Parcimonie en traitement du signal et des images

Partie 4: au delà de la parcimonie

Génie Mathématique - INSA
21/23/28 novembre 2016

N. Bertin, C. Herzet, A. Roumy, *Rémi Gribonval , A. Deleforge
remi.gribonval@inria.fr

Contents

Back to last session: well-posedness

A shorter, easy to memorize formulation of the previous theorem is:

$$\forall \mathbf{X}_0, \mathbf{X}_1 \in \Sigma_r,\ \mathcal{M}(\mathbf{X}_0) = \mathcal{M}(\mathbf{X}_1) \Rightarrow \mathbf{X}_0 = \mathbf{X}_1$$
$$\Leftrightarrow$$
$$\mathrm{Ker}(\mathcal{M}) \cap \Sigma_{2r} = \{\mathbf{0}\}$$

Problem (Homework): consequence for Matrix Completion

- what is the measurement operator $\mathcal{M}$?
- give an upper bound on the rank of an s-sparse matrix $\mathbf{X}$
- for what rank r is the problem well-posed ?

Targeted guarantees

- **Convergence** of algorithms (convergence of cost function and/or of the iterates, to a local or global minimum...),
- **Equivalence** between solutions provided by different algorithms / principles
- **Successful recovery** for a given algorithm, i.e. equivalence with the solution of the ideal low-rank approximation problem

Convergence guarantees

Singular Value Thresholding (SVT) addresses the problem

$$\min_{\mathbf{X}} \tfrac{1}{2}\|\mathcal{M}(\mathbf{X}) - \mathbf{y}\|_2^2 + \lambda\|\mathbf{X}\|_\star$$

with **proximal gradient iterations**, using the proximal operator

$$\mathrm{prox}_{\lambda\|\cdot\|_\star}(\mathbf{Y}) \triangleq \arg\min_{\mathbf{Z}} \tfrac{1}{2}\|\mathbf{Z}-\mathbf{Y}\|_F^2+\lambda\|\mathbf{Z}\|_\star = \mathbf{U}\mathrm{prox}_{\lambda\|\cdot\|_1}(\mathrm{diag}(\mathbf{\Sigma}))\mathbf{V}^T$$

with $\mathbf{Y} = \mathbf{U\Sigma V}^T$ the SVD of $\mathbf{Y}$; $\mathrm{prox}_{\lambda\|\cdot\|_1}(\cdot)$ is soft-thresholding.

Homework 2 (using course on proximal gradient iterations)

Prove the convergence SVT in the sense of *the objective function*

$$\tfrac{1}{2}\|\mathcal{M}(\mathbf{X}_n) - \mathbf{y}\|_2^2 + \lambda\|\mathbf{X}_n\|_\star \leq \frac{C}{n} + \inf_{\mathbf{X}} \tfrac{1}{2}\|\mathcal{M}(\mathbf{X}) - \mathbf{y}\|_2^2 + \lambda\|\mathbf{X}\|_\star.$$

Equivalence guarantees: a Null Space Property

Equivalence (definition): the two problems

$$\min_{\mathbf{X}} \|\mathbf{X}\|_\star \; s.t. \; \mathbf{y} = \mathcal{M}(\mathbf{X}) \quad \textbf{trace} - \textbf{norm minimization}$$

$$\min_{\mathbf{X}} \mathrm{rank}(\mathbf{X}) \; s.t. \; \mathbf{y} = \mathcal{M}(\mathbf{X}) \qquad \textbf{rank} - \textbf{minimization}$$

are equivalent iff: whenever $\mathbf{y} \triangleq \mathcal{M}(\mathbf{X}_0)$ with $\mathrm{rank}(\mathbf{X}_0) \leq r$, their solutions are unique and identical (equal to $\mathbf{X}_0$).
This holds **iff** the following Null Space Property holds:

Null Space Property (NSP) - low rank matrix recovery

For all $p \times q$ matrix $\mathbf{W} \in \mathrm{Ker}(\mathcal{M}) \backslash \{0\}$ with singular values $s_1(\mathbf{W}) \geq \ldots \geq s_n(\mathbf{W}) \geq 0$ where $n \triangleq \min(p, q)$, we have

$$\sum_{j=1}^{r} s_j(\mathbf{V}) < \sum_{j=r+1}^{n} s_j(\mathbf{V}).$$

Demonstration

Step 1 (see course on NSP for sparse recovery)

Prove that equivalence $\Rightarrow$ NSP.

