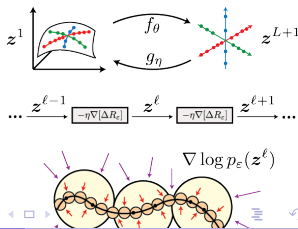


Lecture Two: Learning Low-Dimensional Linear and Independent Structures

Professor Yi Ma

School of Computing and Data Science
The University of Hong Kong

September 21, 2025



① A Low-dim Subspace (PCA)

Singular Value Decomposition

Power Iteration

② Complete Sparsifying Dictionary (DL)

Linear Transform and Dictionary Learning

The MSP Algorithm and Preliminary Experiments [ZYL⁺19]

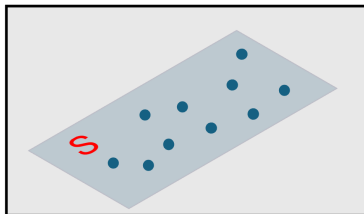
Interpreting ℓ^4 -Maximization and the MSP Algorithm [ZMZM20]

Stability and Robustness of the MSP Algorithm [ZMZM20]

③ Over-Complete Sparsifying Dictionary (ISTA & LISTA)

Learning a Low-dim Subspace

Assumption: the support of the data distribution is on a single low-dim linear subspace \mathcal{S} .



Problem: how to identify base of the subspace from finite noisy samples?

Geometry: Fitting a Linear Model to Data

Find a **“best” hyperplane or subspace** to fit a given set of noisy data [Boscovich 1750, Legendre 1805, Gauss 1809, Wiener 1942, Kalman 1960, Ben Logan 1960, Lasso 1996, Basis Pursuit 1998, Compressive Sensing 2000's]:

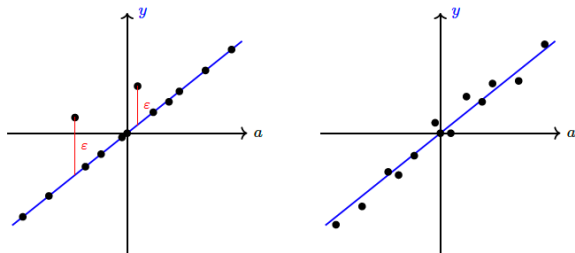


Figure: Left: $\epsilon \sim \frac{1}{2b} \exp\left(-\frac{|\epsilon|}{b}\right)$; Right: $\epsilon \sim \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$

Algebra: Low-Rank Matrix Approximation

Matrix approximation by rank-1 factors [Beltrami and Jordan 1870's]:

$$\mathbf{Y} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^\top + \cdots + \sigma_d \mathbf{u}_d \mathbf{v}_d^\top + \mathbf{E},$$

Low-rank matrix approximation [Eckart and Young 1936]:

Given a matrix of samples: $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$. (1)

find a matrix \mathbf{X}_\star such that

$$\mathbf{X}_\star = \arg \min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{X}\|_2^2 \quad \text{subject to} \quad \text{rank}(\mathbf{X}) \leq d. \quad (2)$$

Statistics: Principal Component Analysis (PCA)

Approximate a high-dim random vector \mathbf{y} by the $d < m$ components as:

$$\mathbf{y} = \mathbf{u}_1 w_1 + \mathbf{u}_2 w_2 + \cdots + \mathbf{u}_d w_d + \boldsymbol{\epsilon} \doteq \mathbf{U} \mathbf{w} + \boldsymbol{\epsilon} \quad \in \mathbb{R}^m, \quad (3)$$

where $\mathbf{U}_d = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d] \in O(m, d)$, $\mathbf{w} = [w_1, w_2, \dots, w_d]^\top \in \mathbb{R}^d$, such that the variance of the residual $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is minimized [Pearson 1901, Hotelling 1933, Jolliffe 1986]:

$$\min_{\mathbf{U}_d, \mathbf{w}} \mathbb{E}[\|\mathbf{y} - \mathbf{U}_d \mathbf{w}\|_2^2]. \quad (4)$$

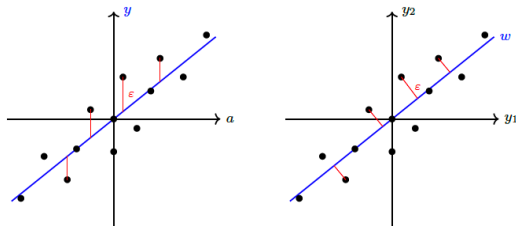


Figure: Left: linear regression; Right: principal component analysis.

Theorem: Singular Value Decomposition (SVD)

Any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with rank $r \leq \min\{m, n\}$ can be decomposed into the following form:

$$\mathbf{A} = \mathbf{U}_r \Sigma_r \mathbf{V}_r^\top = [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_r] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_r \end{bmatrix} \begin{bmatrix} \vec{v}_1^\top \\ \vec{v}_2^\top \\ \vdots \\ \vec{v}_r^\top \end{bmatrix}, \quad (5)$$

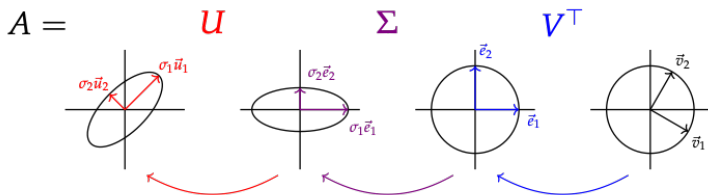
where

$$\begin{aligned} \mathbf{U}_r &\doteq [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_r] \text{ orthogonal,} \\ \mathbf{V}_r &\doteq [\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r] \text{ orthogonal,} \\ \Sigma_r &\doteq \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_r\} > 0 \text{ diagonal.} \end{aligned}$$

Visualization of SVD

Any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with rank $r \leq \min\{m, n\}$ can be decomposed into the following form:

$$\mathbf{A} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top = [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_r] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_r \end{bmatrix} \begin{bmatrix} \vec{v}_1^\top \\ \vec{v}_2^\top \\ \vdots \\ \vec{v}_r^\top \end{bmatrix}. \quad (6)$$



Theorem: How to Compute SVD

Consider the **eigenvalue decomposition** of the real symmetric matrix:

$$\mathbf{A}^\top \mathbf{A} = \sum_{i=1}^r \lambda_i \vec{v}_i \vec{v}_i^\top = \mathbf{V}_r \mathbf{\Lambda}_r \mathbf{V}_r^\top \in \mathbb{R}^{n \times n} \quad (7)$$

with $\lambda_i \geq \lambda_{i+1} \geq 0$ ordered and $\mathbf{V}_r \doteq [\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r] \in \mathbb{R}^{n \times r}$ orthogonal.

Let $\sigma_i = \sqrt{\lambda_i}$ and $\vec{u}_i = \frac{1}{\sigma_i} \mathbf{A} \vec{v}_i \in \mathbb{R}^m$ for $i = 1, \dots, r$. Then we must have $\sigma_i \geq \sigma_{i+1}$ ordered, $\mathbf{U}_r \doteq [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_r] \in \mathbb{R}^{m \times r}$ orthogonal and

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \vec{u}_i \vec{v}_i^\top = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top, \quad \mathbf{\Sigma}_r \doteq \text{diag}\{\sigma_1, \dots, \sigma_r\} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_r \end{bmatrix}.$$

How to Compute SVD: Power Iteration

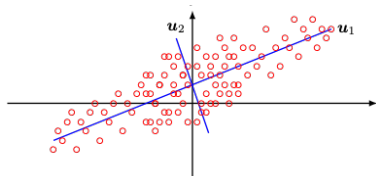
The Power Iteration Algorithm:

- Construct the real symmetric matrix $M \doteq A^\top A \in \mathbb{R}^{n \times n}$.
- Draw a random vector $z_0 \in \mathcal{N}(\mathbf{0}, I_{n \times n}) \in \mathbb{R}^n$.
- For $t = 0, 1, 2, \dots$, **compute iteratively**:

$$z_{t+1} \leftarrow \frac{M z_t}{\|M z_t\|_2} \quad (8)$$

till $\{z_t\}$ converges to the first singular vector u_1 .

This is essentially a “**denoising**” process and converges geometrically fast $O(e^{-t})$.



Theorem: Solution to PCA via SVD

Given a matrix of n samples of \mathbf{y} : $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$, let $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be its Singular Value Decomposition (**SVD**):

$$\mathbf{Y} = \sum_{i=1}^r \sigma_i \vec{u}_i \vec{v}_i^\top = \sum_{i=1}^d \sigma_i \vec{u}_i \vec{v}_i^\top + \sum_{i=d+1}^r \sigma_i \vec{u}_i \vec{v}_i^\top. \quad (9)$$

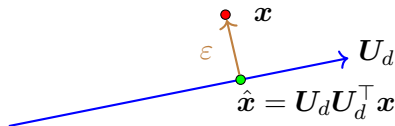
Then the best rank- d approximation to \mathbf{Y} is given by:

$$\mathbf{X}_\star = \sum_{i=1}^d \sigma_i \vec{u}_i \vec{v}_i^\top \doteq \mathbf{U}_d \mathbf{\Sigma}_d \mathbf{V}_d^\top. \quad (10)$$

The optimal estimate for the principal components $[\mathbf{u}_1, \dots, \mathbf{u}_d]$ of \mathbf{y} is given by $[\vec{u}_1, \vec{u}_2, \dots, \vec{u}_d]$.

Denoising Against a Linear Subspace

Figure: Geometry of PCA. A data point x (red) is projected onto the subspace spanned by U_d (blue arrow), as $\hat{x} = U_d U_d^\top x$ (green).



Interpretation as a **two-layer** “deep” network:

$$\text{denoise}(x) = U_\star \circ \underbrace{\text{id} \circ \underbrace{U_\star^\top x}_{\text{first "layer"}}}_{\text{post-activation of first "layer"}}_{\text{output of "NN"}}$$

$$\text{NN}(x) = W_\star \circ \underbrace{\text{ReLU} \circ \underbrace{U_\star^\top x}_{\text{first layer}}}_{\text{post-activation of first layer}}_{\text{output of NN}}$$

Dictionary Learning: General Case

A Fundamental Problems in Data Analysis:

Given an n -dimensional signal: $\mathbf{y} \in \mathbb{R}^n$, find a transformation $\mathcal{T} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ or its “inverse” $\mathbf{D} : \mathbb{R}^m \rightarrow \mathbb{R}^n$, such that

$$\mathbf{x} = \mathcal{T}[\mathbf{y}], \quad \text{or} \quad \mathbf{y} = \mathbf{D}\mathbf{x}$$

where \mathbf{x} highly compressible or the sparsest possible.

