. random corruption model



$$y_{i,j} \overset{\text{ind}}{=} \begin{cases} x_i - x_j \text{ mod } m & \text{with prob. } \pi_0 \\ \text{Uniform}\,(m) & \text{else} \end{cases}$$

- π_0: non-corruption rate

- **Goal:** recover $\{x_i\}$ (up to global offset)

Bandiera, Charikar, Singer, Zhu '13; Chen, Guibas, Huang '14

Motivation: multi-image alignment

Jointly align a collection of images/shapes of the same physical object

Motivation: multi-image alignment

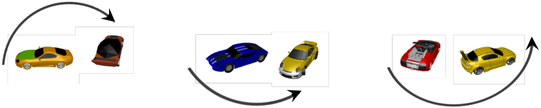
Jointly align a collection of images/shapes of the same physical object

- x_i: angle of rotation associated with each shape

computer vision/graphics

Motivation: multi-image alignment

Step 1: compute pairwise estimates of relative angles of rotations



...

Motivation: multi-image alignment

Step 1: compute pairwise estimates of relative angles of rotations

Step 2: aggregate these pairwise information for joint alignment

Maximum likelihood estimates (MLE)

$$\begin{array}{rl} \text{maximize}_{\{x_i\}} & \displaystyle\sum_{i,j} \ell\left(x_i, x_j; y_{i,j}\right) \\ \text{subj. to} & x_i \in \{1, \cdots, m\}, \quad 1 \le i \le n \end{array}$$

- Log-likelihood function ℓ may be complicated

Maximum likelihood estimates (MLE)

$$\begin{array}{ll} \text{maximize}_{\{x_i\}} & \sum_{i,j} \ell\left(x_i, x_j; y_{i,j}\right) \\ \text{subj. to} & x_i \in \{1, \cdots, m\}, \quad 1 \le i \le n \end{array}$$

- Log-likelihood function ℓ may be complicated
- Discrete input space

Maximum likelihood estimates (MLE)

$$\begin{array}{rl} \text{maximize}_{\{x_i\}} & \sum_{i,j} \ell\left(x_i, x_j; y_{i,j}\right) \\ \text{subj. to} & x_i \in \{1, \cdots, m\}, \quad 1 \le i \le n \end{array}$$

- Log-likelihood function ℓ may be complicated
- Discrete input space
- Looks daunting

Another look in lifted space

Discrete variables $\rightarrow$ orthogonal vectors in higher-dimensional space

$$x_i = 1 \qquad \Longleftrightarrow \qquad \boldsymbol{x}_i = \boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$x_i = 2 \qquad \Longleftrightarrow \qquad \boldsymbol{x}_i = \boldsymbol{e}_1 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$\vdots$

$$x_i = j \qquad \Longleftrightarrow \qquad \boldsymbol{x}_i = \boldsymbol{e}_j$$

Another look in lifted space

Pairwise sample $y_{i,j}$ $\quad\rightarrow\quad$ encode $\ell(x_i, x_j)$ by $\boldsymbol{L}_{i,j} \in \mathbb{R}^{m \times m}$

$$[\boldsymbol{L}_{i,j}]_{\alpha,\beta} = \ell(x_i = \alpha, x_j = \beta)$$

Another look in lifted space

Pairwise sample $y_{i,j} \quad \rightarrow \quad$ encode $\ell(x_i, x_j)$ by $\boldsymbol{L}_{i,j} \in \mathbb{R}^{m \times m}$

$$[\boldsymbol{L}_{i,j}]_{\alpha,\beta} = \ell(x_i = \alpha, x_j = \beta)$$

- e.g. random corruption model

$$y_{i,j} = \begin{cases} x_i - x_j, & \text{w.p. } \pi_0 \\ \mathsf{Unif}(m), & \text{else} \end{cases} \quad \Rightarrow \quad \ell(x_i, x_j) = \begin{cases} \log(\pi_0 + \frac{1-\pi_0}{m}), & \text{if } x_i - x_j = y_{i,j} \\ \log(\frac{1-\pi_0}{m}), & \text{else} \end{cases}$$

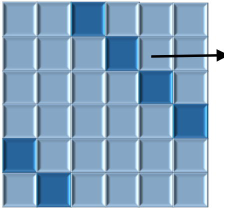
Another look in lifted space

Pairwise sample $y_{i,j} \quad \rightarrow \quad$ encode $\ell(x_i, x_j)$ by $\boldsymbol{L}_{i,j} \in \mathbb{R}^{m \times m}$

$$[\boldsymbol{L}_{i,j}]_{\alpha,\beta} = \ell(x_i = \alpha, x_j = \beta)$$

- e.g. random corruption model

$$y_{i,j} = \begin{cases} x_i - x_j, & \text{w.p. } \pi_0 \\ \text{Unif}(m), & \text{else} \end{cases} \quad \Rightarrow \quad \ell(x_i, x_j) = \begin{cases} \log(\pi_0 + \frac{1-\pi_0}{m}), & \text{if } x_i - x_j = y_{i,j} \\ \log(\frac{1-\pi_0}{m}), & \text{else} \end{cases}$$



$\ell(x_i = 2, x_j = 5; y_{i,j} = 2)$

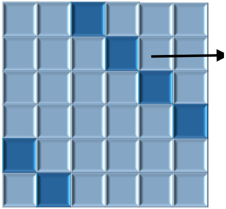
Another look in lifted space

Pairwise sample $y_{i,j}$ $\quad\rightarrow\quad$ encode $\ell(x_i, x_j)$ by $\boldsymbol{L}_{i,j} \in \mathbb{R}^{m\times m}$

$$[\boldsymbol{L}_{i,j}]_{\alpha,\beta} = \ell(x_i = \alpha, x_j = \beta)$$

- e.g. random corruption model

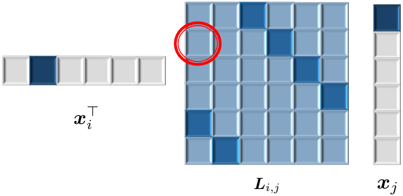
$$y_{i,j} = \begin{cases} x_i - x_j, & \text{w.p. } \pi_0 \\ \text{Unif}(m), & \text{else} \end{cases} \quad \Rightarrow \quad \ell(x_i, x_j) = \begin{cases} \log(\pi_0 + \frac{1-\pi_0}{m}), & \text{if } x_i - x_j = y_{i,j} \\ \log(\frac{1-\pi_0}{m}), & \text{else} \end{cases}$$



$$\ell(x_i = 2, x_j = 5; y_{i,j} = 2)$$

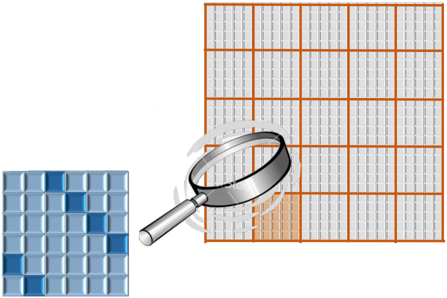
This enables quadratic representation

$$\ell(x_i, x_j) \;=\; \boldsymbol{x}_i^\top \boldsymbol{L}_{i,j} \boldsymbol{x}_j$$