Step 2

Prove that NSP $\Rightarrow$ equivalence.
Hint (see [Foucart & Rauhut 2014, Le A.20]): for any $\mathbf{X}$, $\mathbf{Y}$, ℓ:
$\sum_{j=1}^{\ell} s_j(\mathbf{X}-\mathbf{Y}) \geq \sum_{j=1}^{\ell} |s_j(\mathbf{X}) - s_j(\mathbf{Y})|$

Stability guarantees: the Restricted Isometry Property

Restricted Isometry Property (RIP)- for low rank matrices

$\mathcal{M}$ satisfies the restricted isometry property (RIP) of order s (low-rank version) with constant $\delta \in (0,1)$ **iff**:

$$\forall \mathbf{Z} \in \Sigma_s, \quad (1-\delta)\|\mathbf{Z}\|_F^2 \leq \|\mathcal{M}(\mathbf{Z})\|_2^2 \leq (1+\delta)\|\mathbf{Z}\|_F^2.$$

- The smallest possible δ is denoted $\delta_s(\mathcal{M})$.
- Fully parallel to definition of RIP for s-sparse vectors
- Alternate formulations exist in the litterature: asymetric and/or non-squares. E.g.

$$\alpha \leq \frac{\|\mathcal{M}(\mathbf{Z})\|_2}{\|\mathbf{Z}\|_F} \leq \beta$$

- Informally: preserves the "size" of rank-s matrices

Intuition/link with well-posedness, for $s = 2r$

RIP = preservation of **distance between matrices in** Σ_r

$$\text{consider} \quad \mathbf{Z} = \mathbf{X}_0 - \mathbf{X}_1, \qquad \mathbf{X}_0, \mathbf{X}_1 \in \Sigma_r$$

Stability guarantees: the Restricted Isometry Property

Theorem - Stable Low-rank Recovery assuming the RIP

Assume that $\delta \triangleq \delta_{2r}(\mathcal{M}) < 1/\sqrt{2}$. Then, for any $p \times q$ matrix $\mathbf{X}_0$, considering $\mathbf{y} \triangleq \mathcal{M}(\mathbf{X}_0) + \mathbf{e}$, $\epsilon \geq \|\mathbf{e}\|_2$, and

$$\hat{\mathbf{X}}_\epsilon \triangleq \arg\min_{\mathbf{X}} \|\mathbf{X}\|_\star \; s.t. \; \|\mathbf{y} - \mathcal{M}(\mathbf{X})\|_2 \leq \epsilon$$

we have

$$\|\hat{\mathbf{X}}_\epsilon - \mathbf{X}_0\|_2 \leq C(\delta) \frac{\sigma_r(\mathbf{X}_0)_\star}{\sqrt{r}} + C'(\delta)(\epsilon + \|\mathbf{e}\|_2)$$

- Even when $\mathbf{X}_0$ is *not* low-rank
- Reminder: $\sigma_r(\mathbf{X}_0)_\star \triangleq \inf\{\|\mathbf{X}_0 - \mathbf{Z}\|_\star,\ \mathrm{rank}(\mathbf{Z}) \leq r\}$;
- Explicit expressions for constants $C(\delta)$, $C'(\delta)$
- Fully parallel to result for ℓ_1 and sparse recovery; same dependency of the constants with δ;
- Many successive improvements to bound δ. Bound $\delta < 1/\sqrt{2}$ is "sharp" [Cai and Zhang 2013]

Discussion: when does the RIP hold ?