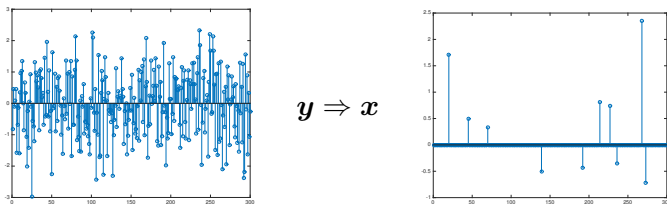


Figure: Sparse Representation Left: a *generic* vector $\mathbf{y} \in \mathbb{R}^n$, Right: a *sparse* representation $\mathbf{x} = \mathcal{T}[\mathbf{y}]$, after a proper transformation \mathcal{T} .

Introduction: History of Finding Good Transform



- **Fourier Transform** $D = F$
- **Wavelet Transform** $D = W$
- **Dictionary Learning**

Figure: Joseph Fourier, 1768 – 1830

Introduction: Fourier Transform

Assumption:

The signal y is **band-limited** and **sparse** in frequency domain: $y_k =$

$$\sum_{l=0}^{n-1} x_l \cdot e^{-\frac{i2\pi}{n}kl} \quad (y = Fx.)$$

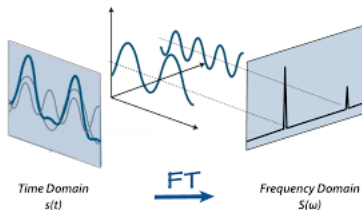


Figure: Fourier Transform

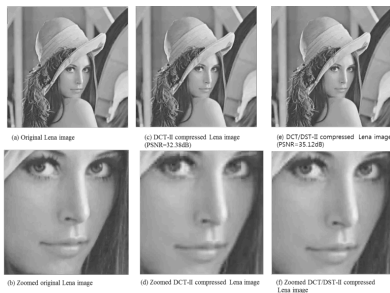


Figure: Lena Compression using Discrete Cosine Transform (JPEG) [pip18]

Introduction: History of Finding Good Transform



Figure: Alfred Haar, 1855 – 1933

- Fourier Transform $D = F$
- **Wavelet Transform** $D = W$
- Dictionary Learning

Introduction: Wavelet Transform

Assumption:

Signal y is piece-wise smooth, scale-invariant, etc: $y = Wx$, $W^*W = I$.

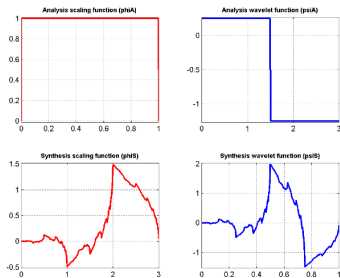


Figure: Haar & Daubechies Wavelets

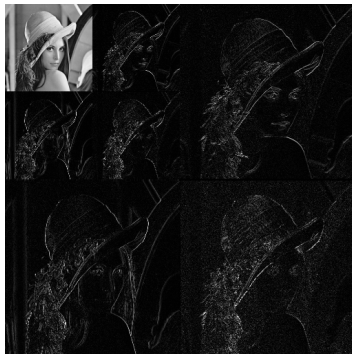


Figure: Lena Compression using Wavelet Transform (JPEG2000) [Jor06]

Why Dictionary Learning?

Limitations of traditional “by-design” methods:

- A transform is not optimal for signals that do not satisfy the conditions under which the transform is designed (e.g. DCT not ideal for images).
- For different classes of signals, we need to design different transforms (e.g. all the x-lets), which may not even be possible if the properties are not clear.

Why Dictionary Learning?

Limitations of traditional “by-design” methods:

- A transform is not optimal for signals that do not satisfy the conditions under which the transform is designed (e.g. DCT not ideal for images).
- For different classes of signals, we need to design different transforms (e.g. all the x-lets), which may not even be possible if the properties are not clear.

For a given class of signals, can we directly “learn” the corresponding optimal transform, from many signal samples?

Dictionary Learning: General Case

Given n -dimensional input data: $\{\mathbf{y}_1, \dots, \mathbf{y}_p\}$, $\forall i \in [p]$, $\mathbf{y}_i \in \mathbb{R}^n$, find a dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$ and its corresponding coefficients $\{\mathbf{x}_1, \dots, \mathbf{x}_p\}$, $\mathbf{x}_i \in \mathbb{R}^m$, such that

$$\mathbf{y}_i = \mathbf{D}\mathbf{x}_i, \quad \forall i \in [p], \quad (11)$$

and \mathbf{x}_i is sufficiently sparse. That is to factor the data matrix \mathbf{Y} into **two structured unknowns**: a matrix \mathbf{D} and a sparse matrix \mathbf{X} :

$$\mathbf{Y} = \underbrace{\begin{pmatrix} | & & | \\ \mathbf{y}_1 & \dots & \mathbf{y}_p \\ | & & | \end{pmatrix}}_{\text{Observations}} = \underbrace{\begin{pmatrix} d_{1,1} & \dots & d_{1,m} \\ \vdots & \ddots & \vdots \\ d_{n,1} & \dots & d_{n,m} \end{pmatrix}}_{\text{Dictionary } \mathbf{D}} \underbrace{\begin{pmatrix} | & & | \\ \mathbf{x}_1 & \dots & \mathbf{x}_p \\ | & & | \end{pmatrix}}_{\mathbf{X} \text{ is sparse, } \|\mathbf{x}_i\|_0 \ll m} = \mathbf{D}\mathbf{X}.$$

Dictionary Learning: General Case

Challenges:

- **Computational complexity**

Optimizing a nonconvex bilinear problem is generally NP-hard.

- **Sample complexity**

Combinatorially many possible patterns of k -sparse x .

- **Signed permutation ambiguities**

$\forall P \in \text{SP}(m)$,¹ $(D_{\star}P, P^*X_{\star})$ and (D_{\star}, X_{\star}) are equally sparse.

¹ $\text{SP}(m)$ denote m dimensional signed permutation group, a group of orthogonal matrices whose entries contain only $0, \pm 1$.

Dictionary Learning: General Case

Challenges:

- **Computational complexity**

Optimizing a nonconvex bilinear problem is generally NP-hard.

- **Sample complexity**

Combinatorially many possible patterns of k -sparse x .

- **Signed permutation ambiguities**

$\forall P \in \text{SP}(m)$,¹ (D_*P, P^*X_*) and (D_*, X_*) are equally sparse.

Some heuristic algorithms:

- K-SVD [AEB⁺06]

- Alternative Direction Methods [SQW17]

¹ $\text{SP}(m)$ denote m dimensional signed permutation group, a group of orthogonal matrices whose entries contain only 0, ± 1 .

Dictionary Learning: General Case

Challenges:

- **Computational complexity**

Optimizing a nonconvex bilinear problem is generally NP-hard.

- **Sample complexity**

Combinatorially many possible patterns of k -sparse x .

- **Signed permutation ambiguities**

$\forall P \in SP(m)$,¹ (D_*P, P^*X_*) and (D_*, X_*) are equally sparse.

Some heuristic algorithms:

- K-SVD [AEB⁺06]

- Alternative Direction Methods [SQW17]

Learn the dictionary with tractable algorithms and sample size?

¹ $SP(m)$ denote m dimensional signed permutation group, a group of orthogonal matrices whose entries contain only 0, ± 1 .

Complete Dictionary Learning

A Random Model:

For complete dictionary learning, [SWW12] assumes data \mathbf{Y} is generated by a **complete**² dictionary $\mathbf{D}_o \in \mathbb{R}^{n \times n}$ and sparse coefficients \mathbf{X}_o :

$$\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o,$$

where \mathbf{X}_o follows a Bernoulli Gaussian model:

$$\mathbf{X}_o = \mathbf{\Omega} \circ \mathbf{G},^3 \quad \Omega_{i,j} \sim_{iid} \text{Ber}(\theta), G_{i,j} \sim_{iid} \mathcal{N}(0, 1).$$

Preconditioning:

[SQW17] shows that learning a complete dictionary is equivalent with learning an **orthogonal** one through preconditioning

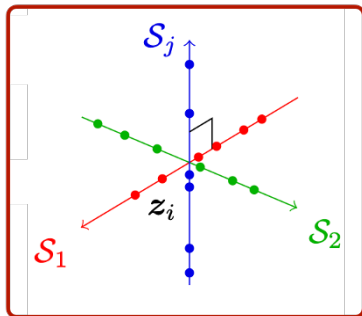
$$\bar{\mathbf{Y}} \leftarrow \left(\frac{1}{p\theta} \mathbf{Y} \mathbf{Y}^* \right)^{-\frac{1}{2}} \mathbf{Y} = \mathbf{D}_o \mathbf{X}_o, \quad \text{with} \quad \mathbf{D}_o \in O(n).$$

²square and invertible

³ \circ denote element-wise product: $\forall \mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}, \{\mathbf{A} \circ \mathbf{B}\}_{i,j} = a_{i,j} b_{i,j}$

Learning a Mixture of Low-dim Subspaces

Assumption: the support of the data distribution is a mixture of low-dimensional **orthogonal** subspaces.



Problem: how to identify bases of the subspaces from finite noisy samples?

Complete Dictionary Learning

Assumes data \mathbf{Y} is generated by a complete **orthogonal dictionary** \mathbf{D}_o and **sparse coefficients** \mathbf{X}_o :

$$\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o,$$

where \mathbf{X}_o follows a Bernoulli Gaussian model:

$$\mathbf{X}_o = \mathbf{\Omega} \circ \mathbf{G}, \quad \Omega_{i,j} \sim_{iid} \text{Ber}(\theta), G_{i,j} \sim_{iid} \mathcal{N}(0, 1).$$

Reduced to find the sparsest direction in a subspace:

- ① \mathbf{D}_o is complete $\implies \boxed{\text{row}(\mathbf{Y}) = \text{row}(\mathbf{X}_o)}$
- ② Rows of \mathbf{X}_o form a *sparse basis* of $\text{row}(\mathbf{Y})$.
- ③ Find \mathbf{x}_1 , the *sparsest vector* in the subspace $\text{row}(\mathbf{Y})$.
- ④ Find \mathbf{x}_i , the *sparsest vector* in $\text{row}(\mathbf{Y}) \setminus \{\mathbf{x}_1, \dots, \mathbf{x}_{i-1}\}$.
- ⑤ Recover \mathbf{D}_o by: $\mathbf{D}_o = \mathbf{Y} \mathbf{X}_o^* (\mathbf{X}_o \mathbf{X}_o^*)^{-1}$.