MLE is equivalent to a binary quadratic program

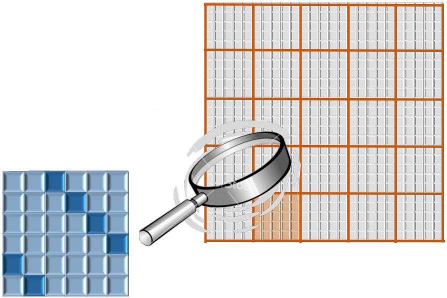
$\boldsymbol{L}_{i,j}$



$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_{1,1} & \cdots & \boldsymbol{L}_{1,n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{L}_{n,1} & \cdots & \boldsymbol{L}_{n,n} \end{bmatrix}$$

MLE is equivalent to a binary quadratic program

$\boldsymbol{L}_{i,j}$

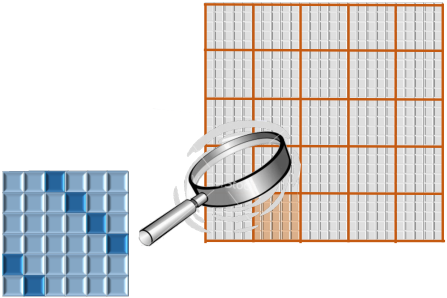


$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_{1,1} & \cdots & \boldsymbol{L}_{1,n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{L}_{n,1} & \cdots & \boldsymbol{L}_{n,n} \end{bmatrix}$$

$$\begin{array}{ll} \text{maximize} & \sum_{i,j} \ell(x_i - x_j; y_{ij}) \\ \text{subj. to} & x_i \in \{1, \cdots, m\} \end{array} \quad \Longleftrightarrow \quad \begin{array}{ll} \text{maximize}_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} \\ \text{subj. to} & \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} \\ & \boldsymbol{x}_i \in \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_m\} \end{array}$$

MLE is equivalent to a binary quadratic program

$\boldsymbol{L}_{i,j}$



$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_{1,1} & \cdots & \boldsymbol{L}_{1,n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{L}_{n,1} & \cdots & \boldsymbol{L}_{n,n} \end{bmatrix}$$

$$\begin{aligned} \text{maximize} \quad & \sum_{i,j} \ell(x_i - x_j; y_{ij}) \\ \text{subj. to} \quad & x_i \in \{1, \cdots, m\} \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \text{maximize}_{\boldsymbol{x}} \quad & \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} \\ \text{subj. to} \quad & \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} \\ & \boldsymbol{x}_i \in \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_m\} \end{aligned}$$

This is essentially nonconvex constrained PCA

How to solve nonconvex constrained PCA?

PCA

$$\begin{array}{ll} \text{maximize}_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} \\ \text{subj. to} & \|\boldsymbol{x}\| = 1 \end{array}$$

Power method:

for $t = 1,\ 2,\ \cdots$

$$\boldsymbol{z}^{(t)} = \boldsymbol{L} \boldsymbol{z}^{(t-1)}$$

$$\boldsymbol{z}^{(t)} \leftarrow \text{normalize}\left(\boldsymbol{z}^{(t)}\right)$$

How to solve nonconvex constrained PCA?

PCA

$$\begin{array}{ll} \text{maximize}_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} \\ \text{subj. to} & \|\boldsymbol{x}\| = 1 \end{array}$$

Constrained PCA

$$\begin{array}{ll} \text{maximize}_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} \\ \text{subj. to} & \boldsymbol{x}_i \in \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_m\} \end{array}$$

Power method:

for $t = 1,\ 2,\ \cdots$

$$\begin{aligned} \boldsymbol{z}^{(t)} &= \boldsymbol{L}\boldsymbol{z}^{(t-1)} \\ \boldsymbol{z}^{(t)} &\leftarrow \text{normalize}\left(\boldsymbol{z}^{(t)}\right) \end{aligned}$$

Projected power method:

for $t = 1,\ 2,\ \cdots$

$$\begin{aligned} \boldsymbol{z}^{(t)} &= \boldsymbol{L}\boldsymbol{z}^{(t-1)} \\ \boldsymbol{z}^{(t)} &\leftarrow \text{Project}_{\Delta^n}\left(\mu\boldsymbol{z}^{(t)}\right) \end{aligned}$$

- μ: scaling factor

Projection onto standard simplex

$$\text{maximize}_{\boldsymbol{x}=\{\boldsymbol{x}_i\}} \quad \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} \qquad \text{s.t. } \boldsymbol{x}_i \in \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_m\}$$

$$\begin{aligned} \boldsymbol{z}^{(t)} &= \boldsymbol{L}\boldsymbol{z}^{(t-1)} \\ \boldsymbol{z}^{(t)} &\leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{z}^{(t)}\right) \end{aligned}$$

