# Learning Deep Low-Dimensional Models from High-Dimensional Data: From Theory to Practice

(Deductive Approaches to Analytical Low-Dimensional Models)

**Professor Yi Ma**

University of Hong Kong

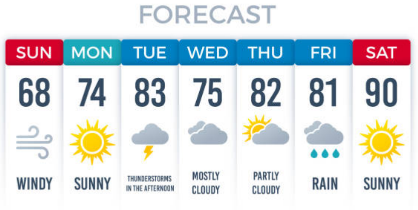
October 19, 2025

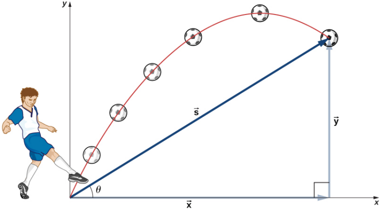
*"Everything should be made as simple as possible, but not any simpler."*

*– Albert Einstein*

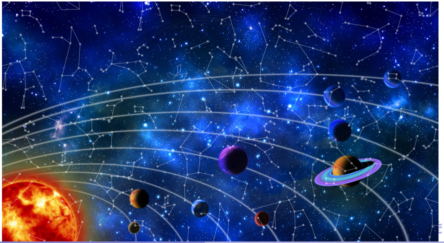
# Science of Intelligence

**Objective of Intelligence:** Learn what is predictable of the external world from sensed data.



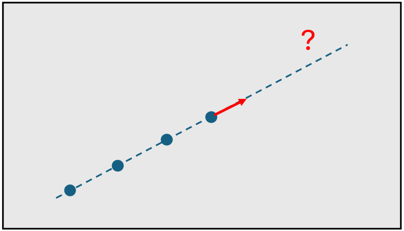



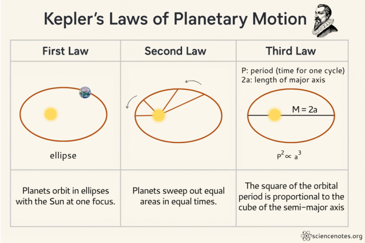

Schema huius præmissæ diuisionis Sphærarum.

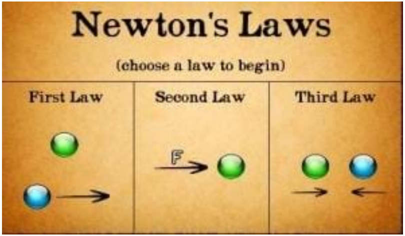


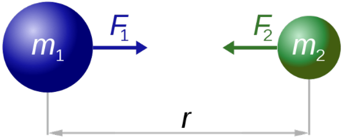
# Scientific Knowledge

How to predict the motion of an object (say a planet), analytically?




$$F_1 = F_2 = G\frac{m_1 \times m_2}{r^2}$$

# 17 Equations that Changed the World

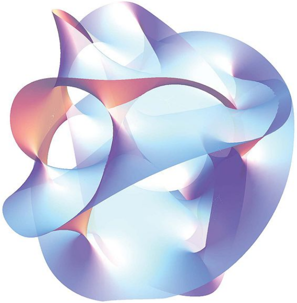
**Physically feasible** phenomena
are always enforced to be on
a low-dimensional space via equations,
**at all scale of space and time.**

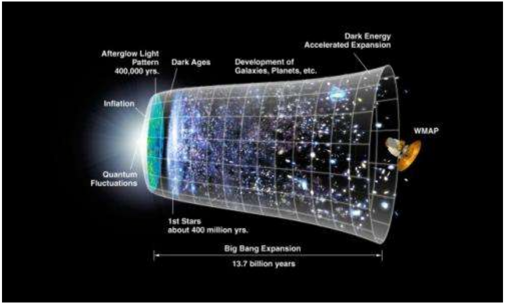
### 17 Equations That Changed the World
by Ian Stewart