Complete Dictionary Learning – Prior Arts

Finding the sparsest vector in $\text{row}(\mathbf{Y})$
can be naïvely formulated as:

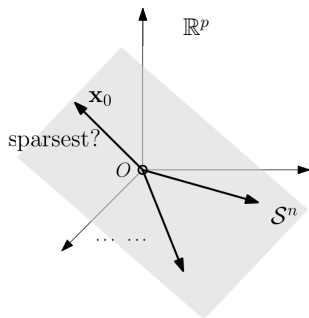
$$\min_{\mathbf{q}} \|\mathbf{q}^* \mathbf{Y}\|_0, \quad \text{s.t.} \quad \mathbf{q} \neq \mathbf{0}.$$

Or minimize the ℓ^1 norm on a sphere
[SQW17, BJS18] (**next lecture**):

$$\min_{\mathbf{q}} \|\mathbf{q}^* \mathbf{Y}\|_1, \quad \text{s.t.} \quad \|\mathbf{q}\|_2 = 1.$$

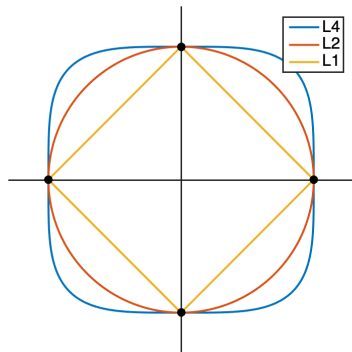
Or maximize the ℓ^4 norm (**this lecture**):

$$\max_{\mathbf{q}} \|\mathbf{q}^* \mathbf{Y}\|_4^4, \quad \text{s.t.} \quad \|\mathbf{q}\|_2 = 1.$$



Solving the same optimization n times (high computation cost)!

Intuition for ℓ^1 and ℓ^4 Norm



Minimizing ℓ^1 norm or maximizing ℓ^4 norm both promote sparsity or spikiness [Wright and Ma, 2022]:

$$\arg \min_{\mathbf{q} \in \mathbb{S}^n} \|\mathbf{q}\|_1 \quad \Leftrightarrow \quad \arg \min_{\mathbf{q} \in \mathbb{S}^n} \|\mathbf{q}\|_0.$$

$$\arg \max_{\mathbf{q} \in \mathbb{S}^n} \|\mathbf{q}\|_4 \quad \Leftrightarrow \quad \arg \min_{\mathbf{q} \in \mathbb{S}^n} \|\mathbf{q}\|_0.$$

Figure: ℓ^1 -, ℓ^2 -, and ℓ^4 -spheres in \mathbb{R}^2

Solving the same optimization n times (high computation cost)!

Intuition for ℓ^4 Norm Maximization [ZYL⁺19]

Consider finding the whole dictionary by the following nonconvex program:

$$\max_{\mathbf{A} \in \mathbf{O}(n; \mathbb{R})} f(\mathbf{A}) = \|\mathbf{A}\mathbf{Y}\|_4^4, \quad (12)$$

which is equivalent to

$$\max_{\mathbf{A} \in \mathbf{O}(n; \mathbb{R})} \|\mathbf{X}\|_4^4, \quad \text{s.t.} \quad \mathbf{Y} = \mathbf{A}^* \mathbf{X}, \quad (13)$$

where maximizing ℓ^4 norm with spherical constraints is promoting “spikiness” [ZKW18].

Related Works of ℓ^4 Norm

- Spherical Harmonic Analysis [SW81, Lu87].
- Independent Component Analysis (ICA) [HO97, HO00]
- Sum of Square (SoS) [BKS15, MSS16, SS17]
- Blind Deconvolution [ZKW18, LB18]

Main Results I

Theorem: Relation to a Deterministic Objective

$\forall \theta \in (0, 1)$, let $\mathbf{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} \text{BG}(\theta)$, $\mathbf{D}_o \in \text{O}(n; \mathbb{R})$ is any orthogonal matrix, and $\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o$. Then $\forall \mathbf{A} \in \text{O}(n; \mathbb{R})$, the expectation of $\|\mathbf{A}\mathbf{Y}\|_4^4$ is determined by function over $\text{O}(n; \mathbb{R})$:

$$\frac{1}{3p\theta} \mathbb{E}_{\mathbf{X}_o} \|\mathbf{A}\mathbf{Y}\|_4^4 = (1 - \theta) \|\mathbf{A}\mathbf{D}_o\|_4^4 + \theta n. \quad (14)$$

Main Results I

Theorem: Relation to a Deterministic Objective

$\forall \theta \in (0, 1)$, let $\mathbf{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} \text{BG}(\theta)$, $\mathbf{D}_o \in \text{O}(n; \mathbb{R})$ is any orthogonal matrix, and $\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o$. Then $\forall \mathbf{A} \in \text{O}(n; \mathbb{R})$, the expectation of $\|\mathbf{A}\mathbf{Y}\|_4^4$ is determined by function over $\text{O}(n; \mathbb{R})$:

$$\frac{1}{3p\theta} \mathbb{E}_{\mathbf{X}_o} \|\mathbf{A}\mathbf{Y}\|_4^4 = (1 - \theta) \|\mathbf{A}\mathbf{D}_o\|_4^4 + \theta n. \quad (14)$$

Global Maxima of the Deterministic Objective

$$\mathbf{W}_\star \in \arg \max_{\mathbf{W} \in \text{O}(n; \mathbb{R})} \|\mathbf{W}\|_4^4 \iff \mathbf{W}_\star \in \text{SP}(n) \quad (15)$$

Main Results I

Theorem: Relation to a Deterministic Objective

$\forall \theta \in (0, 1)$, let $\mathbf{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} \text{BG}(\theta)$, $\mathbf{D}_o \in \text{O}(n; \mathbb{R})$ is any orthogonal matrix, and $\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o$. Then $\forall \mathbf{A} \in \text{O}(n; \mathbb{R})$, the expectation of $\|\mathbf{A}\mathbf{Y}\|_4^4$ is determined by function over $\text{O}(n; \mathbb{R})$:

$$\frac{1}{3p\theta} \mathbb{E}_{\mathbf{X}_o} \|\mathbf{A}\mathbf{Y}\|_4^4 = (1 - \theta) \|\mathbf{A}\mathbf{D}_o\|_4^4 + \theta n. \quad (14)$$

Global Maxima of the Deterministic Objective

$$\mathbf{W}_\star \in \arg \max_{\mathbf{W} \in \text{O}(n; \mathbb{R})} \|\mathbf{W}\|_4^4 \iff \mathbf{W}_\star \in \text{SP}(n) \quad (15)$$

Global maxima of $\|\mathbf{A}\mathbf{D}_o\|_4^4$ are the correct dictionaries (up to signed permutation)!

Main Results II

Theorem: Correctness of Global Optimal

$\forall \theta \in (0, 1)$, let $\mathbf{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} \text{BG}(\theta)$, $\mathbf{D}_o \in \text{O}(n; \mathbb{R})$ is any orthogonal matrix, and $\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o$. Suppose $\hat{\mathbf{A}}_\star$ is a global maximizer of optimization:

$$\max_{\mathbf{A}} \|\mathbf{A}\mathbf{Y}\|_4^4, \quad \text{s.t.} \quad \mathbf{A} \in \text{O}(n; \mathbb{R}), \quad (16)$$

then for any $\varepsilon \in [0, 1]$, there exists a signed permutation matrix

$\mathbf{P} \in \text{SP}(n)$, such that $\frac{1}{n} \left\| \hat{\mathbf{A}}_\star^* - \mathbf{D}_o \mathbf{P} \right\|_F^2 \leq C\varepsilon$, with probability at least $1 - \frac{1}{p}$, when $p = \Omega(\theta n^2 \ln n / \varepsilon^2)$, for a constant $C > \frac{4}{3\theta(1-\theta)}$.

Main Results II

Theorem: Correctness of Global Optimal

$\forall \theta \in (0, 1)$, let $\mathbf{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} \text{BG}(\theta)$, $\mathbf{D}_o \in \text{O}(n; \mathbb{R})$ is any orthogonal matrix, and $\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o$. Suppose $\hat{\mathbf{A}}_\star$ is a global maximizer of optimization:

$$\max_{\mathbf{A}} \|\mathbf{A}\mathbf{Y}\|_4^4, \quad \text{s.t.} \quad \mathbf{A} \in \text{O}(n; \mathbb{R}), \quad (16)$$

then for any $\varepsilon \in [0, 1]$, there exists a signed permutation matrix

$\mathbf{P} \in \text{SP}(n)$, such that $\frac{1}{n} \left\| \hat{\mathbf{A}}_\star^* - \mathbf{D}_o \mathbf{P} \right\|_F^2 \leq C\varepsilon$, with probability at least $1 - \frac{1}{p}$, when $p = \Omega(\theta n^2 \ln n / \varepsilon^2)$, for a constant $C > \frac{4}{3\theta(1-\theta)}$.

With nearly minimal # samples, w.h.p., global maxima of $\|\mathbf{A}\mathbf{Y}\|_4^4$ are arbitrarily close to the correct dictionary!

Optimization Algorithm

The program:

$$\max_{\mathbf{A}} f(\mathbf{A}) \doteq \|\mathbf{A}\mathbf{Y}\|_4^4, \quad \text{s.t.} \quad \mathbf{A} \in \mathcal{O}(n; \mathbb{R})$$

seems to be the worst case for optimization:

- **concave objective;**
- **geometric constraints;**
- **very high dimensional.**

Try projected (Riemannian) gradient descent anyway:

$$\mathbf{A}_{t+1} = \mathcal{P}_{\mathcal{O}(n)}[\mathbf{A}_t + \alpha \nabla f(\mathbf{A}_t)] = \mathcal{P}_{\mathcal{O}(n)}[\mathbf{A}_t + \underbrace{\alpha 4(\mathbf{A}_t \mathbf{Y})^{\circ 3} \mathbf{Y}^*}_{\partial \mathbf{A}_t}].$$

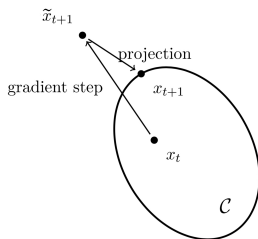
Optimization Algorithm

Solve the program:

$$\max_{\mathbf{A}} f(\mathbf{A}) \doteq \|\mathbf{A}\mathbf{Y}\|_4^4, \quad \text{s.t.} \quad \mathbf{A} \in \mathcal{O}(n; \mathbb{R})$$

with projected (Riemannian) gradient descent:

$$\mathbf{A}_{t+1} = \mathcal{P}_{\mathcal{O}(n)}[\mathbf{A}_t + \alpha \nabla f(\mathbf{A}_t)] = \mathcal{P}_{\mathcal{O}(n)}[\mathbf{A}_t + \underbrace{\alpha 4(\mathbf{A}_t \mathbf{Y})^{\circ 3} \mathbf{Y}^*}_{\partial \mathbf{A}_t}].$$



A happy accident:

observed that this converges faster as $\alpha \rightarrow \infty$!

Why?

something to do with power iteration. (later...)