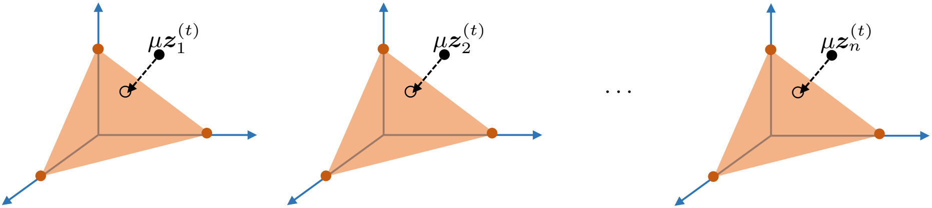
Projection onto standard simplex

$$\text{maximize}_{\boldsymbol{x}=\{\boldsymbol{x}_i\}} \quad \boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} \qquad \text{s.t. } \boldsymbol{x}_i \in \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_m\}$$

$$\boldsymbol{z}^{(t)} = \boldsymbol{L}\boldsymbol{z}^{(t-1)}$$

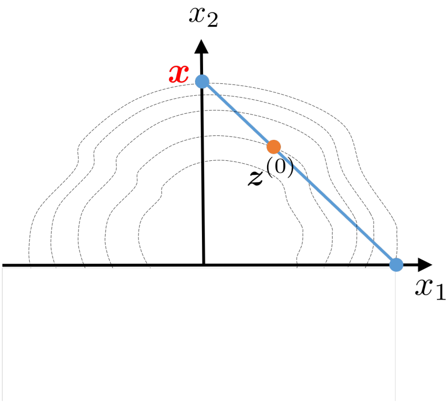
$$\boldsymbol{z}^{(t)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{z}^{(t)}\right)$$

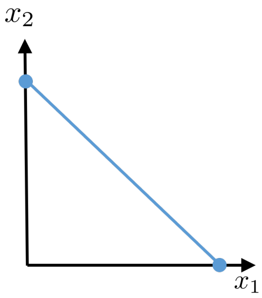
Δ^n is convex hull of feasibility set, i.e. $\left\{\boldsymbol{z} = [\boldsymbol{z}_i]_{1\leq i\leq n} \;\middle|\; \forall i: \; \boldsymbol{1}^\top \boldsymbol{z}_i = 1; \; \boldsymbol{z}_i \geq \boldsymbol{0} \right\}$



Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

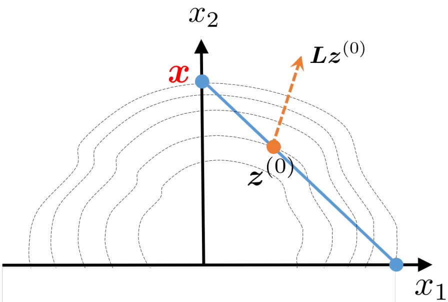




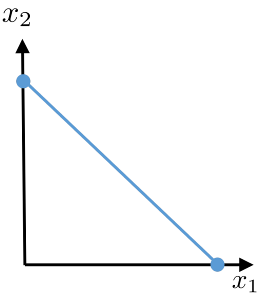
Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

- $\boldsymbol{L}\boldsymbol{z}$ is gradient of $\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}$



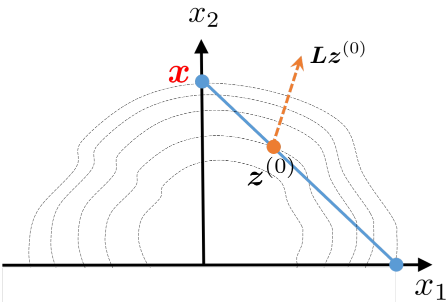
power iteration



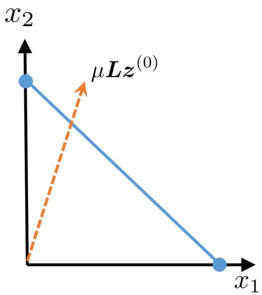
Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

- $\boldsymbol{L}\boldsymbol{z}$ is gradient of $\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}$



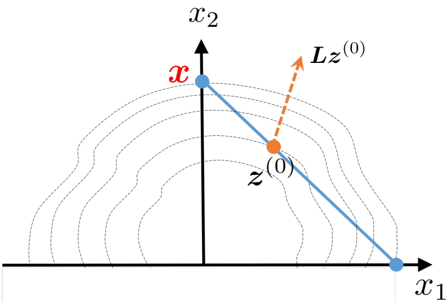
power iteration



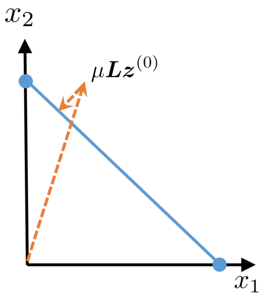
Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

- $\boldsymbol{L}\boldsymbol{z}$ is gradient of $\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}$



power iteration

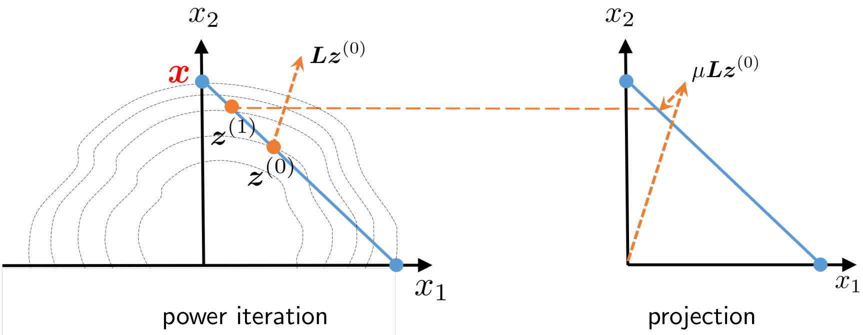


projection

Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

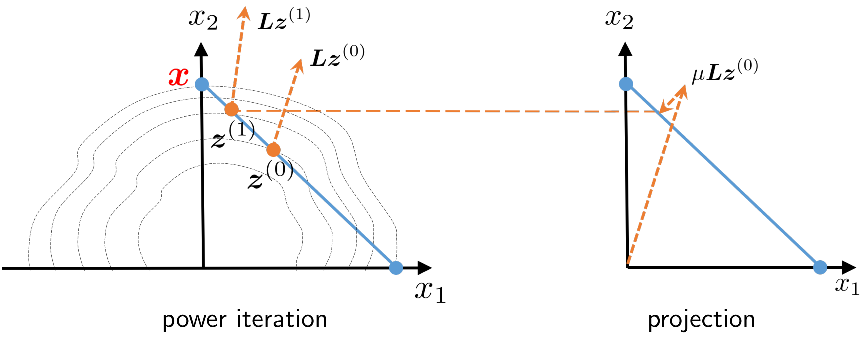
- $\boldsymbol{L}\boldsymbol{z}$ is gradient of $\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}$



Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

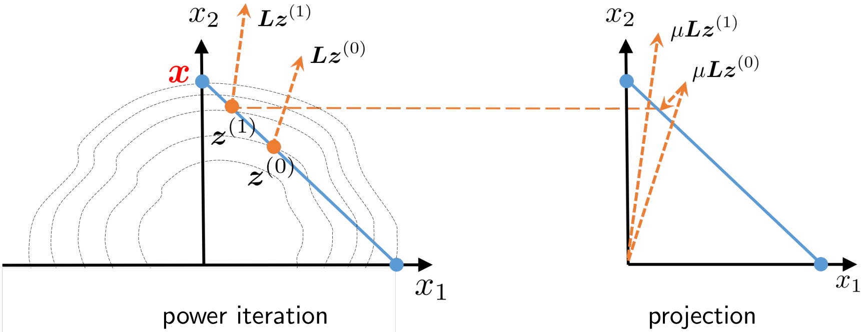
- $\boldsymbol{L}\boldsymbol{z}$ is gradient of $\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}$



Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

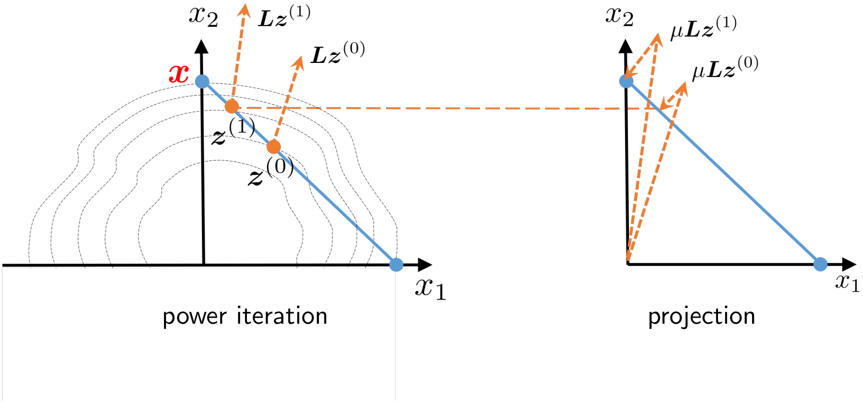
- $\boldsymbol{L}\boldsymbol{z}$ is gradient of $\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}$



Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

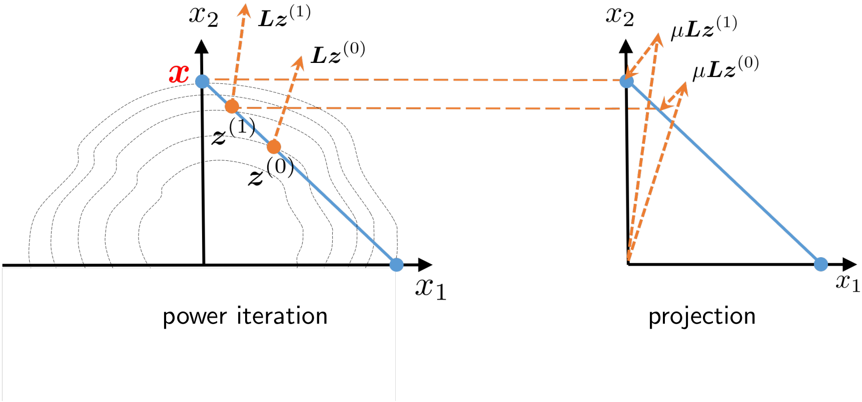
- $\boldsymbol{L}\boldsymbol{z}$ is gradient of $\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}$



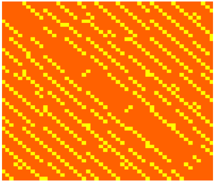
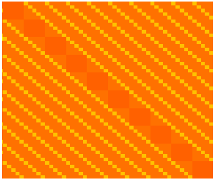
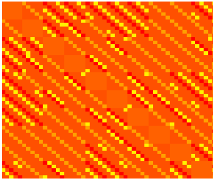
Illustration

Projected power method: $\boldsymbol{z}^{(t+1)} \leftarrow \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t)}\right)$

- $\boldsymbol{L}\boldsymbol{z}$ is gradient of $\frac{1}{2}\boldsymbol{z}^\top \boldsymbol{L}\boldsymbol{z}$

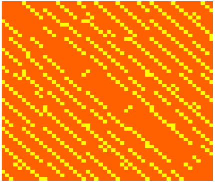


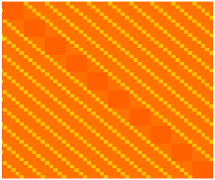
Initialization?

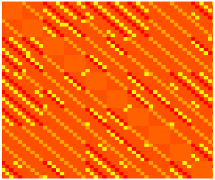
  

$$\boldsymbol{L} = \underbrace{\mathbb{E}[\boldsymbol{L}]}_{\text{approx. low-rank}} + \boldsymbol{L} - \mathbb{E}[\boldsymbol{L}]$$

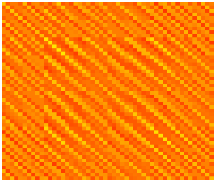
Initialization?







$$\boldsymbol{L} = \underbrace{\mathbb{E}[\boldsymbol{L}]}_{\text{approx. low-rank}} + \boldsymbol{L} - \mathbb{E}\left[\boldsymbol{L}\right]$$

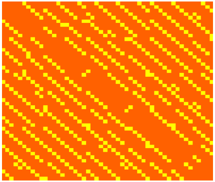


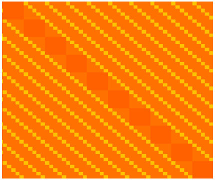
$\hat{\boldsymbol{L}}$

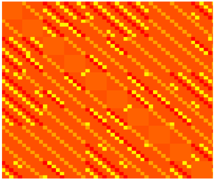
Spectral initialization

1. $\hat{\boldsymbol{L}} \leftarrow$ rank-m approximation of $\boldsymbol{L}$

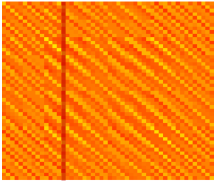
Initialization?







$$\boldsymbol{L} = \underbrace{\mathbb{E}[\boldsymbol{L}]}_{\text{approx. low-rank}} + \boldsymbol{L} - \mathbb{E}[\boldsymbol{L}]$$