| | | | |
|---|---|---|---|
| 1. | **Pythagoras's Theorem** | $a^2+b^2=c^2$ | Pythagoras,530 BC |
| 2. | **Logarithms** | $\log xy = \log x + \log y$ | John Napier, 1610 |
| 3. | **Calculus** | $\frac{df}{dt} = \lim_{h\to 0} \frac{f(t+h)-f(t)}{h}$ | Newton, 1668 |
| 4. | **Law of Gravity** | $F = G\frac{m_1 m_2}{r^2}$ | Newton, 1687 |
| 5. | **The Square Root of Minus One** | $i^2=-1$ | Euler, 1750 |
| 6. | **Euler's Formula for Polyhedra** | $V-E+F=2$ | Euler, 1751 |
| 7. | **Normal Distribution** | $\Phi(x) = \frac{1}{\sqrt{2\pi\rho}} e^{\frac{(x-\mu)^2}{2\rho^2}}$ | C.F. Gauss, 1810 |
| 8. | **Wave Equation** | $\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$ | J. d'Almbert, 1746 |
| 9. | **Fourier Transform** | $f(\omega) = \int_{\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$ | J. Fourier, 1822 |
| 10. | **Navier-Stokes Equation** | $\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \nabla\cdot\mathbf{T} + \mathbf{f}$ | C. Navier, G. Stokes, 1845 |
| 11. | **Maxwell's Equations** | $\nabla\cdot\mathbf{E} = \frac{\rho}{\varepsilon_0}$, $\nabla\cdot\mathbf{H}=0$, $\nabla\times\mathbf{E} = -\frac{1}{c}\frac{\partial \mathbf{H}}{\partial t}$, $\nabla\times\mathbf{H} = \frac{1}{c}\frac{\partial E}{\partial t}$ | J.C. Maxwell, 1865 |
| 12. | **Second Law of Thermodynamics** | $dS \geq 0$ | L. Boltzmann, 1874 |
| 13. | **Relativity** | $E=mc^2$ | Einstein, 1905 |
| 14. | **Schrodinger's Equation** | $i\hbar\frac{\partial}{\partial t}\Psi = H\Psi$ | E. Schrodinger, 1927 |
| 15. | **Information Theory** | $H = -\sum p(x)\log p(x)$ | C. Shannon, 1949 |
| 16. | **Chaos Theory** | $x_{t+1} = kx_t(1-x_t)$ | Robert May, 1975 |
| 17. | **Black-Scholes Equation** | $\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} - rV = 0$ | F. Black, M. Scholes, 1990 |

# The 10-Dimensional Universe

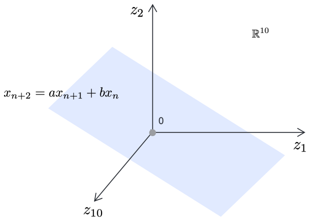
The Calabi-Yau manifold and a 10-dimensional model for the universe based on the string theory (that unifies almost everything we know so far).

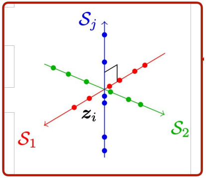


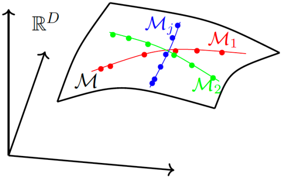


# Intelligence: A Ubiquitous Mathematical Problem

**Mathematically**, all predictable information is encoded as a distribution $p(\boldsymbol{x})$ of low-dimensional supports in observed high-dimensional data space.




*"Entities should not be multiplied without necessity."*
– William of Ockham, *Law of Parsimony*

# Importance of Low-Dimensionality: Completion

Allow us to effectively conduct (empirical or analytical) Bayesian inference based on partial, noisy, and corrrupted observations:

$$\boldsymbol{y} = \mathcal{P}_{\Omega}(\boldsymbol{x}) + \boldsymbol{n} \rightarrow \hat{\boldsymbol{x}} \sim p(\boldsymbol{x} \mid \boldsymbol{y}).$$

Partial observations

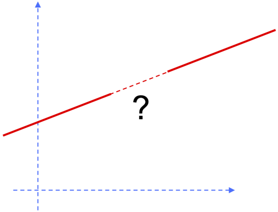
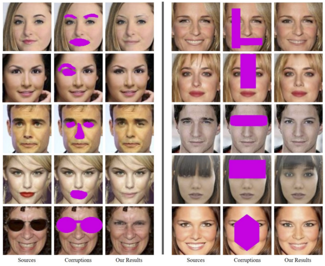
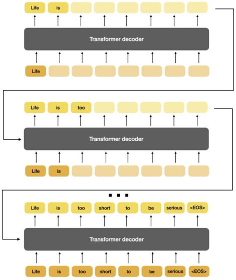



Image completion




Text prediction (GPT)



# Importance of Low-Dimensionality: Denoise

Allow us to effectively conduct (empirical or analytical) Bayesian inference based on partial, noisy, and corrrupted observations:

$$\boldsymbol{y} = \mathcal{P}_\Omega(\boldsymbol{x}) + \boldsymbol{n} \rightarrow \hat{\boldsymbol{x}} \sim p(\boldsymbol{x} \mid \boldsymbol{y}).$$

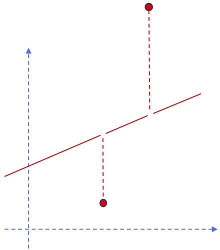
Properties of low-dimensional structures: Denoising

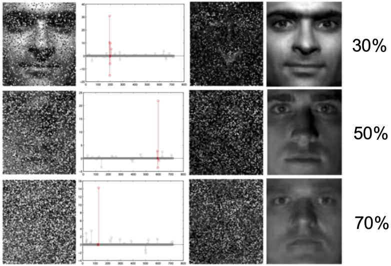
# Importance of Low-Dimensionality: Error Correction