The RIP implies

$$\|\hat{\mathbf{X}}_\epsilon - \mathbf{X}_0\|_2 \leq C(\delta)\frac{\sigma_r(\mathbf{X}_0)_\star}{\sqrt{r}} + C'(\delta)(\epsilon + \|\mathbf{e}\|_2)$$

- Implies **exact-recovery**: when $\mathbf{X}_0 \in \Sigma_r$ and $\epsilon = \|\mathbf{e}\|_2 = 0$

The Restricted Isometry Property implies the Null Space Property

RIP $\Rightarrow$ NSP $\Leftrightarrow$ exact recovery with trace-norm minimization

$\Rightarrow$ RIP does not hold in the Matrix Completion setting

- **Given** an operator $\mathcal{M}$, its RIP constant is hard to compute

$\Rightarrow$ **design** operator $\mathcal{M}$ with small RIP constant (**& small** m)

Contents

Lessons learned from inverse problems

1. Goal: **given** $\mathcal{M}$, to address inverse problem $\mathbf{y} \approx \mathcal{M}(\mathbf{X})$
2. Model: $\mathbf{X}$ is sparse/low-rank, often reasonable assumption
3. Approach: greedy / iterative / convex algorithms
4. Guarantees: well-posedness, recovery, stability (NSP, RIP)

Necessary dimension

$$m \geq \dim(\Sigma_{2r})$$

Going further: **voluntarily reduce dimension**

1. **Model**: $\mathbf{X}$ is sparse/low-rank, often reasonable assumption

Going further: **voluntarily reduce dimension**

1. Model: $\mathbf{X}$ is sparse/low-rank, often reasonable assumption
2. Goal: to **design** $\mathcal{M} : \mathbf{X} \to \mathbf{y} \triangleq \mathcal{M}(\mathbf{X}) \in \mathbb{R}^m$ so that
   (i) $\mathbf{X}$ **can be stably reconstructed** from $\mathbf{y}$ (with above tools)

Going further: **voluntarily reduce dimension**

1. **Model**: $\mathbf{X}$ is sparse/low-rank, often reasonable assumption
2. **Goal**: to **design** $\mathcal{M} : \mathbf{X} \to \mathbf{y} \triangleq \mathcal{M}(\mathbf{X}) \in \mathbb{R}^m$ so that
   (i) $\mathbf{X}$ **can be stably reconstructed** from $\mathbf{y}$ (with above tools)
   (ii) m is small, to ensure **dimension reduction**

Going further: **voluntarily reduce dimension**

1. **Model**: $\mathbf{X}$ is sparse/low-rank, often reasonable assumption
2. **Goal**: to **design** $\mathcal{M} : \mathbf{X} \to \mathbf{y} \triangleq \mathcal{M}(\mathbf{X}) \in \mathbb{R}^m$ so that
   - (i) $\mathbf{X}$ **can be stably reconstructed** from $\mathbf{y}$ (with above tools)
   - (ii) m is small, to ensure **dimension reduction**
3. **Approach**: design $\mathcal{M}$ that **satisfies the RIP**

Example: Principal Component Analysis (PCA)

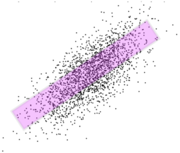
- **Input data**: collection of vectors $\mathbf{z}_i \in \mathbb{R}^d$, $1 \leq i \leq n$

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$$

- **Goal**: find an r-dimensional subspace $V \subset \mathbb{R}^d$ minimizing the average squared approximation error

$$\min_{\dim(V) \leq r} \mathrm{Cost}(V) \triangleq \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{z}_i - P_V \mathbf{z}_i\|_2^2$$

with P_V orthoprojection onto V.



Example: Principal Component Analysis (PCA)

- **Traditional approach**
  1. compute (uncentered) **covariance matrix**

$$\mathbf{X} \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^T = \frac{1}{n}\mathbf{Z}\mathbf{Z}^T$$

  2. compute **eigen value decomp.** of $\mathbf{X}$, (or SVD $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$)

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{U}^T$$

with $\mathbf{D}$ diagonal, decreasing entries

  3. define **matrix of r leading eigenvectors**

$$\mathbf{U}_r \triangleq \mathbf{U}(:, 1:r)$$

and set

$$V \triangleq \mathrm{colspan}(\mathbf{U}_r)$$

Example: Principal Component Analysis (PCA)

- **Traditional approach**

1. compute (uncentered) **covariance matrix** $d \times d$

$$\mathbf{X} \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^T = \frac{1}{n} \mathbf{Z}\mathbf{Z}^T$$

2. compute **eigen value decomp.** of $\mathbf{X}$, (or SVD $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$)