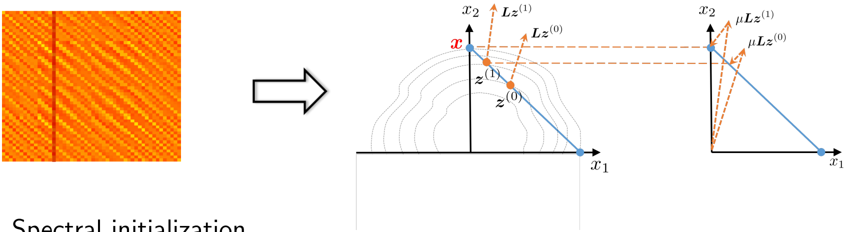


$\hat{\boldsymbol{L}}$

Spectral initialization

1. $\hat{\boldsymbol{L}} \leftarrow$ rank-m approximation of $\boldsymbol{L}$
2. $\boldsymbol{z}^{(0)} \leftarrow$ Project$_{\Delta^n}(\mu\hat{\boldsymbol{z}})$, where $\hat{\boldsymbol{z}}$ is a random column of $\hat{\boldsymbol{L}}$

Summary of projected power method (PPM)



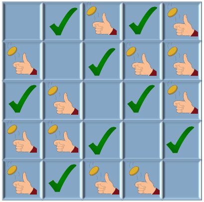
1. Spectral initialization

2. For $t = 1, 2, \cdots$

$$\boldsymbol{z}^{(t)} \quad \leftarrow \quad \text{Project}_{\Delta^n}\left(\mu \boldsymbol{L}\boldsymbol{z}^{(t-1)}\right)$$

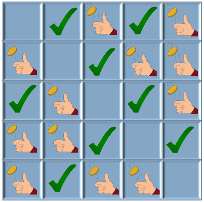
Random corruption model

$$y_{i,j} \overset{\text{ind}}{=} \begin{cases} x_i - x_j \ \mathsf{mod}\ m & \text{with prob. } \pi_0 \\ \mathsf{Uniform}\,(m) & \text{else} \end{cases}$$



Random corruption model

$$y_{i,j} \overset{\text{ind}}{=} \begin{cases} x_i - x_j \ \mathsf{mod}\ m & \text{with prob. } \pi_0 \\ \mathsf{Uniform}\,(m) & \text{else} \end{cases}$$



Theorem (Chen-Candès'16) Fix $m > 0$ and set $\mu \gtrsim 1/\sigma_2(\boldsymbol{L})$. With high prob., PPM recovers the truth exactly within $O(\log n)$ iterations if

- signal-to-noise ratio (SNR) not too small: $\pi_0 > 2\sqrt{\frac{\log n}{mn}}$

Implications

Theorem (Chen-Candès'16) $\cdots$ PPM succeeds within $O(\log n)$ iterations if

non-corruption rate $\pi_0 > 2\sqrt{\frac{\log n}{mn}}$

- PPM succeeds even when most (i.e. $1 - O\big(\sqrt{\frac{\log n}{n}}\big)$) entries are corrupted

Implications

Theorem (Chen-Candès'16) $\cdots$ PPM succeeds within $O(\log n)$ iterations if

$$\text{non-corruption rate} \;\; \pi_0 > 2\sqrt{\frac{\log n}{mn}}$$

- PPM succeeds even when most (i.e. $1 - O\big(\sqrt{\frac{\log n}{n}}\big)$) entries are corrupted
- Nearly linear time algorithm

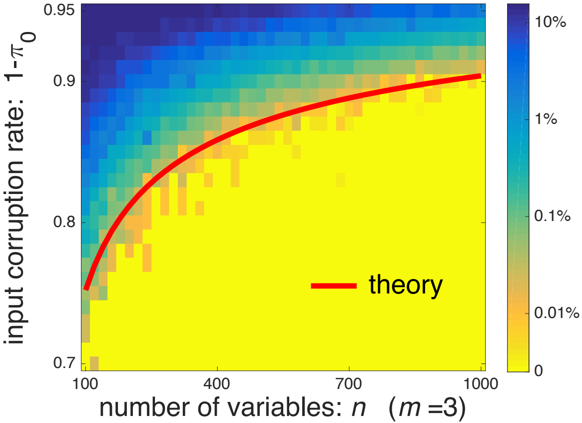
Implications

Theorem (Chen-Candès'16) $\cdots$ PPM succeeds within $O(\log n)$ iterations if

$$\text{non-corruption rate} \quad \pi_0 > 2\sqrt{\frac{\log n}{mn}}$$

- PPM succeeds even when most (i.e. $1 - O\big(\sqrt{\frac{\log n}{n}}\big)$) entries are corrupted
- Nearly linear time algorithm
- Works for any initialization obeying $\|\boldsymbol{z}^{(0)} - \boldsymbol{x}\| < 0.5\|\boldsymbol{x}\|$

Empirical misclassification rate



Misclassification rate when n and π_0 vary $\quad (\mu = 10/\sigma_2(\boldsymbol{L}))$

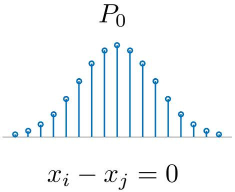
More general noise models

$$y_{i,j} = x_i - x_j + \eta_{i,j} \text{ mod } m, \qquad \text{where } \eta_{i,j} \overset{\text{i.i.d.}}{\sim} P_0$$

More general noise models

$$y_{i,j} = x_i - x_j + \eta_{i,j} \bmod m, \qquad \text{where } \eta_{i,j} \overset{\text{i.i.d.}}{\sim} P_0$$

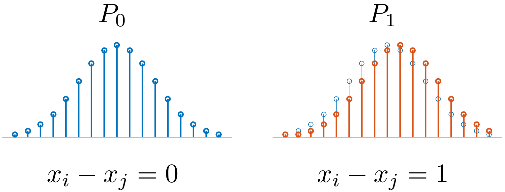
Distributions of $y_{i,j}$ under different hypotheses



More general noise models

$$y_{i,j} = x_i - x_j + \eta_{i,j} \text{ mod } m, \qquad \text{where } \eta_{i,j} \overset{\text{i.i.d.}}{\sim} P_0$$

Distributions of $y_{i,j}$ under different hypotheses

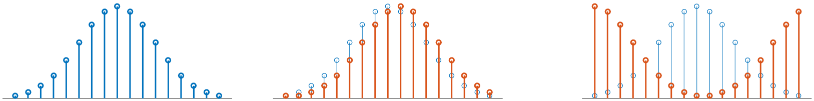


More general noise models

$$y_{i,j} = x_i - x_j + \eta_{i,j} \text{ mod } m, \qquad \text{where } \eta_{i,j} \overset{\text{i.i.d.}}{\sim} P_0$$