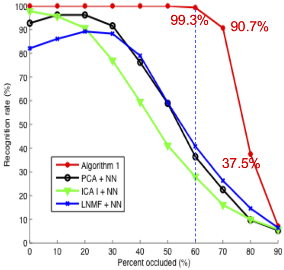
Allow us to effectively conduct Bayesian inference based on partial, noisy, and corrupted observations:

$$\boldsymbol{y} = \mathcal{P}_{\Omega}(\boldsymbol{x}) + \boldsymbol{n} \rightarrow \hat{\boldsymbol{x}} \sim p(\boldsymbol{x} \mid \boldsymbol{y}).$$

Properties of low-dimensional structures: Error Correction

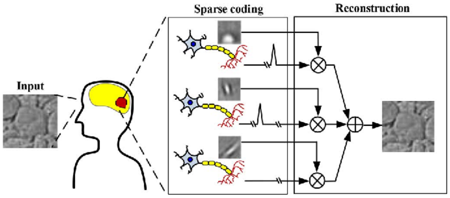


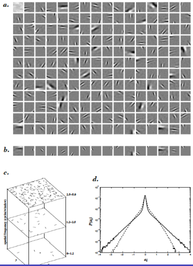


# History: Nature and Neuroscience

**Dogma for natural vision** [Barlow 1972]: *"... to represent the input as completely as possible by activity in as few neurons as possible."*



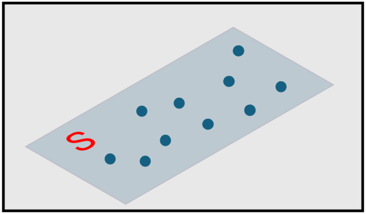
Find sparse $\{x_i\}$ such that

$$\boldsymbol{y} = \sum_{i=1}^{n} x_i \boldsymbol{a}_i + \boldsymbol{\epsilon} \quad \in \mathbb{R}^m, \tag{1}$$

[Nature, Olshausen and Field 1996.]

# 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]:

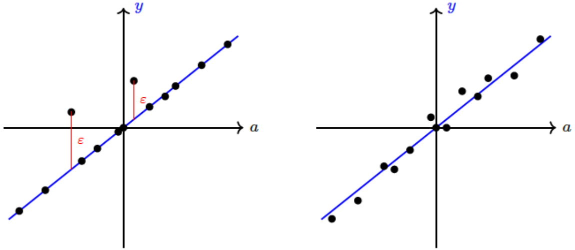



Figure: Left: $\epsilon \sim \frac{1}{2b} \exp\left(-\frac{|\epsilon|}{b}\right)$; Right: $\epsilon \sim \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$

# Geometry: Modeling Data with Linear Structures

Model $y$ as a **linear function** (regression) of variables $a_1, \ldots, a_n$:

$$y = f(\boldsymbol{a}) = \boldsymbol{a}^* \boldsymbol{x} = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n, \tag{2}$$

from noisy measurements:

$$y_i = \boldsymbol{a}_i^* \boldsymbol{x} + \epsilon_i, \quad i = 1, 2, \ldots, m, \tag{3}$$

where $\epsilon_i$ is possible measurement noise or error.

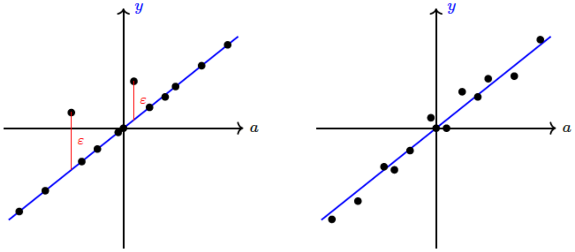



Figure: Left: $\epsilon \sim \frac{1}{2b} \exp\left(-\frac{|\epsilon|}{b}\right)$; Right: $\epsilon \sim \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$

# Algebra: Low-Rank Matrix Approximation

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

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

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

$$\text{Given a matrix of samples:} \quad \boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n] \quad \in \mathbb{R}^{m \times n}. \tag{4}$$

find a matrix $\boldsymbol{X}_\star$ such that

$$\boldsymbol{X}_\star = \arg\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_2^2 \quad \text{subject to} \quad \text{rank}(\boldsymbol{X}) \leq d. \tag{5}$$

# Statistics: Principal Component Analysis (PCA)

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

$$\boldsymbol{y} = \boldsymbol{u}_1 w_1 + \boldsymbol{u}_2 w_2 + \cdots + \boldsymbol{u}_d w_d + \boldsymbol{\epsilon} \doteq \boldsymbol{U}\boldsymbol{w} + \boldsymbol{\epsilon} \quad \in \mathbb{R}^m, \tag{6}$$

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

$$\min_{\boldsymbol{U}_d, \boldsymbol{w}} \mathbb{E}\big[\|\boldsymbol{y} - \boldsymbol{U}_d \boldsymbol{w}\|_2^2\big]. \tag{7}$$

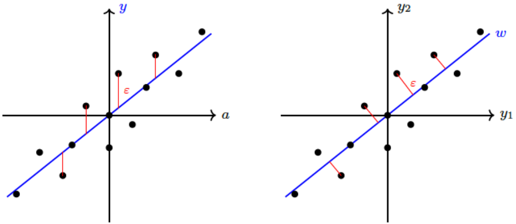



Figure: Left: linear regression; Right: principal component analysis.