The MSP Algorithm I

A novel algorithm, with Matching, Stretching (or Sparsifying) and Projection (MSP) to maximize $\|\mathbf{A}\mathbf{Y}\|_4^4$:

Algorithm MSP Algorithm on ℓ^4 Dictionary Learning

- 1: **Initialize** $\mathbf{A}_0 \in \mathcal{O}(n, \mathbb{R})$ ▷ Initialize \mathbf{A}_0 for iteration
 - 2: **for** $t = 0, 1, \dots$
 - 3: $\partial \mathbf{A}_t = 4(\mathbf{A}_t \mathbf{Y})^{\circ 3} \mathbf{Y}^*$ ▷ Matching and Stretching⁴
 - 4: $\mathbf{U} \Sigma \mathbf{V}^* = \text{svd}(\partial \mathbf{A}_t)$
 - 5: $\mathbf{A}_{t+1} = \mathbf{U} \mathbf{V}^*$ ▷ Project \mathbf{A} onto orthogonal group
 - 6: **end for**
 - 7: **Output** $\mathbf{A}_{t+1}, \|\mathbf{A}_{t+1} \mathbf{Y}\|_4^4 / 3np\theta, \|\mathbf{A}_{t+1} \mathbf{D}_o\|_4^4 / n$
-

⁴ $\nabla_{\mathbf{A}} \|\mathbf{A}\mathbf{Y}\|_4^4 = 4(\mathbf{A}\mathbf{Y})^{\circ 3} \mathbf{Y}^*$

A Few Interpretations

NOT Gradient Descent!

“Fixed point” interpretation:

$$\mathbf{A}_{t+1} = \mathcal{P}_{O(n)}[\partial \mathbf{A}_t] = \mathcal{P}_{O(n)}[(\mathbf{A}_t \mathbf{Y})^{\circ 3} \mathbf{Y}^*].$$

“Deep learning” interpretation: $\delta \mathbf{A}_{t+1} = \mathbf{A}_{t+1} \mathbf{A}_t^*$ and $\mathbf{Z}_t = \mathbf{A}_t \mathbf{Y}$,

$$\delta \mathbf{A}_{t+1} = \mathcal{P}_{O(n)}[(\mathbf{Z}_t)^{\circ 3} \mathbf{Z}_t^*], \quad \mathbf{X} \leftarrow \underbrace{\delta \mathbf{A}_{t+1} \delta \mathbf{A}_t \dots \delta \mathbf{A}_1}_{\text{forward constructed layers!}} \mathbf{Y}.$$

“Stochastic batch” variation:

$$\delta \mathbf{A}_{t+1} = \mathcal{P}_{O(n)}[(\tilde{\mathbf{Z}}_t)^{\circ 3} \tilde{\mathbf{Z}}_t^*], \quad \tilde{\mathbf{Z}}_t \subseteq \mathbf{Z}_t.$$

The MSP Algorithm II

Since $\|\mathbf{A}\mathbf{D}_o\|_4^4$ has a linear relation with $\frac{1}{np}\mathbb{E}_{\mathbf{X}_o}\|\mathbf{A}\mathbf{Y}\|_4^4$, a similar algorithm also can be applied to maximize $\|\mathbf{A}\mathbf{D}_o\|_4^4$:

Algorithm MSP Algorithm on ℓ^4 over Orthogonal Group

- | | |
|---|--|
| 1: Initialize $\mathbf{A}_0 \in O(n, \mathbb{R})$ | ▷ Initialize \mathbf{A}_0 for iteration |
| 2: for $t = 0, 1, \dots$ | |
| 3: $\partial\mathbf{A}_t = 4(\mathbf{A}_t\mathbf{D}_o)^{\circ 3}\mathbf{D}_o^*$ | ▷ Matching and Stretching |
| 4: $\mathbf{U}\Sigma\mathbf{V}^* = \text{svd}(\partial\mathbf{A}_t)$ | |
| 5: $\mathbf{A}_{t+1} = \mathbf{U}\mathbf{V}^*$ | ▷ Project \mathbf{A} onto orthogonal group |
| 6: end for | |
| 7: Output $\mathbf{A}_{t+1}, \ \mathbf{A}_{t+1}\mathbf{D}_o\ _4^4/n$ | |
-

One Run of the MSP Algorithm

$$\begin{array}{ll}
 \mathbf{A}_0 = \begin{pmatrix} -0.8249 & 0.3820 & -0.4168 \\ -0.5240 & -0.2398 & 0.8173 \\ -0.2122 & -0.8925 & -0.3979 \end{pmatrix} & \xrightarrow{\text{stretching}} \mathbf{A}_0^{\circ 3} = \begin{pmatrix} -0.5613 & 0.0557 & -0.0724 \\ -0.1439 & -0.0138 & 0.5459 \\ -0.0096 & -0.7109 & -0.0630 \end{pmatrix} \\
 \xrightarrow{\text{projection}} \mathbf{A}_1 = \begin{pmatrix} -0.9795 & 0.0621 & -0.1917 \\ -0.1953 & -0.0594 & 0.9789 \\ -0.0494 & -0.9963 & -0.0703 \end{pmatrix} & \xrightarrow{\text{stretching}} \mathbf{A}_1^{\circ 3} = \begin{pmatrix} -0.9397 & 0.0002 & -0.0070 \\ -0.0075 & -0.0002 & 0.9381 \\ -0.0001 & -0.9889 & -0.0003 \end{pmatrix} \\
 \xrightarrow{\text{projection}} \mathbf{A}_2 = \begin{pmatrix} -1.0000 & 0.0002 & -0.0077 \\ -0.0077 & -0.0003 & 1.000 \\ -0.0002 & -1.0000 & -0.0003 \end{pmatrix} & \xrightarrow{\text{stretching}} \mathbf{A}_2^{\circ 3} = \begin{pmatrix} -0.9999 & 0.0000 & -0.0000 \\ -0.0000 & -0.0000 & 0.9999 \\ -0.0000 & -1.0000 & -0.0000 \end{pmatrix} \\
 \xrightarrow{\text{projection}} \mathbf{A}_3 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} & \xrightarrow{\text{output}} \mathbf{A}_3^{\circ 3} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}.
 \end{array}$$

Figure: One run of the MSP algorithm for maximizing $\|\mathbf{A}\mathbf{D}_o\|_4^4$ over orthogonal group $O(3)$ with $\mathbf{D}_o = \mathbf{I}$.

Convergence Guarantee of the MSP Algorithm

Theorem (Local Convergence of the MSP Algorithm)

Given an orthogonal matrix $\mathbf{A} \in O(n; \mathbb{R})$, let \mathbf{A}' denote the output of the MSP Algorithm 2 after one iteration: $\mathbf{A}' = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \text{SVD}(\mathbf{A}^{\circ 3})$. If $\|\mathbf{A} - \mathbf{I}\|_F^2 = \varepsilon$, for $\varepsilon < 0.579$, then we have $\|\mathbf{A}' - \mathbf{I}\|_F^2 < \|\mathbf{A} - \mathbf{I}\|_F^2$ and $\|\mathbf{A}' - \mathbf{I}\|_F^2 < O(\varepsilon^3)$.

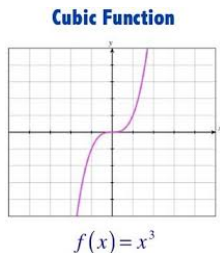


Figure: Cubic Function from ℓ^4 .

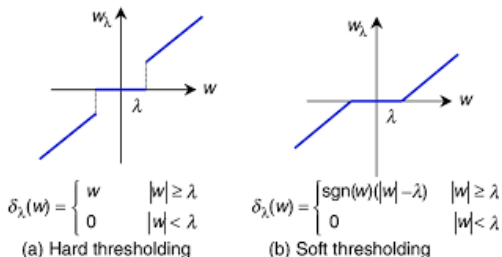


Figure: Thresholding from ℓ^1 .

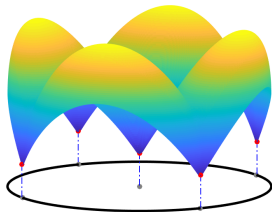
Convergence Guarantee of the MSP Algorithm

Theorem (Local Convergence of the MSP Algorithm)

Given an orthogonal matrix $\mathbf{A} \in \mathcal{O}(n; \mathbb{R})$, let \mathbf{A}' denote the output of the MSP Algorithm 2 after one iteration: $\mathbf{A}' = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^$, where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \text{SVD}(\mathbf{A}^{\circ 3})$. If $\|\mathbf{A} - \mathbf{I}\|_F^2 = \varepsilon$, for $\varepsilon < 0.579$, then we have $\|\mathbf{A}' - \mathbf{I}\|_F^2 < \|\mathbf{A} - \mathbf{I}\|_F^2$ and $\|\mathbf{A}' - \mathbf{I}\|_F^2 < O(\varepsilon^3)$.*

Generalization to all Signed Permutation Matrices

The Identity can be generalized to any signed permutation matrix!



MSP algorithm in Maximizing $\|\mathbf{AY}\|_4^4$

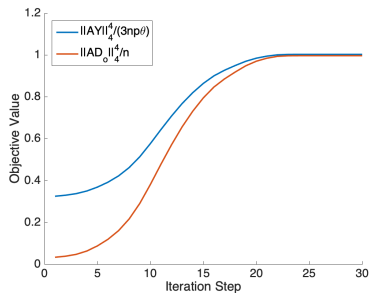
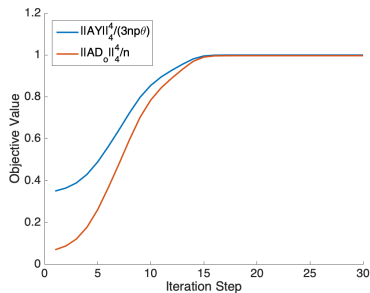


Figure: The value of $\frac{1}{3np^\theta} \|\mathbf{AY}\|_4^4$ and $\frac{1}{n} \|\mathbf{AD}_o\|_4^4$ in two experiments with different settings: left: $n = 50, p = 20000, \theta = 0.3$, right: $n = 100, p = 40000, \theta = 0.3$. **The MSP algorithm converges quickly and smoothly with dozens of iterations.**

MSP algorithm in Maximizing $\|\mathbf{A}\mathbf{Y}\|_4^4$

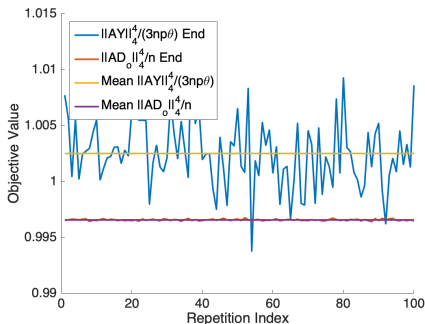
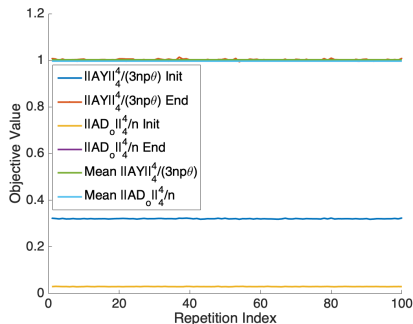