Distributions of $y_{i,j}$ under different hypotheses

P_0 $\quad$ P_1 $\quad$ P_9



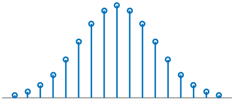
$x_i - x_j = 0$ $\quad$ $x_i - x_j = 1$ $\quad$ $x_i - x_j = 9$

More general noise models

$$y_{i,j} = x_i - x_j + \eta_{i,j} \text{ mod } m, \qquad \text{where } \eta_{i,j} \overset{\text{i.i.d.}}{\sim} P_0$$

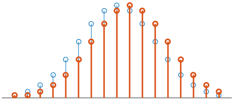
Distributions of $y_{i,j}$ under different hypotheses

P_0



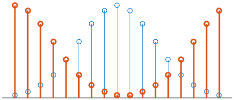
$x_i - x_j = 0$

P_1



$$x_i - x_j = 1$$
$$\Downarrow$$
$$\mathsf{KL}(P_0 \parallel P_1)$$

P_9



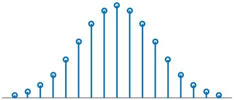
$$x_i - x_j = 9$$
$$\Downarrow$$
$$\mathsf{KL}(P_0 \parallel P_9)$$

More general noise models

$$y_{i,j} = x_i - x_j + \eta_{i,j} \text{ mod } m, \qquad \text{where } \eta_{i,j} \overset{\text{i.i.d.}}{\sim} P_0$$

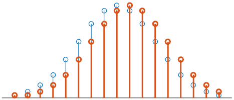
Distributions of $y_{i,j}$ under different hypotheses

P_0



$x_i - x_j = 0$

P_1

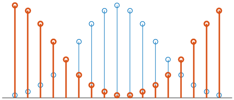


$x_i - x_j = 1$

$\Downarrow$

$\mathsf{KL}(P_0 \parallel P_1)$

P_9



$x_i - x_j = 9$

$\Downarrow$

$\mathsf{KL}(P_0 \parallel P_9)$

Theorem (Chen-Candès'16) Fix $m > 0$ and set $\mu \gtrsim 1/\sigma_2(\boldsymbol{L})$. Under mild conditions, PPM succeeds within $O(\log n)$ iterations with high prob., provided that

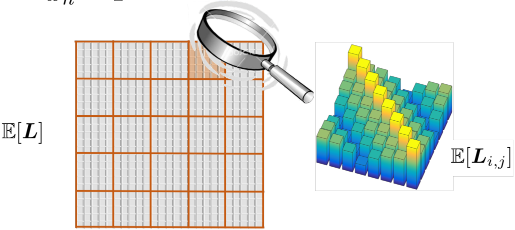
$$\mathsf{KL}_{\min} := \min_{1 \le l < m} \mathsf{KL}(P_0 \parallel P_l) \; > \; \frac{4 \log n}{n}$$

Interpretation: why $\mathsf{KL}_{\min}$ matters

Theorem (Chen-Candès'16) ... PPM succeeds within $O(\log n)$ iterations if

$$\mathsf{KL}_{\min} := \min_{1 \leq l < m} \mathsf{KL}(P_0 \parallel P_l) \; > \; \frac{4 \log n}{n}$$

Suppose $x_1 = \cdots = x_n = 1$

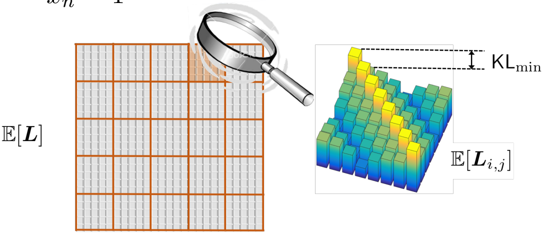


Interpretation: why $\mathsf{KL}_{\min}$ matters

Theorem (Chen-Candès'16) ... PPM succeeds within $O(\log n)$ iterations if

$$\mathsf{KL}_{\min} := \min_{1 \leq l < m} \mathsf{KL}(P_0 \parallel P_l) \;>\; \frac{4 \log n}{n}$$

Suppose $x_1 = \cdots = x_n = 1$

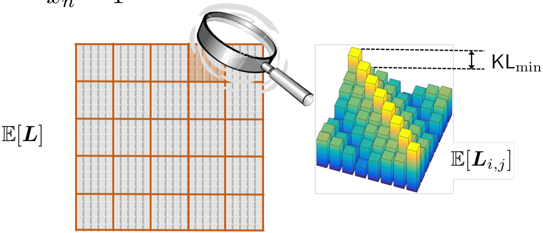


Interpretation: why $\mathsf{KL}_{\min}$ matters

Theorem (Chen-Candès'16) ... PPM succeeds within $O(\log n)$ iterations if

$$\mathsf{KL}_{\min} := \min_{1 \le l < m} \mathsf{KL}(P_0 \parallel P_l) \; > \; \frac{4 \log n}{n}$$

Suppose $x_1 = \cdots = x_n = 1$



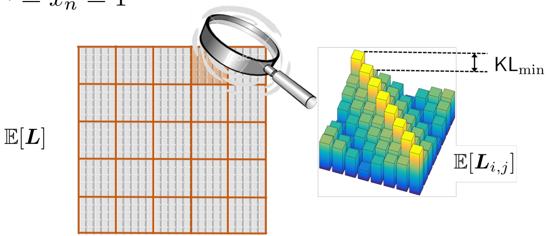
- Peaks of $\mathbb{E}[\boldsymbol{L}]$ reveal ground truth $\mathbb{E}[\boldsymbol{L}_{i,j}]$

Interpretation: why $\mathsf{KL}_{\min}$ matters

Theorem (Chen-Candès'16) ... PPM succeeds within $O(\log n)$ iterations if

$$\mathsf{KL}_{\min} := \min_{1 \leq l < m} \mathsf{KL}(P_0 \parallel P_l) \; > \; \frac{4 \log n}{n}$$