# Theorem: Singular Value Decomposition (SVD)

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

$$\boldsymbol{A} = \boldsymbol{U}_r \Sigma_r \boldsymbol{V}_r^\top = [\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_r \end{bmatrix} \begin{bmatrix} \vec{v}_1^\top \\ \vec{v}_2^\top \\ \vdots \\ \vec{v}_r^\top \end{bmatrix}, \tag{8}$$

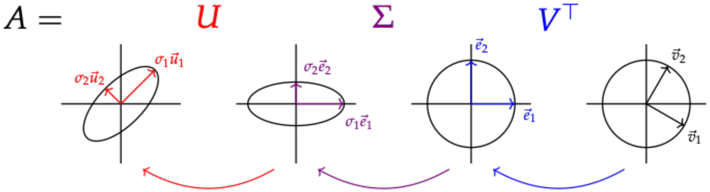
where

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

# Visualization of SVD

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

$$\boldsymbol{A} = \boldsymbol{U}_r \Sigma_r \boldsymbol{V}_r^\top = [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_r] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_r \end{bmatrix} \begin{bmatrix} \vec{v}_1^\top \\ \vec{v}_2^\top \\ \vdots \\ \vec{v}_r^\top \end{bmatrix}. \tag{9}$$



# Theorem: How to Compute SVD

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

$$\boldsymbol{A}^\top \boldsymbol{A} = \sum_{i=1}^{r} \lambda_i \vec{v}_i \vec{v}_i^\top = \boldsymbol{V}_r \Lambda_r \boldsymbol{V}_r^T \quad \in \mathbb{R}^{n\times n} \tag{10}$$

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

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

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

# How to Compute SVD: Power Iteration

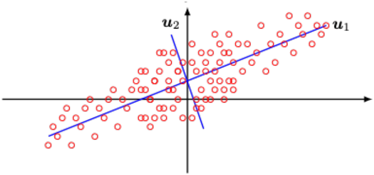
**The Power Iteration Algorithm:**

- Construct the real symmetric matrix $\boldsymbol{M} \doteq \boldsymbol{A}^\top \boldsymbol{A} \in \mathbb{R}^{n \times n}$.
- Draw a random vector $\boldsymbol{z}_0 \in \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{n\times n}) \in \boldsymbol{R}^n$.
- For $t = 0, 1, 2, \ldots,$ **compute iteratively**:

$$\boldsymbol{z}_{t+1} \leftarrow \frac{\boldsymbol{M}\boldsymbol{z}_t}{\|\boldsymbol{M}\boldsymbol{z}_t\|_2} \tag{11}$$

till $\{\boldsymbol{z}_t\}$ converges to the first singular vector $\boldsymbol{u}_1$.

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 $\boldsymbol{y}$: $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n] \in \mathbb{R}^{m \times n}$, let $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be its Singular Value Decomposition (**SVD**):

$$\boldsymbol{Y} = \sum_{i=1}^{r} \sigma_i \vec{u}_i \vec{v}_i^\top = \sum_{i=1}^{d} \sigma_i \vec{u}_i \vec{v}_i^\top + \sum_{i=d+1}^{r} \sigma_i \vec{u}_i \vec{v}_i^\top. \tag{12}$$

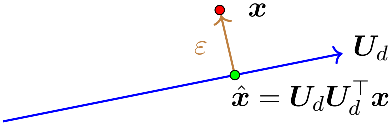
Then the best rank-$d$ approximation to $\boldsymbol{Y}$ is given by:

$$\boldsymbol{X}_\star = \sum_{i=1}^{d} \sigma_i \vec{u}_i \vec{v}_i^\top \doteq \boldsymbol{U}_d \boldsymbol{\Sigma}_d \boldsymbol{V}_d^\top. \tag{13}$$

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

# Denoising Against a Linear Subspace

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



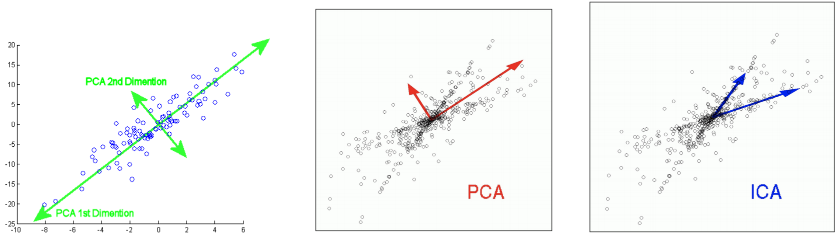

Interpretation as a **two-layer** "deep" network:

$$\mathrm{denoise}(\boldsymbol{x}) = \underbrace{\boldsymbol{U}_\star \circ \underbrace{\mathrm{id} \circ \underbrace{\boldsymbol{U}_\star^\top \boldsymbol{x}}_{\text{first ``layer''}}}_{\text{post-activation of first ``layer''}}}_{\text{output of ``NN''}}$$