Figure: Initial value and final value of $\frac{1}{3np^\theta} \|\mathbf{A}\mathbf{Y}\|_4^4$ and $\frac{1}{n} \|\mathbf{A}\mathbf{D}_o\|_4^4$ for dictionary learning, with $n = 100, p = 40000, \theta = 0.3$, left: with initial values; right: without initial values. **All 100 trials converge to the global optima within statistical errors.**

Phase Transition of the MSP algorithm

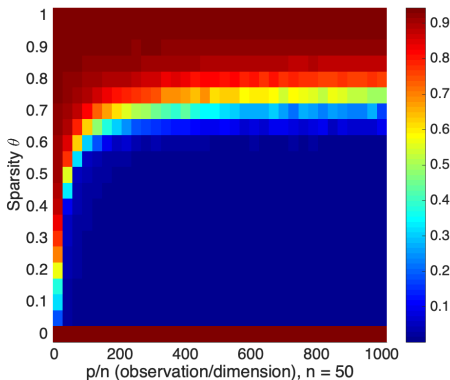


Figure: Phase transition plot of average normalized error $\left|1 - \|\mathbf{AD}_o\|_4^4/n\right|$ for 10 trials of MSP algorithm 1 with $n = 50$. Red area indicates large error and blue area small error. Plot shows results for varying p versus θ . **The algorithm succeeds even when θ is up to 0.6!**

Phase Transition of the MSP algorithm

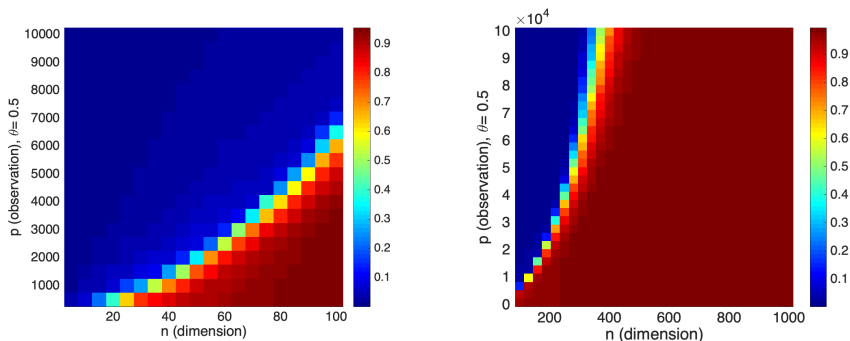


Figure: Phase transition plot of average normalized error $\left|1 - \|\mathbf{AD}_o\|_4^4/n\right|$ for 10 trials of MSP algorithm 1 with $\theta = 0.5$. Red area indicates large error and blue area small error, left: n from 10 to 100 and p from 10^3 to 10^4 , right: changing n from 100 to 10^3 and p from 10^4 to 10^5 . **The number of samples p needed is quadratic in n .**

Optimal Choice of ℓ^{2k} Norm

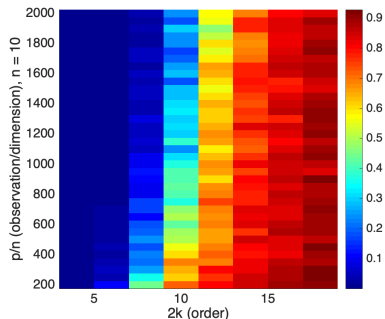
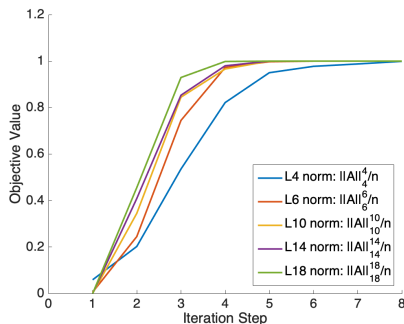


Figure: Experiments with different ℓ^{2k} norm. Left: Maximizing $\|A\|_{2k}^{2k}$ for different order k . Right: Average normalized error of $\left|1 - \|AD_o\|_{2k}^{2k}/n\right|$ for maximizing $\|AY\|_{2k}^{2k}$ for 20 trials, with $n = 10$, varying k and p . **ℓ^4 strikes a good balance between convergence and concentration.**

Comparison with the State of the Art

	KSVD		Subgradient		MSP (Ours)	
Trials	Error	Time	Error	Time	Error	Time
(a)	12.35%	51.2s	0.27%	35.6s	0.34%	0.4s
(b)	8.63%	244.4s	0.28%	354.9s	0.34%	1.5s
(c)	6.15%	684.9s	1.28%	6924.6s	0.35%	7.6s
(d)	8.61%	1042.3s	N/A	> 12h	0.35%	48.0s
(e)	13.07%	5401.9s	N/A	> 12h	0.35%	374.2s

Table: Comparison experiments with KSVD [AEB⁺06] and Subgradient method [BJS18] in different trials of dictionary learning: (a) $n = 25, p = 1 \times 10^4, \theta = 0.3$; (b) $n = 50, p = 2 \times 10^4, \theta = 0.3$; (c) $n = 100, p = 4 \times 10^4, \theta = 0.3$; (d) $n = 200, p = 4 \times 10^4, \theta = 0.3$; (e) $n = 400, p = 16 \times 10^4, \theta = 0.3$. Recovery error is measured as $|1 - \|\mathbf{A}\mathbf{D}_o\|_4^4/n|$. All experiments are conducted on a 2.7 GHz Intel Core i5 processor (CPU of a 13-inch Mac Pro 2015).

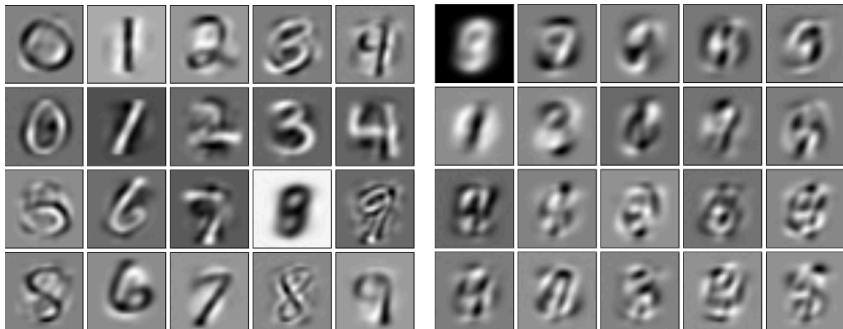
MSP versus PCA on the MNIST Dataset [LBB⁺98]

Figure: Bases learned from the MNIST dataset. Left: Some selected “meaningful” bases learned through MSP; Right: Top bases learned through PCA.

MSP versus PCA on the MNIST Dataset [LBB⁺98]



(a) Original Images from the MNIST dataset



(b) Original Images from the MNIST dataset



(c) Reconstruction with top 1 Basis by the MSP algorithm



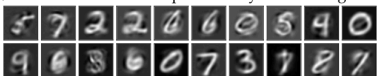
(d) Reconstruction with top 1 Basis by PCA



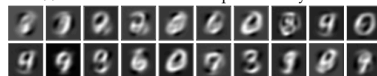
(e) Reconstruction with top 2 Bases by the MSP algorithm



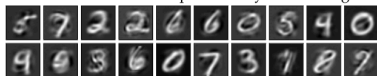
(f) Reconstruction with top 2 Bases by PCA



(g) Reconstruction with top 3 Bases by the MSP algorithm



(h) Reconstruction with top 3 Bases by PCA



(i) Reconstruction with top 4 Bases by the MSP algorithm



(j) Reconstruction with top 4 Bases by PCA

Figure: Reconstruction result comparison between MSP and PCA using different number of bases.

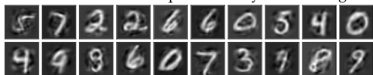
MSP versus PCA on the MNIST Dataset [LBB⁺98]



(k) Reconstruction with top 5 Bases by the MSP algorithm



(m) Reconstruction with top 10 Bases by the MSP algorithm



(o) Reconstruction with top 15 Bases by the MSP algorithm



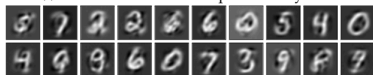
(q) Reconstruction with top 20 Bases by the MSP algorithm



(s) Reconstruction with top 25 Bases by the MSP algorithm



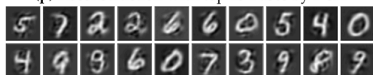
(l) Reconstruction with top 5 Bases by PCA



(n) Reconstruction with top 10 Bases by PCA



(p) Reconstruction with top 15 Bases by PCA



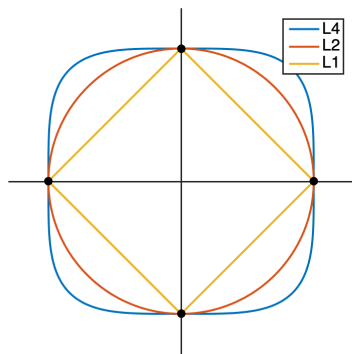
(r) Reconstruction with top 20 Bases by PCA



(t) Reconstruction with top 25 Bases by PCA

Figure: Reconstruction result comparison between MSP and PCA using different number of bases.

Generalization to Stiefel Manifold [ZMZM20]

Figure: ℓ^1 -, ℓ^2 -, and ℓ^4 -spheres in \mathbb{R}^2

Given data matrix $\mathbf{Y} \in \mathbb{R}^{n \times p}$, recall the ℓ^4 dictionary learning

$$\max_{\mathbf{A} \in \mathbf{O}(n; \mathbb{R})} \frac{1}{4} \|\mathbf{A}\mathbf{Y}\|_4^4, \quad (17)$$

where the orthogonality constraint $\mathbf{A} \in \mathbf{O}(n; \mathbb{R})$ can be viewed as *enforcing orthogonality constraint of n unit vectors*.

Generalization to Stiefel Manifold [ZMZM20]

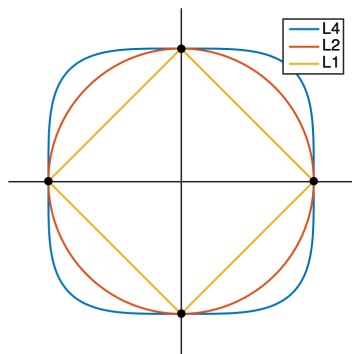


Figure: ℓ^1 -, ℓ^2 -, and ℓ^4 -spheres in \mathbb{R}^2

Given data matrix $\mathbf{Y} \in \mathbb{R}^{n \times p}$, recall the ℓ^4 dictionary learning

$$\max_{\mathbf{A} \in \mathbf{O}(n; \mathbb{R})} \frac{1}{4} \|\mathbf{A}\mathbf{Y}\|_4^4, \quad (17)$$

where the orthogonality constraint $\mathbf{A} \in \mathbf{O}(n; \mathbb{R})$ can be viewed as *enforcing orthogonality constraint of n unit vectors*.