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{U}^T$$

with $\mathbf{D}$ diagonal, decreasing entries

3. define **matrix of r leading eigenvectors** $r \times d \ll d \times d$

$$\mathbf{U}_r \triangleq \mathbf{U}(:, 1:r)$$

and set

$$V \triangleq \mathrm{colspan}(\mathbf{U}_r)$$

Example: Principal Component Analysis (PCA)

- **Traditional approach**
  1. compute (uncentered) **covariance matrix** $d \times d$
     Memory inefficient! Is this really needed ?

$$\mathbf{X} \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^T = \frac{1}{n} \mathbf{Z}\mathbf{Z}^T$$

  2. compute **eigen value decomp.** of $\mathbf{X}$, (or SVD $\mathbf{Z} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$)

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{U}^T$$

with $\mathbf{D}$ diagonal, decreasing entries

  3. define **matrix of r leading eigenvectors** $r \times d \ll d \times d$

$$\mathbf{U}_r \triangleq \mathbf{U}(:, 1:r)$$

and set

$$V \triangleq \mathrm{colspan}(\mathbf{U}_r)$$

Example: Principal Component Analysis (PCA)

- **Traditional approach**

1. compute (uncentered) **covariance matrix** $d \times d$
   Memory inefficient! Is this really needed ?

$$\mathbf{X} \triangleq \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^T = \frac{1}{n} \mathbf{Z}\mathbf{Z}^T$$

2. compute **eigen value decomp.** of $\mathbf{X}$, (or SVD $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$)
   Just need $\mathbf{X}_r = \mathbf{U}_r \mathbf{D}_r \mathbf{U}_r^T$ = *best rank-r approx. to* $\mathbf{X}$

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{U}^T$$

   with $\mathbf{D}$ diagonal, decreasing entries

3. define **matrix of r leading eigenvectors** $r \times d \ll d \times d$

$$\mathbf{U}_r \triangleq \mathbf{U}(:, 1:r)$$

   and set

$$V \triangleq \mathrm{colspan}(\mathbf{U}_r)$$

Example: *Compressive* PCA

- **Main idea**: no need to compute $\mathbf{X} \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i\mathbf{z}_i^T = \frac{1}{n}\mathbf{Z}\mathbf{Z}^T$
  $\Rightarrow$ *just need its best rank-r approximation $\mathbf{X}_r$*
- **Approach**:
  1. **Design** $\mathcal{M}$ satisfying the RIP-δ on low-rank matrices
  2. **Compute** the m entries of $\mathbf{y} = \mathcal{M}(\mathbf{X})$ as

$$y_\ell \triangleq \langle \mathbf{A}_\ell, \mathbf{X} \rangle_F = \frac{1}{n}\sum_{i=1}^{n} \langle \mathbf{A}_\ell, \mathbf{z}_i\mathbf{z}_i^T \rangle_F = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i^T \mathbf{A}_\ell \mathbf{z}_i$$

  3. **Use a low-rank recovery algorithm** to find $\hat{\mathbf{X}} \approx \mathbf{X}_r$.

Example: *Compressive* PCA

- **Main idea**: no need to compute $\mathbf{X} \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i\mathbf{z}_i^T = \frac{1}{n}\mathbf{Z}\mathbf{Z}^T$
  $\Rightarrow$ *just need its best rank-r approximation $\mathbf{X}_r$*
- **Approach**:
  1. **Design** $\mathcal{M}$ satisfying the RIP-δ on low-rank matrices
  2. **Compute** the m entries of $\mathbf{y} = \mathcal{M}(\mathbf{X})$ as

$$y_\ell \triangleq \langle \mathbf{A}_\ell, \mathbf{X}\rangle_F = \frac{1}{n}\sum_{i=1}^{n}\langle \mathbf{A}_\ell, \mathbf{z}_i\mathbf{z}_i^T\rangle_F = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i^T\mathbf{A}_\ell\mathbf{z}_i$$

  3. **Use a low-rank recovery algorithm** to find $\hat{\mathbf{X}} \approx \mathbf{X}_r$.