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

# A Classic Example: Independent Component Analysis

[Ans, Hérault, and Jutten 1985]




The observed random variable $\boldsymbol{x}$ is **a linear superposition of multiple independent components** $z_i$ (ICA [Oja and Hyvärinen, 1990s]):

$$\boldsymbol{y} = \boldsymbol{d}_1 z_1 + \boldsymbol{d}_2 z_2 + \cdots + \boldsymbol{d}_d z_d + \boldsymbol{\epsilon} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{\epsilon}. \tag{14}$$

where $x_i$ are assumed to be independent *non-Gaussian* variables, say

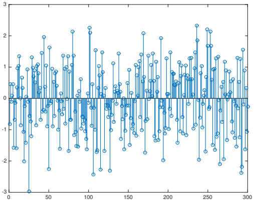
$$x_i = \sigma_i \cdot w_i, \quad \sigma_i \sim B(1, p). \tag{15}$$

# Dictionary Learning: General Case

**A Fundamental Problems in Data Analysis:**
Given an $n$-dimensional signal: $\boldsymbol{y} \in \mathbb{R}^n$, find a transformation $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^m$ or its "inverse" $\boldsymbol{D} : \mathbb{R}^m \to \mathbb{R}^n$, such that

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

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




$\boldsymbol{y} \Rightarrow \boldsymbol{x}$

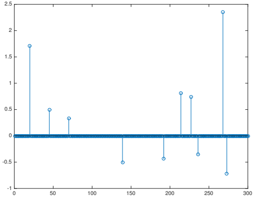



Figure: **Sparse Representation** Left: a *generic* vector $\boldsymbol{y} \in \mathbb{R}^n$, Right: a *sparse* representation $\boldsymbol{x} = \mathcal{T}[\boldsymbol{y}]$, after a proper transformation $\mathcal{T}$.

# Introduction: History of Finding Good Transform

Figure: Joseph Fourier, $1768 - 1830$

- **Fourier Transform $\boldsymbol{D} = \boldsymbol{F}$**
- Wavelet Transform $\boldsymbol{D} = \boldsymbol{W}$
- Dictionary Learning

# Introduction: Fourier Transform

**Assumption:**
The signal $\boldsymbol{y}$ is **band-limited and sparse** in frequency domain: $y_k = \sum_{l=0}^{n-1} x_l \cdot e^{-\frac{i2\pi}{n}kl}$ ($\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x}$.)

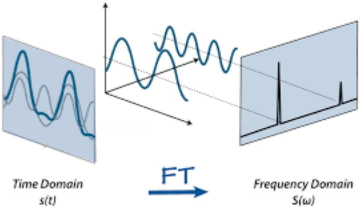



Figure: Fourier Transform


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

# Introduction: History of Finding Good Transform

Figure: Alfred Haar, $1855 - 1933$

- Fourier Transform $\boldsymbol{D} = \boldsymbol{F}$
- **Wavelet Transform** $\boldsymbol{D} = \boldsymbol{W}$
- Dictionary Learning

# Introduction: Wavelet Transform

**Assumption:**

Signal $\boldsymbol{y}$ is piece-wise smooth, scale-invariant, etc: $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$, $\boldsymbol{W}^*\boldsymbol{W} = \boldsymbol{I}$.

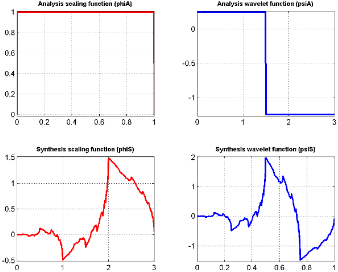



Figure: Haar & Daubechies Wavelets

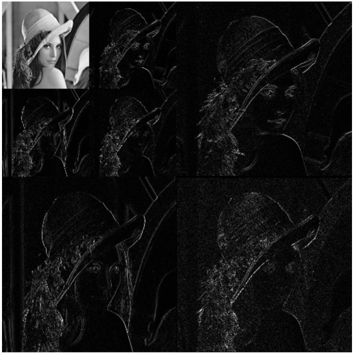


Figure: Lena Compression using Wavelet Transform (JPEG2000) [Jor06]

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

$$\boldsymbol{y}_i = \boldsymbol{D}\boldsymbol{x}_i, \quad \forall i \in [p], \tag{16}$$

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

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

# 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 $\boldsymbol{x}$.
- **Signed permutation ambiguities**
  $\forall \boldsymbol{P} \in \mathsf{SP}(m)$, [1] $(\boldsymbol{D}_\star \boldsymbol{P}, \boldsymbol{P}^* \boldsymbol{X}_\star)$ and $(\boldsymbol{D}_\star, \boldsymbol{X}_\star)$ are equally sparse.