Can we further reduce computation complexity if we are only interested in the top k ($1 \leq k \leq n$) bases?

Generalization to Stiefel Manifold

Consider generalized Dictionary Learning from orthogonal group to Stiefel manifold $\text{St}(k, n; \mathbb{R})$:⁵

$$\max_{\mathbf{W}} \frac{1}{4} \|\mathbf{W}^* \mathbf{Y}\|_4^4 \quad \text{s.t.} \quad \mathbf{W} \in \text{St}(k, n; \mathbb{R}) \subset \mathbb{R}^{n \times k}. \quad (18)$$

The MSP Algorithm can also be generalized to finding the top k bases:

$$\mathbf{W}_{t+1} = \mathcal{P}_{\text{St}(k, n; \mathbb{R})} [\nabla_{\mathbf{W}} \phi(\mathbf{W}_t)] = \mathbf{U}_t \mathbf{V}_t^*, \quad (19)$$

where $\mathbf{U}_t \Sigma_t \mathbf{V}_t^* = \text{SVD}[\mathbf{Y}(\mathbf{Y}^* \mathbf{W}_t)^{\circ 3}]$.

⁵For any $1 \leq k \leq n$, $\text{St}(k, n; \mathbb{R}) \doteq \{\mathbf{W} \in \mathbb{R}^{n \times k} : \mathbf{W}^* \mathbf{W} = \mathbf{I}_k\}$.

Relation with Geometric Interpretation of PCA

For data matrix $\mathbf{Y} \in \mathbb{R}^{n \times p}$:

- PCA aims at finding the top (k) left singular vector(s) of \mathbf{Y} :

$$\max_{\mathbf{W}} \frac{1}{2} \|\mathbf{W}^* \mathbf{Y}\|_F^2 \quad \text{s.t.} \quad \mathbf{W} \in \text{St}(k, n; \mathbb{R})$$

can be considered as finding a direction (a k -dimensional subspace) in $\text{row}(\mathbf{Y})$ where \mathbf{Y} has the largest ℓ^2 (Frobenius) norm.

- ℓ^4 -Norm maximization

$$\max_{\mathbf{W}} \frac{1}{4} \|\mathbf{W}^* \mathbf{Y}\|_4^4 \quad \text{s.t.} \quad \mathbf{W} \in \text{St}(k, n; \mathbb{R})$$

aims at finding a direction (a k -dimensional subspace) in $\text{row}(\mathbf{Y})$ where the projection of \mathbf{Y} has the largest ℓ^4 -norm.

Relation with Statistical Interpretation of PCA

View each column $\mathbf{y}_j, j \in [p]$ of data matrix \mathbf{Y} as an n dimensional random vector that are i.i.d. drawn from a distribution of random variable \mathbf{y} . Let \mathbf{Y}_c denote the centered $\mathbf{Y}: \mathbf{Y}_c \doteq \mathbf{Y} \left[\mathbf{I} - \frac{1}{p} \mathbf{1}\mathbf{1}^* \right]$. Then:

- $\max_{\mathbf{W} \in \text{St}(k, n; \mathbb{R})} \frac{1}{2} \|\mathbf{W}^* \mathbf{Y}_c\|_F^2$ finds the top k uncorrelated projections of \mathbf{y} with largest sample variance.
- $\max_{\mathbf{W} \in \text{St}(k, n; \mathbb{R})} \frac{1}{4} \|\mathbf{W}^* \mathbf{Y}_c\|_4^4$ finds the top k uncorrelated projections of \mathbf{y} with largest 4th order moments.

Relation with ICA and 4th Order Moment

In Independent Component Analysis (ICA) [HO97, HO00], finding maximizer or minimizer of *kurtosis*:

$$\text{kurt}(\mathbf{w}^* \mathbf{y}) = \mathbb{E}[\mathbf{w}^* \mathbf{y}]^4 - 3 \|\mathbf{w}\|_2^4 \quad (20)$$

can identify one independent component of \mathbf{y} .

Relation with ICA and 4th Order Moment

In Independent Component Analysis (ICA) [HO97, HO00], finding maximizer or minimizer of *kurtosis*:

$$\text{kurt}(\mathbf{w}^* \mathbf{y}) = \mathbb{E}[\mathbf{w}^* \mathbf{y}]^4 - 3 \|\mathbf{w}\|_2^4 \quad (20)$$

can identify one independent component of \mathbf{y} .

Importance of 4th Order Statistics

- The 4th order statistics carries more “abnormal” information regarding nonnormality [Hub85, DeC97, CZY17]
- The distributions of real data (images) are usually not Gaussian [LPM03, HHH09].

Fixed-Point Style Algorithms

- **PCA**

- Optimization:

$$\max_{w \in \mathbb{S}^{n-1}} \varphi(w) \doteq \frac{1}{2} \|w^* Y\|_2^2$$

- Algorithm:

$$w_{t+1} = \mathcal{P}_{\mathbb{S}^{n-1}} [\nabla_w \varphi(w_t)] = \frac{Y Y^* w_t}{\|Y Y^* w_t\|_2}$$

- **ICA**

- Optimization:

$$\max_{w \in \mathbb{S}^{n-1}} \psi(w) \doteq \frac{1}{4} \text{kurt}[w^* y] = \frac{1}{4} \mathbb{E}[w^* y]^4 - \frac{3}{4} \|w\|_2^4$$

- Algorithm:

$$w_{t+1} = \mathcal{P}_{\mathbb{S}^{n-1}} [\nabla_w \psi(w_t)] = \frac{\mathbb{E}[y(y^* w_t)^3] - 3 \|w_t\|_2^2 w_t}{\|\mathbb{E}[y(y^* w_t)^3] - 3 \|w_t\|_2^2 w_t\|_2}$$

- **DL**

- Optimization:

$$\max_{W \in \text{St}(k, n; \mathbb{R})} \phi(W) \doteq \frac{1}{4} \|W^* Y\|_4^4$$

- Algorithm:

$$W_{t+1} = \mathcal{P}_{\text{St}(k, n; \mathbb{R})} [\nabla_W \phi(W_t)] = U_t V_t^*,$$

where $U_t \Sigma_t V_t^* = \text{SVD}[Y(Y^* W)^{\circ 3}]$.

Gradient-Based Fixed Point Algorithms

Newton's Method [1669]: finding the zero x_* of a function $f(x)$ such that $f(x_*) = 0$ as a fixed point to the mapping:

$$x_{t+1} = g(x_t) = x_t - \frac{f(x_t)}{f'(x_t)}. \quad (21)$$

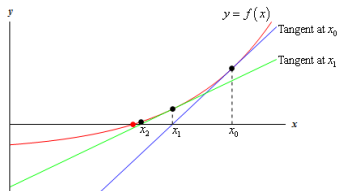
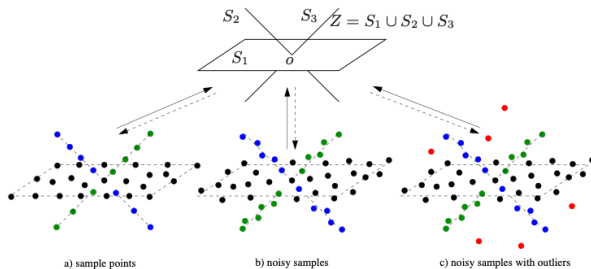


Table: PCA (Power iteration), ICA (FastICA), and DL (MSP) all are fixed-point algorithms based on **projected gradient descent**.

	Objectives	Constraint Sets	Algorithms
Power Iter.	$\varphi(\mathbf{w}) \doteq \frac{1}{2} \ \mathbf{w}^* \mathbf{Y}\ _2^2$	$\mathbf{w} \in \mathbb{S}^{n-1}$	$\mathbf{w}_{t+1} = \mathcal{P}_{\mathbb{S}^{n-1}} [\nabla_{\mathbf{w}} \varphi(\mathbf{w}_t)]$
FastICA	$\psi(\mathbf{w}) \doteq \frac{1}{4} \text{kurt}[\mathbf{w}^* \mathbf{y}]$	$\mathbf{w} \in \mathbb{S}^{n-1}$	$\mathbf{w}_{t+1} = \mathcal{P}_{\mathbb{S}^{n-1}} [\nabla_{\mathbf{w}} \psi(\mathbf{w}_t)]$
MSP	$\phi(\mathbf{W}) \doteq \frac{1}{4} \ \mathbf{W}^* \mathbf{Y}\ _4^4$	$\mathbf{W} \in \text{St}(k, n; \mathbb{R})$	$\mathbf{W}_{t+1} = \mathcal{P}_{\text{St}(k, n; \mathbb{R})} [\nabla_{\mathbf{W}} \phi(\mathbf{W}_t)]$

Different Types of Imperfect Measurements

Data are distributed around a mixture of low-dimensional subspaces or Gaussians, possibly with noise, outliers, and corruptions [Vidal, Ma, and Sastry, 2016].



Imperfect Measurements Type I: Noise

Noisy Measurements: $\mathbf{Y}_N := \mathbf{Y} + \mathbf{G}$, $\mathbf{G} \in \mathbb{R}^{n \times p}$ is matrix with $g_{i,j} \sim_{iid} \mathcal{N}(0, \eta^2)$ and $\eta > 0$ the variance of the noise.

Imperfect Measurements Type I: Noise

Noisy Measurements: $\mathbf{Y}_N := \mathbf{Y} + \mathbf{G}$, $\mathbf{G} \in \mathbb{R}^{n \times p}$ is matrix with $g_{i,j} \sim_{iid} \mathcal{N}(0, \eta^2)$ and $\eta > 0$ the variance of the noise.

Proposition (Objective with Small Noise)

$\forall \theta \in (0, 1)$, let $\mathbf{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} BG(\theta)$, $\mathbf{D}_o \in O(n; \mathbb{R})$ is any orthogonal matrix, and $\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o$. For any orthogonal matrix $\mathbf{W} \in O(n; \mathbb{R})$ and any random Gaussian matrix $\mathbf{G} \in \mathbb{R}^{n \times p}$, $g_{i,j} \sim_{iid} \mathcal{N}(0, \eta^2)$ independent of \mathbf{X}_o , let $\mathbf{Y}_N = \mathbf{Y} + \mathbf{G}$ denote the data with noise. Then the expectation of $\|\mathbf{W}^* \mathbf{Y}_N\|_4^4$ is:

$$\frac{1}{np} \mathbb{E}_{\mathbf{X}_o, \mathbf{G}} \|\mathbf{W}^* \mathbf{Y}_N\|_4^4 = 3\theta(1 - \theta) \frac{\|\mathbf{W}^* \mathbf{D}_o\|_4^4}{n} + C_{\theta, \eta},$$

where $C_{\theta, \eta}$ is a constant depending on θ and η .