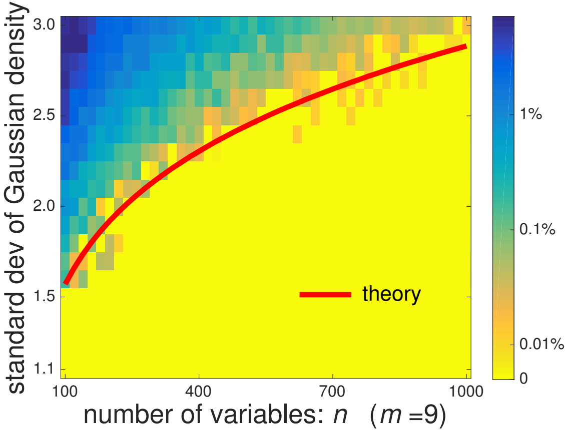
Suppose $x_1 = \cdots = x_n = 1$



- Peaks of $\mathbb{E}[\boldsymbol{L}]$ reveal ground truth $\mathbb{E}[\boldsymbol{L}_{i,j}]$
- $\boldsymbol{L} \; \approx \; \mathbb{E}[\boldsymbol{L}]$ if $\mathsf{KL}_{\min}$ is sufficiently large

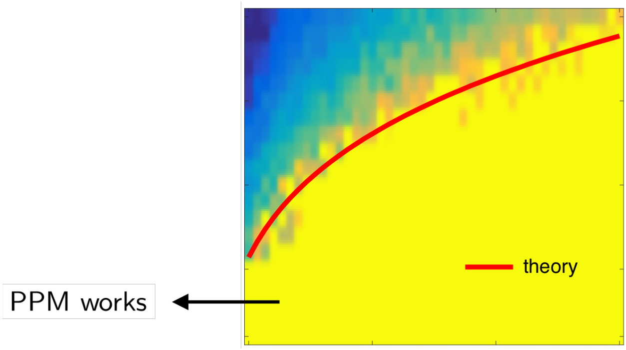
Empirical misclassification rate

Modified Gaussian noise model: $\mathbb{P}\{\eta_{i,j}=z\} \;\propto\; \exp\left(-\frac{z^2}{2\sigma^2}\right), \quad |z| \le \frac{m-1}{2}$



$\mu = 20/\sigma_m(\boldsymbol{L})$

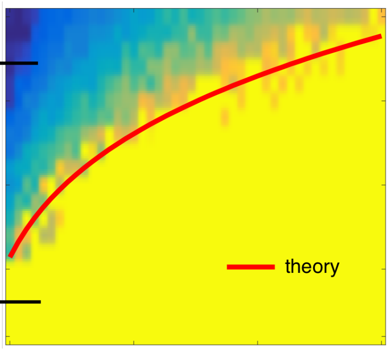
PPM is information-theoretically optimal



PPM is information-theoretically optimal

No algorithm works

PPM works



Theorem (Chen-Candès'16) Fix $m > 0$. No method achieves exact recovery if

$$\mathsf{KL}_{\min} < \frac{4 \log n}{n}$$

Large-m case: random corruption model

$$y_{i,j} = \begin{cases} x_i - x_j, & \text{with prob. } \pi_0 \\ \mathsf{Unif}(m), & \text{else} \end{cases}$$

Theorem (Chen-Candès'16) Suppose $\log n \lesssim m \lesssim \mathrm{poly}(n)$. PPM succeeds if

$$\pi_0 \gtrsim \frac{1}{\sqrt{n}}$$

Singer'09; Wang & Singer'12; Bandeira et al'14; Boumal'16; Liu et al'16, Perry et al'16 ...

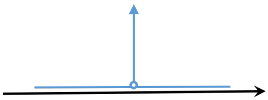
Large-m case: random corruption model

$$y_{i,j} = \begin{cases} x_i - x_j, & \text{with prob. } \pi_0 \\ \text{Unif}(m), & \text{else} \end{cases}$$

Theorem (Chen-Candès'16) Suppose $\log n \lesssim m \lesssim \text{poly}(n)$. PPM succeeds if

$$\pi_0 \gtrsim \frac{1}{\sqrt{n}}$$

- Spiky model: when $m \gg n$, model converges to



$$x_i \in [0,1), \qquad y_{i,j} = \begin{cases} x_i - x_j, & \text{with prob. } \pi_0 \\ \text{Unif}(0,1), & \text{else} \end{cases}$$

Singer'09; Wang & Singer'12; Bandeira et al'14; Boumal'16; Liu et al'16, Perry et al'16 ...

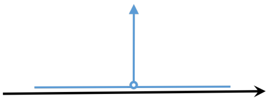
Large-m case: random corruption model

$$y_{i,j} = \begin{cases} x_i - x_j, & \text{with prob. } \pi_0 \\ \text{Unif}(m), & \text{else} \end{cases}$$

Theorem (Chen-Candès'16) Suppose $\log n \lesssim m \lesssim \mathrm{poly}(n)$. PPM succeeds if

$$\pi_0 \gtrsim \frac{1}{\sqrt{n}}$$

- Spiky model: when $m \gg n$, model converges to



$$x_i \in [0,1), \qquad y_{i,j} = \begin{cases} x_i - x_j, & \text{with prob. } \pi_0 \\ \text{Unif}(0,1), & \text{else} \end{cases}$$

  - Succeeds even if a dominant fraction $1 - O(1/\sqrt{n})$ of inputs are corrupted

---

Singer'09; Wang & Singer'12; Bandeira et al'14; Boumal'16; Liu et al'16, Perry et al'16 ...

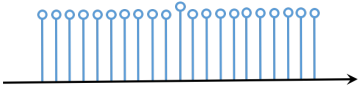
Large-m case: random corruption model

$$y_{i,j} = \begin{cases} x_i - x_j, & \text{with prob. } \pi_0 \\ \mathrm{Unif}(m), & \text{else} \end{cases}$$

Theorem (Chen-Candès'16) Suppose $\log n \lesssim m \lesssim \mathrm{poly}(n)$. PPM succeeds if

$$\pi_0 \gtrsim \frac{1}{\sqrt{n}}$$

- "Smooth" noise model if $m \lesssim \sqrt{n}$



Singer'09; Wang & Singer'12; Bandeira et al'14; Boumal'16; Liu et al'16, Perry et al'16 ...