[1] $\mathsf{SP}(m)$ 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 $\boldsymbol{x}$.
- **Signed permutation ambiguities**
  $\forall \boldsymbol{P} \in \mathsf{SP}(m)$, [1] $(\boldsymbol{D}_\star \boldsymbol{P}, \boldsymbol{P}^* \boldsymbol{X}_\star)$ and $(\boldsymbol{D}_\star, \boldsymbol{X}_\star)$ are equally sparse.

**Some heuristic algorithms:**

- K-SVD [AEB+06]
- Alternative Direction Methods [SQW17]

[1] $\mathsf{SP}(m)$ 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 $\boldsymbol{x}$.
- **Signed permutation ambiguities**
  $\forall \boldsymbol{P} \in \mathsf{SP}(m)$, [1] $(\boldsymbol{D}_\star \boldsymbol{P}, \boldsymbol{P}^* \boldsymbol{X}_\star)$ and $(\boldsymbol{D}_\star, \boldsymbol{X}_\star)$ are equally sparse.

**Some heuristic algorithms:**

- K-SVD [AEB+06]
- Alternative Direction Methods [SQW17]

**Learn the dictionary with tractable algorithms and sample size?**

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

# Complete Dictionary Learning

**A Random Model:**

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

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

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

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

**Preconditioning:**

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

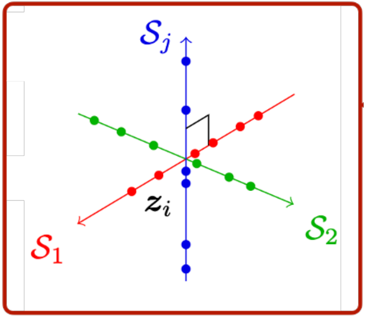
$$\bar{\boldsymbol{Y}} \leftarrow \left(\frac{1}{p\theta} \boldsymbol{Y} \boldsymbol{Y}^*\right)^{-\frac{1}{2}} \boldsymbol{Y} = \boldsymbol{D}_o \boldsymbol{X}_o, \quad \text{with} \quad \boldsymbol{D}_o \in O(n).$$

[2] square and invertible

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

# 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 $\boldsymbol{Y}$ is generated by a complete **orthogonal dictionary** $\boldsymbol{D}_o$ and **sparse coefficients** $\boldsymbol{X}_o$:

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

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

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

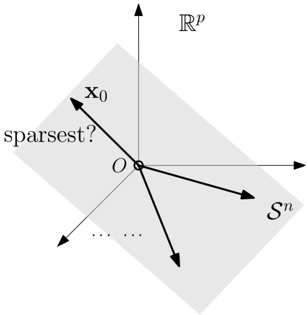
**Reduced to find the sparsest direction in a subspace:**

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

# Complete Dictionary Learning – Prior Arts

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

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




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

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

Or maximize the $\ell^4$ norm (**this lecture**):

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

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

# Intuition for $\ell^1$ and $\ell^4$ Norm

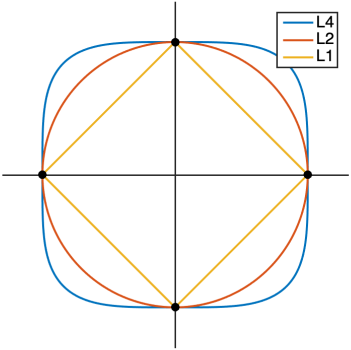



Figure: $\ell^1$-,$\ell^2$-, and $\ell^4$-spheres in $\mathbb{R}^2$

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

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

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

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

# Intuition for $\ell^4$ Norm Maximization [ZYL+19]

Consider finding the whole dictionary by the following nonconvex program:

$$\max_{\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})} f(\boldsymbol{A}) = \|\boldsymbol{A}\boldsymbol{Y}\|_4^4, \tag{17}$$

which is equivalent to

$$\max_{\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})} \|\boldsymbol{X}\|_4^4, \quad \text{s.t.} \quad \boldsymbol{Y} = \boldsymbol{A}^* \boldsymbol{X}, \tag{18}$$

where maximizing $\ell^4$ norm with spherical constraints is promoting "spikiness" [ZKW18].

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

$$\frac{1}{3p\theta}\mathbb{E}_{\boldsymbol{X}_o}\|\boldsymbol{A}\boldsymbol{Y}\|_4^4 = (1-\theta)\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4 + \theta n. \tag{19}$$

# Main Results I

**Theorem: Relation to a Deterministic Objective**

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

$$\frac{1}{3p\theta}\mathbb{E}_{\boldsymbol{X}_o}\|\boldsymbol{A}\boldsymbol{Y}\|_4^4 = (1-\theta)\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4 + \theta n. \tag{19}$$

**Global Maxima of the Deterministic Objective**

$$\boldsymbol{W}_\star \in \operatorname*{arg\,max}_{\boldsymbol{W} \in \mathsf{O}(n;\mathbb{R})} \|\boldsymbol{W}\|_4^4 \iff \boldsymbol{W}_\star \in \mathsf{SP}(n) \tag{20}$$