Imperfect Measurements Type II: Outliers

Measurements with Outliers: $Y_O := [Y, G']$, where Y_O contains extra columns $(G' \in \mathbb{R}^{n \times \tau p})$ ⁶ that is generated from an independent Gaussian process $g'_{i,j} \sim_{iid} \mathcal{N}(0, 1)$, and τ controls the portion of the outliers, w.r.t. the clean data size p .

⁶When τp is not an integer, τp is rounded to the closest integer.

Imperfect Measurements Type II: Outliers

Measurements with Outliers: $\mathbf{Y}_O := [\mathbf{Y}, \mathbf{G}']$, where \mathbf{Y}_O contains extra columns $(\mathbf{G}' \in \mathbb{R}^{n \times \tau p})^6$ that is generated from an independent Gaussian process $g'_{i,j} \sim_{iid} \mathcal{N}(0, 1)$, and τ controls the portion of the outliers, w.r.t. the clean data size p .

Proposition (Objective with Outliers)

$\forall \theta \in (0, 1)$, let $\mathbf{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} BG(\theta)$, $\mathbf{D}_o \in \mathcal{O}(n; \mathbb{R})$ is any orthogonal matrix and $\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o$. For any orthogonal matrix $\mathbf{W} \in \mathcal{O}(n; \mathbb{R})$ and any random Gaussian matrix $\mathbf{G}' \in \mathbb{R}^{n \times \tau p}$, $g'_{i,j} \sim_{iid} \mathcal{N}(0, 1)$ independent of \mathbf{X}_o , let $\mathbf{Y}_O = [\mathbf{Y}, \mathbf{G}']$ denote the data with outliers \mathbf{G}' . Then the expectation of $\|\mathbf{W}^* \mathbf{Y}_O\|_4^4$ is:

$$\frac{1}{np} \mathbb{E}_{\mathbf{X}_o, \mathbf{G}'} \|\mathbf{W}^* \mathbf{Y}_O\|_4^4 = 3\theta(1 - \theta) \frac{\|\mathbf{W}^* \mathbf{D}_o\|_4^4}{n} + C_\theta,$$

where C_θ is a constant depending on θ .

⁶When τp is not an integer, τp is rounded to the closest integer.

Imperfect Measurements Type III: Corruptions

Measurements with Sparse Corruptions: $\mathbf{Y}_C := \mathbf{Y} + \sigma \mathbf{B} \circ \mathbf{S}$, where $\sigma > 0$ controls the scale of corrupting entries, $\mathbf{B} \in \mathbb{R}^{n \times p}$ is a Bernoulli matrix with $b_{i,j} \sim_{iid} \text{Ber}(\beta)$, where $\beta \in (0, 1)$ controls the ratio of the sparse corruptions, and entries $s_{i,j}$ of $\mathbf{S} \in \mathbb{R}^{n \times p}$ are i.i.d. drawn from a *Rademacher* distribution:

$$s_{i,j} = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}.$$

Imperfect Measurements Type III: Corruptions

Measurements with Sparse Corruptions: $\mathbf{Y}_C := \mathbf{Y} + \sigma \mathbf{B} \circ \mathbf{S}$, where $\sigma > 0$ controls the scale of corrupting entries, $\mathbf{B} \in \mathbb{R}^{n \times p}$ is a Bernoulli matrix with $b_{i,j} \sim_{iid} \text{Ber}(\beta)$, where $\beta \in (0, 1)$ controls the ratio of the sparse corruptions, and entries $s_{i,j}$ of $\mathbf{S} \in \mathbb{R}^{n \times p}$ are i.i.d. drawn from a *Rademacher* distribution:

$$s_{i,j} = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}.$$

Proposition (Objective with Sparse Corruptions)

$\forall \theta \in (0, 1)$, let $\mathbf{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} BG(\theta)$, $\mathbf{D}_o \in \mathcal{O}(n; \mathbb{R})$ is any orthogonal matrix and $\mathbf{Y} = \mathbf{D}_o \mathbf{X}_o$. For any orthogonal matrix $\mathbf{W} \in \mathcal{O}(n; \mathbb{R})$ and any random Bernoulli matrix $\mathbf{B} \in \mathbb{R}^{n \times p}$, $b_{i,j} \sim_{iid} \text{Ber}(\beta)$ independent of \mathbf{X}_o , let $\mathbf{Y}_C = \mathbf{Y} + \sigma \mathbf{B} \circ \mathbf{S}$ denote the data with sparse corruptions, and $\mathbf{S} \in \mathbb{R}^{n \times p}$ is defined as (57). Then the expectation of $\|\mathbf{W}^* \mathbf{Y}_C\|_4^4$ is:

$$\frac{1}{np} \mathbb{E}_{\mathbf{X}_o, \mathbf{B}, \mathbf{S}} \|\mathbf{W}^* \mathbf{Y}_C\|_4^4 = 3\theta(1 - \theta) \frac{\|\mathbf{W}^* \mathbf{D}_o\|_4^4}{n} + \sigma^4 \beta(1 - 3\beta) \frac{\|\mathbf{W}\|_4^4}{n} + C_{\theta, \sigma, \beta},$$

where $C_{\theta, \sigma, \beta}$ is a constant depending on θ, σ and β .

Numerical Experiments I

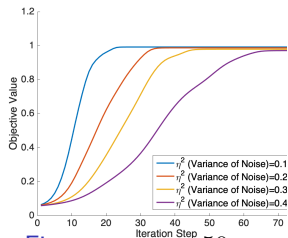


Figure: $n = 50, p = 20,000, \theta = 0.3$, varying η^2 from 0.1 to 0.4

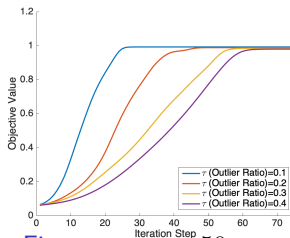


Figure: $n = 50, p = 20,000, \theta = 0.3$, varying τ from 0.1 to 0.4

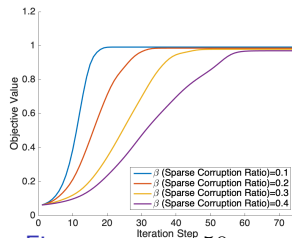


Figure: $n = 50, p = 20,000, \theta = 0.3, \sigma = 1$, varying β from 0.1 to 0.4

Figure: Normalized $\|\mathbf{W}^* \mathbf{D}_o\|_4^4 / n$ of the MSP algorithm for dictionary learning, using imperfect measurements $\mathbf{Y}_N, \mathbf{Y}_O, \mathbf{Y}_C$, respectively.

Numerical Experiments II

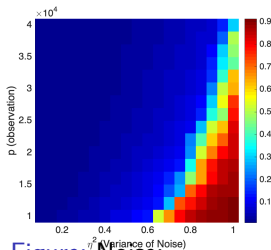


Figure: Noise:
 $n = 50, \theta = 0.3$

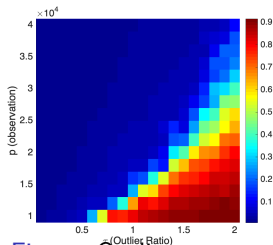


Figure: Outliers:
 $n = 50, \theta = 0.3$

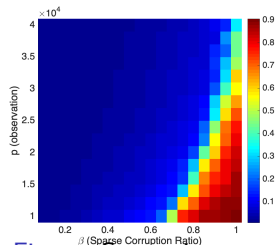


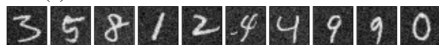
Figure: Corruptions:
 $n = 50, \theta = 0.3$

Figure: Average normalized error $|1 - \|\mathbf{W}^* \mathbf{D}_o\|_4^4 / n|$ of 10 random trials for the MSP Algorithm: **(a)** Varying sample size p and variance of noise η^2 ; **(b)** Varying sample size p and Gaussian Outlier ratio τ ; **(c)** Varying sample size p and sparse corruption ratio β , with fixed $\sigma = 1$.

Real Image Data: MNIST



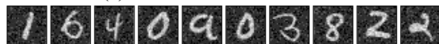
(a) Normalized MNIST to mean 0 and 1 std



(c) MNIST with noise, $\text{SNR} = 3.333$



(e) MNIST with 50% outliers



(g) MNIST with 50% sparse corruptions



(b) Top bases from MNIST



(d) Top bases from MNIST with noise



(f) Top bases from MNIST with outliers



(h) Top bases from MNIST with sparse corruptions

Figure: Top Bases learned from imperfect measurements of MNIST.

Real Image Data: Single Image

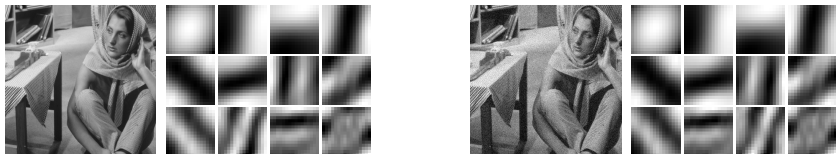


Figure: The top 12 bases learned from all 16×16 patches of Barbara, both with (right) and without (left) Gaussian noise. The noisy image is produced by adding Gaussian noise to the clean image, resulting in SNR of 5.87.



Figure: The top 12 bases learned from all $8 \times 8 \times 3$ color patches of the clean and noisy image, respectively. Here, the SNR of the noisy image is 6.56.

Real Image Data: Single Image

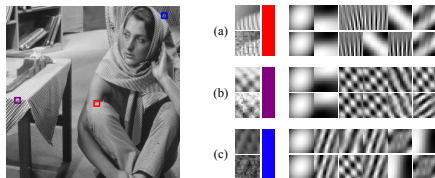


Figure: Representations of three 16×16 patches from Barbara w/ and w/o noise. Each selected patch is visualized, both w/ and w/o noise, and the top 6 corresponding bases are shown.

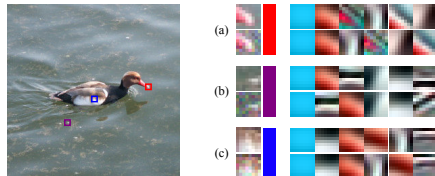


Figure: Representations of three $8 \times 8 \times 3$ patches from duck w/ and w/o noise. Each selected patch is visualized, both w/ and w/o noise, and the top 6 corresponding bases are shown.