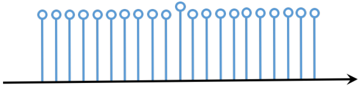
Large-m case: random corruption model

$$y_{i,j} = \begin{cases} x_i - x_j, & \text{with prob. } \pi_0 \\ \mathsf{Unif}(m), & \text{else} \end{cases}$$

Theorem (Chen-Candès'16) Suppose $\log n \lesssim m \lesssim \mathrm{poly}(n)$. PPM succeeds if

$$\pi_0 \gtrsim \frac{1}{\sqrt{n}}$$

- "Smooth" noise model if $m \lesssim \sqrt{n}$



  - Recovers each $x_i \in [0,1)$ up to a resolution of $\frac{1}{m} \asymp \frac{1}{\sqrt{n}}$

Singer'09; Wang & Singer'12; Bandeira et al'14; Boumal'16; Liu et al'16, Perry et al'16 ...

Joint shape alignment: Chair dataset from ShapeNet[1]

20 representative shapes (out of 50)

[1] We add extra noise to each point of the shapes to make it more challenging.

Joint shape alignment: Chair dataset from ShapeNet[1]

20 representative shapes (out of 50)

pairwise cost $-\ell_{i,j}(x_i, x_j)$:
avg nearest-neighbor squared distance

[1]We add extra noise to each point of the shapes to make it more challenging.

Joint shape alignment: Chair dataset from ShapeNet[1]

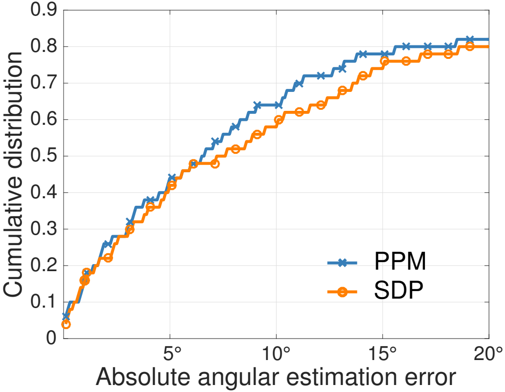
20 representative shapes (out of 50)

pairwise cost $-\ell_{i,j}(x_i, x_j)$:
avg nearest-neighbor squared distance

aligned shapes

[1] We add extra noise to each point of the shapes to make it more challenging.

Joint shape alignment: angular estimation errors[2]



	projected power method	semidefinite relaxation
Runtime	2.4 sec	895.6 sec

[2] We add extra noise to each point of the shapes to make it more challenging.

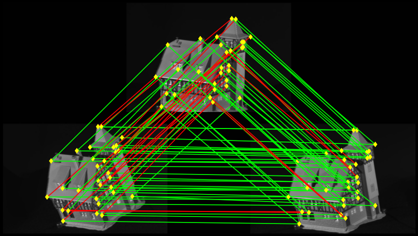
Joint graph matching: CMU House dataset

...

111 images of a toy house

Joint graph matching: CMU House dataset

111 images of a toy house

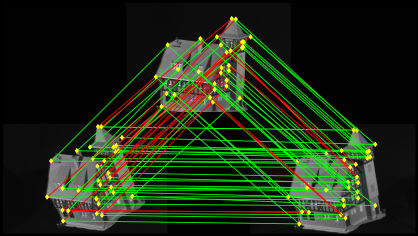


input matches

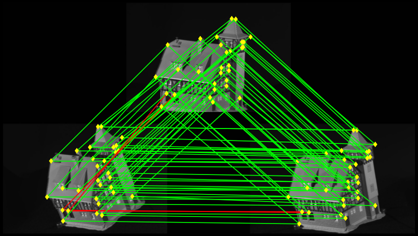
3 representative images

Joint graph matching: CMU House dataset

111 images of a toy house



input matches



optimized matches

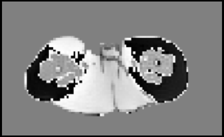
3 representative images

Dixon imaging in body MRI

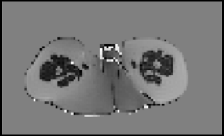
Zhang et al., Magn. Reson. Med., 2016

2 phasor candidates for field inhomogeneity at *each voxel*

candidate 1



candidate 2

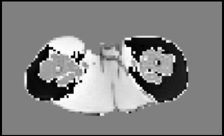


Dixon imaging in body MRI

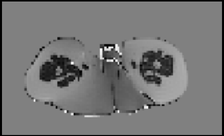
Zhang et al., Magn. Reson. Med., 2016

2 phasor candidates for field inhomogeneity at *each voxel*

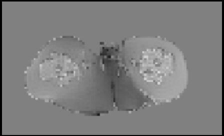
candidate 1



candidate 2



optimize some
pariwise cost function



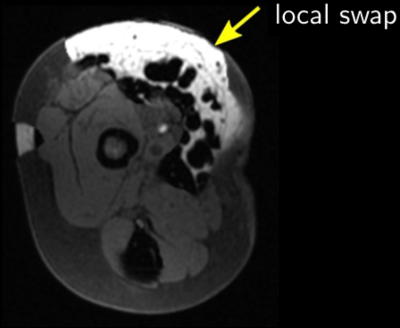
recovery

$$\begin{array}{rl} \text{maximize} & \sum \ell(x_i, x_j) \\ \text{subject to} & x_i \in \{1, 2\} \end{array}$$

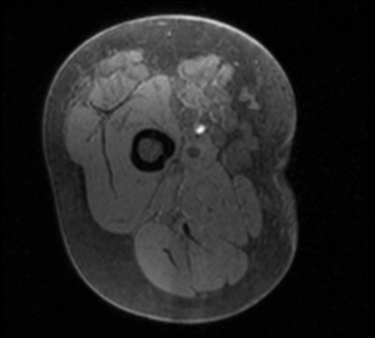
Dixon imaging in body MRI

Zhang et al., Magn. Reson. Med., 2016

Representative cases of water signal recovery



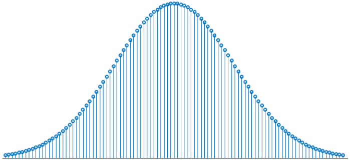
commercial software



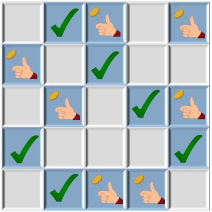
projected power method

Things I have not talked about ...

1. General noise model with large m



2. Incomplete data



Concluding remarks

A new approach to discrete assignment problems

- Finds MLE in suitable regimes
- Computationally efficient

Paper: "The projected power method: an efficient algorithm for joint alignment from pairwise differences", Y. Chen and E. Candès, 2016