# Main Results I

**Theorem: Relation to a Deterministic Objective**

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

$$\frac{1}{3p\theta}\mathbb{E}_{\boldsymbol{X}_o}\|\boldsymbol{A}\boldsymbol{Y}\|_4^4 = (1-\theta)\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4 + \theta n. \tag{19}$$

**Global Maxima of the Deterministic Objective**

$$\boldsymbol{W}_\star \in \underset{\boldsymbol{W}\in\mathsf{O}(n;\mathbb{R})}{\arg\max}\,\|\boldsymbol{W}\|_4^4 \iff \boldsymbol{W}_\star \in \mathsf{SP}(n) \tag{20}$$

**Global maxima of $\|AD_o\|_4^4$ are the correct dictionaries (up to signed permutation)!**

# Main Results II

**Theorem: Correctness of Global Optimal**

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

$$\max_{\boldsymbol{A}} \|\boldsymbol{A}\boldsymbol{Y}\|_4^4, \quad \text{s.t.} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}), \tag{21}$$

then for any $\varepsilon \in [0,1]$, there exists a signed permutation matrix $\boldsymbol{P} \in \mathsf{SP}(n)$, such that $\frac{1}{n}\left\|\hat{\boldsymbol{A}}_\star^* - \boldsymbol{D}_o\boldsymbol{P}\right\|_F^2 \le C\varepsilon$, 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 $\boldsymbol{X}_o \in \mathbb{R}^{n \times p}$, $x_{i,j} \sim_{iid} \mathsf{BG}(\theta)$, $\boldsymbol{D}_o \in \mathsf{O}(n;\mathbb{R})$ is any orthogonal matrix, and $\boldsymbol{Y} = \boldsymbol{D}_o \boldsymbol{X}_o$. Suppose $\hat{\boldsymbol{A}}_\star$ is a global maximizer of optimization:

$$\max_{\boldsymbol{A}} \|\boldsymbol{A}\boldsymbol{Y}\|_4^4, \quad \text{s.t.} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}), \tag{21}$$

then for any $\varepsilon \in [0,1]$, there exists a signed permutation matrix $\boldsymbol{P} \in \mathsf{SP}(n)$, such that $\frac{1}{n}\left\|\hat{\boldsymbol{A}}_\star^* - \boldsymbol{D}_o\boldsymbol{P}\right\|_F^2 \le C\varepsilon$, 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 $\|\boldsymbol{A}\boldsymbol{Y}\|_4^4$ are arbitrarily close to the correct dictionary!**

# Optimization Algorithm

The program:

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

seems to be the worst case for optimization:

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

Try projected (Riemannian) gradient descent anyway:

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

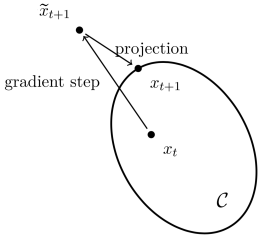
# Optimization Algorithm

Solve the program:

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

with projected (Riemannian) gradient descent:

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




**A happy accident:**
observed that this converges faster as $\alpha \to \infty$!

**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 $\|\boldsymbol{A}\boldsymbol{Y}\|_4^4$:

**Algorithm** MSP Algorithm on $\ell^4$ Dictionary Learning

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

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

# A Few Interpretations

**NOT Gradient Descent!**

"Fixed point" interpretation:

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

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

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

"Stochastic batch" variation:

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

# The MSP Algorithm II

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

**Algorithm** MSP Algorithm on $\ell^4$ over Orthogonal Group

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

# One Run of the MSP Algorithm

$$
\boldsymbol{A}_0 = \begin{pmatrix} -0.8249 & 0.3820 & -0.4168 \\ -0.5240 & -0.2398 & 0.8173 \\ -0.2122 & -0.8925 & -0.3979 \end{pmatrix} \xrightarrow{\textit{stretching}} \boldsymbol{A}_0^{\circ 3} = \begin{pmatrix} -0.5613 & 0.0557 & -0.0724 \\ -0.1439 & -0.0138 & 0.5459 \\ -0.0096 & -0.7109 & -0.0630 \end{pmatrix}
$$

$$
\xrightarrow{\textit{projection}} \boldsymbol{A}_1 = \begin{pmatrix} -0.9795 & 0.0621 & -0.1917 \\ -0.1953 & -0.0594 & 0.9789 \\ -0.0494 & -0.9963 & -0.0703 \end{pmatrix} \xrightarrow{\textit{stretching}} \boldsymbol{A}_1^{\circ 3} = \begin{pmatrix} -0.9397 & 0.0002 & -0.0070 \\ -0.0075 & -0.0002 & 0.9381 \\ -0.0001 & -0.9889 & -0.0003 \end{pmatrix}
$$

$$
\xrightarrow{\textit{projection}} \boldsymbol{A}_2 = \begin{pmatrix} -1.0000 & 0.0002 & -0.0077 \\ -0.0077 & -0.0003 & 1.000 \\ -0.0002 & -1.0000 & -0.0003 \end{pmatrix} \xrightarrow{\textit{stretching}} \boldsymbol{A}_2^{\circ 3} = \begin{pmatrix} -0.9999 & 0.0000 & -0.0000 \\ -0.0000 & -0.0000 & 0.9999 \\ -0.0000 & -1.0000 & -0.0000 \end{pmatrix}
$$

$$
\xrightarrow{\textit{projection}} \boldsymbol{A}_3 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \xrightarrow{\textit{output}} \boldsymbol{A}_3^{\circ 3} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}.
$$