Real Image Data: CIFAR-10

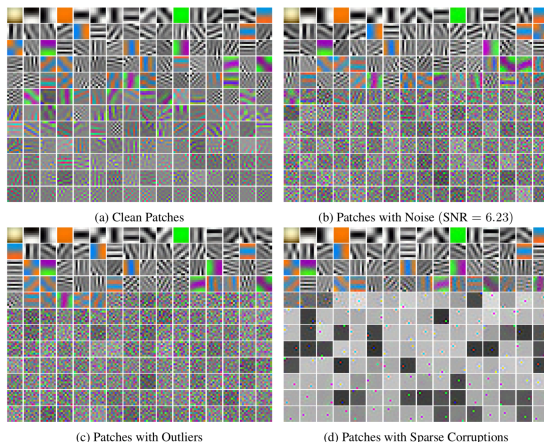


Figure: All $8 \times 8 \times 3 = 192$ bases learned from 100,000 random 8×8 colored patches sampled from the CIFAR-10 data-set. (a) Learned Bases from clean CIFAR-10; (b) Learned Bases from CIFAR-10 with Gaussian noise, SNR = 6.23; (c) Learned Bases from CIFAR-10 with 20% of Gaussian outliers; (d) Learned Bases from CIFAR-10 with 50% of sparse corruptions.

Summary

[ZYL⁺19]:

- The MSP algorithm solves complete dictionary learning **holistically**.
- The sample complexity $\Omega(n^2 \ln n)$ corroborates with experiments.
- Special **symmetries** help nonconvex optimization.

[ZMZM20]:

- The MSP algorithm is a **fixed-point** type algorithm just like Power-iteration [Jol11] and FastICA [HO97].
- The MSP algorithm is robust to stable to noise, robust to outliers and resilient to sparse corruptions.

A Low-Dimensional Subspace

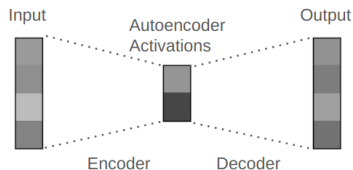
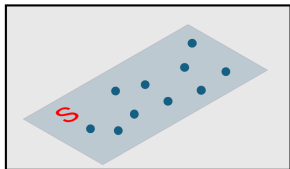
Canonical example:

subspace $U \in \mathbb{R}^{D \times d}$ ($d \leq D$):

$$\mathbf{x} \xrightarrow{f=U^\top} \mathbf{z} \xrightarrow{g=U} \hat{\mathbf{x}}$$

Principal component analysis:

$$\min_U \mathbb{E} \left[\|\mathbf{x} - UU^\top \mathbf{x}\|_2^2 \right]$$



Sparsity and Sparse Coding

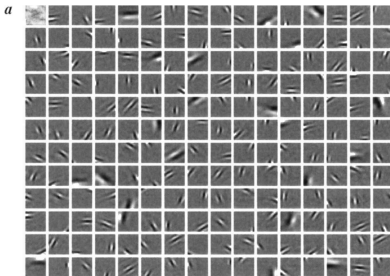
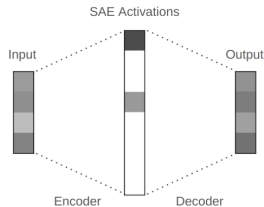
A significant generalization: unions of subspaces

ℓ^0 “norm”: **# of nonzero entries**, $\|x\|_0 = |\{i \mid x_i \neq 0\}|$.

Given (learned) $A \in \mathbb{R}^{D \times d}$ ($d \gg D$), represent x as a sparse code:

$$f(x) = \arg \min_z \|z\|_0 \quad \text{s.t.} \quad x = Az$$

$$x \xrightarrow{f} z \xrightarrow{g=A} \hat{x}$$



How to Learn: Optimization for Low-Dim Structures

We can compute sparse coding with (proximal) gradient descent

$$f(\mathbf{x}) = \min_{\mathbf{z} \geq 0} \|\mathbf{x} - \mathbf{A}\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_1$$

- ① Given the current code \mathbf{z}^ℓ , gradient descent to better fit \mathbf{x} ;
- ② Without moving too much, sparsify the updated code

Then $f(\mathbf{x}) = \mathbf{z}^\infty$, where

$$\mathbf{z}^{\ell+1} = \text{ReLU} \left(\eta \mathbf{A}^\top \mathbf{x} + \left(\mathbf{I} - \eta \mathbf{A}^\top \mathbf{A} \right) \mathbf{z}^\ell - \lambda \eta \mathbf{1} \right)$$

Unrolled Optimization: From Objectives to Deep Networks

Recall the sparse coding objective:

$$f(\mathbf{x}) = \min_{\mathbf{z} \geq 0} \|\mathbf{x} - \mathbf{A}\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_1$$

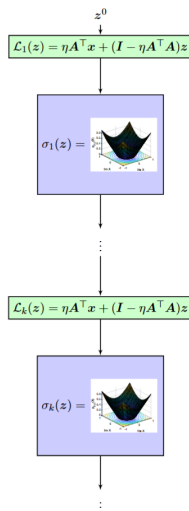
Then $f(\mathbf{x}) = z^\infty$, where

$$z^{\ell+1} = \text{ReLU} \left(\eta \mathbf{A}^\top \mathbf{x} + \left(\mathbf{I} - \eta \mathbf{A}^\top \mathbf{A} \right) z^\ell - \lambda \eta \mathbf{1} \right)$$

Truncate the network, and **learn** its parameters using data.

This approach is called LISTA [Gregor and Lecun, 2010].

⇒ each layer learns its own dictionary!



More Supporting Materials

A Textbook:

High-Dim Analysis with Low-Dim Models

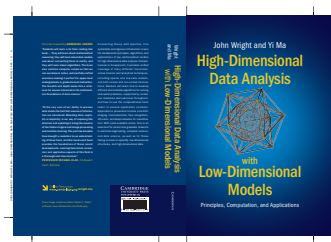
<https://book-wright-ma.github.io/>

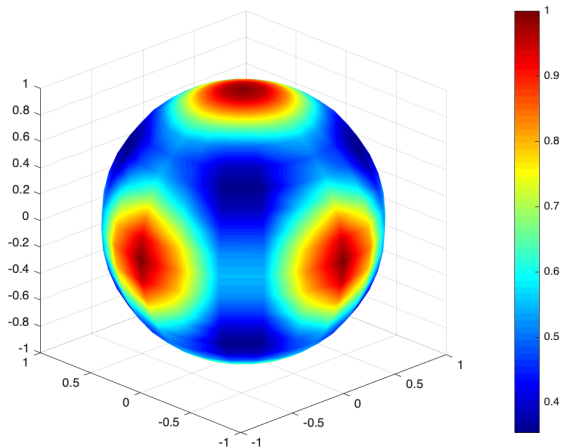
A Course:

Berkeley EECS208/HKU DATA8001

https://pages.github.berkeley.edu/UCB-EECS208/course_site/

<https://book-wright-ma.github.io/Lecture-Slides/>





Thanks! & Questions?

References I



Michal Aharon, Michael Elad, Alfred Bruckstein, et al.

K-svd: An algorithm for designing overcomplete dictionaries for sparse representation.
IEEE Transactions on signal processing, 54(11):4311, 2006.



Yu Bai, Qijia Jiang, and Ju Sun.

Subgradient descent learns orthogonal dictionaries.
arXiv preprint arXiv:1810.10702, 2018.



Boaz Barak, Jonathan Kelner, and David Steurer.

Dictionary learning and tensor decomposition via the sum-of-squares method.
In *STOC*, 2015.



Meghan K Cain, Zhiyong Zhang, and Ke-Hai Yuan.

Univariate and multivariate skewness and kurtosis for measuring nonnormality: Prevalence, influence and estimation.
Behavior Research Methods, 49(5):1716–1735, 2017.



Lawrence T DeCarlo.

On the meaning and use of kurtosis.
Psychological methods, 2(3):292, 1997.



Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer.

Natural image statistics: A probabilistic approach to early computational vision., volume 39.
Springer Science & Business Media, 2009.



Aapo Hyvärinen and Erkki Oja.

A fast fixed-point algorithm for independent component analysis.
Neural Computation, 9:1483–1492, 1997.

References II



Aapo Hyvärinen and Erkki Oja.

Independent component analysis: algorithms and applications.
Neural networks, 13(4-5):411–430, 2000.



Peter J Huber.

Projection pursuit.
The annals of Statistics, pages 435–475, 1985.



Ian Jolliffe.

Principal component analysis.
Springer, 2011.



Palle Jorgensen.

<http://homepage.divms.uiowa.edu/~jorgen/Haar.html>, 2006.



Yanjun Li and Yoram Bresler.

Global geometry of multichannel sparse blind deconvolution on the sphere.
In Advances in Neural Information Processing Systems, pages 1132–1143, 2018.



Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al.

Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324, 1998.



Ann B Lee, Kim S Pedersen, and David Mumford.

The nonlinear statistics of high-contrast patches in natural images.
International Journal of Computer Vision, 54(1-3):83–103, 2003.

References III



Jiang-Hua Lu.

A note on a theorem of Stanton-Weinstein on the L_4 -norm of spherical harmonics.

In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 102, page 561, 1987.



Tengyu Ma, Jonathan Shi, and David Steurer.

Polynomial-time tensor decompositions with sum-of-squares.

In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 438–446. IEEE, 2016.



[pipo1995_2.](https://www.taringa.net/+info/como-una-foto-de-una-playboy-se-convirtio-en-el-formato-jpg_1ejzk6)

https://www.taringa.net/+info/como-una-foto-de-una-playboy-se-convirtio-en-el-formato-jpg_1ejzk6, 2018.



Ju Sun, Qing Qu, and John Wright.

Complete dictionary recovery over the sphere i: Overview and the geometric picture.

IEEE Transactions on Information Theory, 63(2):853–884, 2017.



Tselil Schramm and David Steurer.

Fast and robust tensor decomposition with applications to dictionary learning.

arXiv preprint arXiv:1706.08672, 2017.



Robert J Stanton and Alan Weinstein.

On the L_4 norm of spherical harmonics.

In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 89, page 343, 1981.



Daniel A Spielman, Huan Wang, and John Wright.

Exact recovery of sparsely-used dictionaries.

In *Conference on Learning Theory*, pages 37–1, 2012.

References IV



Yuqian Zhang, Han-Wen Kuo, and John Wright.

Structured local optima in sparse blind deconvolution.
arXiv preprint arXiv:1806.00338, 2018.



Yuexiang Zhai, Hermish Mehta, Zhengyuan Zhou, and Yi Ma.

Understanding l4-based dictionary learning: Interpretation, stability, and robustness.
In International Conference on Learning Representations (ICLR), 2020.



Yuexiang Zhai, Zitong Yang, Zhenyu Liao, John Wright, and Yi Ma.

Complete dictionary learning via ℓ^4 -norm maximization over the orthogonal group.
arXiv preprint arXiv:1906.02435, 2019.