$$\|\hat{\mathbf{X}} - \mathbf{X}\|_F \quad \leq \quad C(\delta)\frac{\overbrace{\sigma_r(\mathbf{X})_\star}^{\|\mathbf{X}-\mathbf{X}_r\|_\star}}{\sqrt{r}} + \overbrace{C'(\delta)(\epsilon + \|\mathbf{e}\|_2)}^{=0}$$

Example: *Compressive* PCA

- **Main idea**: no need to compute $\mathbf{X} \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i\mathbf{z}_i^T = \frac{1}{n}\mathbf{Z}\mathbf{Z}^T$
  $\Rightarrow$ *just need its best rank-r approximation $\mathbf{X}_r$*
- **Approach**:
  1. **Design** $\mathcal{M}$ satisfying the RIP-δ on low-rank matrices
  2. **Compute** the m entries of $\mathbf{y} = \mathcal{M}(\mathbf{X})$ as

$$y_\ell \triangleq \langle \mathbf{A}_\ell, \mathbf{X}\rangle_F = \frac{1}{n}\sum_{i=1}^{n}\langle \mathbf{A}_\ell, \mathbf{z}_i\mathbf{z}_i^T\rangle_F = \frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i^T\mathbf{A}_\ell\mathbf{z}_i$$

  3. **Use a low-rank recovery algorithm** to find $\hat{\mathbf{X}} \approx \mathbf{X}_r$.

$$\|\hat{\mathbf{X}} - \mathbf{X}\|_F \leq C(\delta)\frac{\overbrace{\sigma_r(\mathbf{X})_\star}^{\|\mathbf{X}-\mathbf{X}_r\|_\star}}{\sqrt{r}}$$

$$\|\hat{\mathbf{X}} - \mathbf{X}_r\|_F \leq \|\hat{\mathbf{X}} - \mathbf{X}\|_F + \|\mathbf{X} - \mathbf{X}_r\|_F \lesssim \|\mathbf{X} - \mathbf{X}_r\|$$

Example: *Compressive* PCA

- **Main idea**: no need to compute $\mathbf{X} \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i\mathbf{z}_i^T = \frac{1}{n}\mathbf{Z}\mathbf{Z}^T$
  $\Rightarrow$ *just need its best rank-r approximation $\mathbf{X}_r$*
- **Approach**:
  1. **Design** $\mathcal{M}$ satisfying the RIP-δ on low-rank matrices
  2. **Compute** the m entries of $\mathbf{y} = \mathcal{M}(\mathbf{X})$ as

$$y_\ell \triangleq \langle \mathbf{A}_\ell, \mathbf{X}\rangle_F = \frac{1}{n}\sum_{i=1}^{n}\langle \mathbf{A}_\ell, \mathbf{z}_i\mathbf{z}_i^T\rangle_F = \frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i^T\mathbf{A}_\ell\mathbf{z}_i$$

  3. **Use a low-rank recovery algorithm** to find $\hat{\mathbf{X}} \approx \mathbf{X}_r$. ✓

$$\|\hat{\mathbf{X}} - \mathbf{X}\|_F \leq C(\delta)\frac{\overbrace{\sigma_r(\mathbf{X})_\star}^{\|\mathbf{X}-\mathbf{X}_r\|_\star}}{\sqrt{r}}$$

$$\|\hat{\mathbf{X}} - \mathbf{X}_r\|_F \leq \|\hat{\mathbf{X}} - \mathbf{X}\|_F + \|\mathbf{X} - \mathbf{X}_r\|_F \lesssim \|\mathbf{X} - \mathbf{X}_r\|$$

Example: *Compressive* PCA

- **Main idea**: no need to compute $\mathbf{X} \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i\mathbf{z}_i^T = \frac{1}{n}\mathbf{Z}\mathbf{Z}^T$

  $\Rightarrow$ *just need its best rank-r approximation* $\mathbf{X}_r$
- **Approach**:
  1. **Design** $\mathcal{M}$ satisfying the RIP-δ on low-rank matrices

     $\Rightarrow$ how to design $\mathcal{M}$ to ensure small δ & small m?
  2. **Compute** the m entries of $\mathbf{y} = \mathcal{M}(\mathbf{X})$ as

     $$y_\ell \triangleq \langle \mathbf{A}_\ell, \mathbf{X}\rangle_F = \frac{1}{n}\sum_{i=1}^{n}\langle \mathbf{A}_\ell, \mathbf{z}_i\mathbf{z}_i^T\rangle_F = \frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i^T\mathbf{A}_\ell\mathbf{z}_i$$
  3. **Use a low-rank recovery algorithm** to find $\hat{\mathbf{X}} \approx \mathbf{X}_r$. ✓

     $$\|\hat{\mathbf{X}} - \mathbf{X}\|_F \leq C(\delta)\frac{\overbrace{\sigma_r(\mathbf{X})_\star}^{\|\mathbf{X}-\mathbf{X}_r\|_\star}}{\sqrt{r}}$$

     $$\|\hat{\mathbf{X}} - \mathbf{X}_r\|_F \leq \|\hat{\mathbf{X}} - \mathbf{X}\|_F + \|\mathbf{X} - \mathbf{X}_r\|_F \lesssim \|\mathbf{X} - \mathbf{X}_r\|$$

Example: *Compressive* PCA

- **Main idea**: no need to compute $\mathbf{X} \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i\mathbf{z}_i^T = \frac{1}{n}\mathbf{Z}\mathbf{Z}^T$
  $\Rightarrow$ *just need its best rank-r approximation $\mathbf{X}_r$*
- **Approach**:
  1. **Design** $\mathcal{M}$ satisfying the RIP-δ on low-rank matrices
     $\Rightarrow$ how to design $\mathcal{M}$ to ensure small δ & small m?
     $\Rightarrow$ how to design $\{\mathbf{A}_\ell\}_{\ell=1}^{m}$?
  2. **Compute** the m entries of $\mathbf{y} = \mathcal{M}(\mathbf{X})$ as

$$y_\ell \triangleq \langle \mathbf{A}_\ell, \mathbf{X} \rangle_F = \frac{1}{n}\sum_{i=1}^{n} \langle \mathbf{A}_\ell, \mathbf{z}_i\mathbf{z}_i^T \rangle_F = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i^T \mathbf{A}_\ell \mathbf{z}_i$$

  3. **Use a low-rank recovery algorithm** to find $\hat{\mathbf{X}} \approx \mathbf{X}_r$. ✓

$$\|\hat{\mathbf{X}} - \mathbf{X}\|_F \leq C(\delta) \frac{\overbrace{\sigma_r(\mathbf{X})_\star}^{\|\mathbf{X}-\mathbf{X}_r\|_\star}}{\sqrt{r}}$$

$$\|\hat{\mathbf{X}} - \mathbf{X}_r\|_F \leq \|\hat{\mathbf{X}} - \mathbf{X}\|_F + \|\mathbf{X} - \mathbf{X}_r\|_F \lesssim \|\mathbf{X} - \mathbf{X}_r\|$$

Randomly measuring matrices

1. **Goal**: to **design** $\mathcal{M} : \mathbf{X} \to \mathbf{y} \triangleq \mathcal{M}(\mathbf{X}) \in \mathbb{R}^m$ so that

   (i) $\mathcal{M}$ satisfies the RIP

   (ii) $m \gtrsim \dim(\Sigma_{2r}) = 2r(p + q - 2r)$

Randomly measuring matrices

1. **Goal**: to **design** $\mathcal{M} : \mathbf{X} \to \mathbf{y} \triangleq \mathcal{M}(\mathbf{X}) \in \mathbb{R}^m$ so that
   (i) $\mathcal{M}$ satisfies the RIP
   (ii) $m \gtrsim \dim(\Sigma_{2r}) = 2r(p + q - 2r)$
2. **Approach**: design $\mathcal{M}$ **at random**

Randomly measuring matrices

1. **Goal**: to **design** $\mathcal{M} : \mathbf{X} \to \mathbf{y} \triangleq \mathcal{M}(\mathbf{X}) \in \mathbb{R}^m$ so that

   (i) $\mathcal{M}$ satisfies the RIP $\Rightarrow$ with high probability

   (ii) $m \gtrsim \dim(\Sigma_{2r}) = 2r(p + q - 2r)$

2. **Approach**: design $\mathcal{M}$ **at random**