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

# Convergence Guarantee of the MSP Algorithm

## Theorem (Local Convergence of the MSP Algorithm)

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

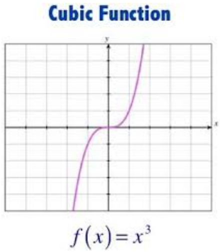



Figure: Cubic Function from $\ell^4$.

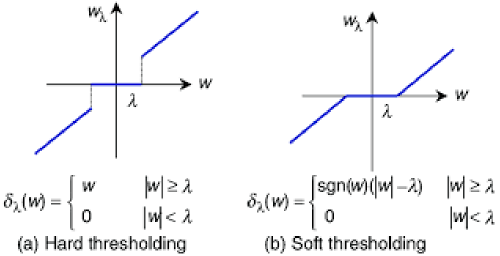



Figure: Thresholding from $\ell^1$.

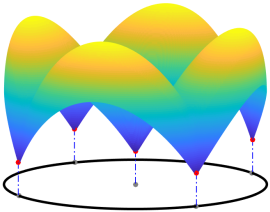
# Convergence Guarantee of the MSP Algorithm

## Theorem (Local Convergence of the MSP Algorithm)

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

## Generalization to all Signed Permutation Matrices

The Identity can be generalized to any signed permutation matrix!



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

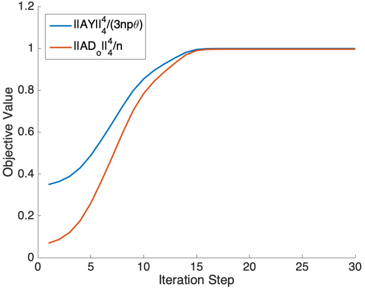
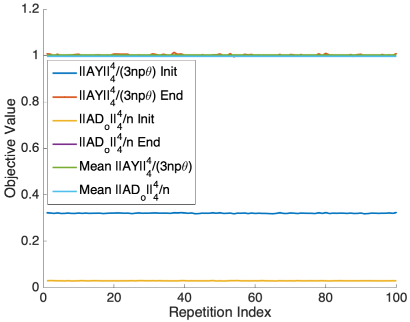



Figure: The value of $\frac{1}{3np\theta}\|\boldsymbol{AY}\|_4^4$ and $\frac{1}{n}\|\boldsymbol{AD}_o\|_4^4$ in two experiments with different settings: left: $n = 50, p = 20000, \theta = 0.3$, right: $n = 100, p = 40000, \theta = 0.3$. **The MSP algorithm converges quickly and smoothly with dozens of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

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



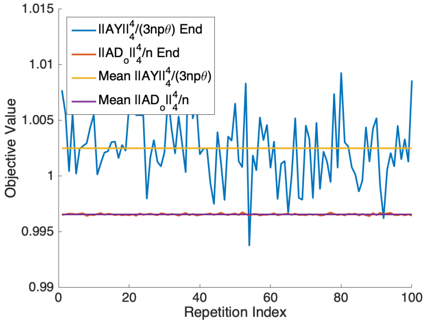




Figure: Initial value and final value of $\frac{1}{3np\theta}\|\boldsymbol{AY}\|_4^4$ and $\frac{1}{n}\|\boldsymbol{AD}_o\|_4^4$ for dictionary learning, with $n = 100, p = 40000$, $\theta = 0.3$, left: with initial values; right: without initial values. **All 100 trials converge to the global optima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

# Phase Transition of the MSP algorithm

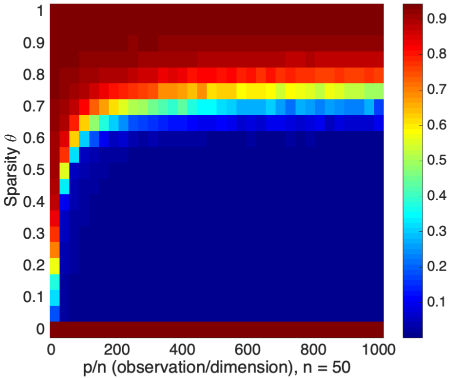



Figure: Phase transition plot of average normalized error $\left|1 - \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4 / n\right|$ for 10 trials of MSP algorithm 1 with $n = 50$. Red area indicates large error and blue area small error. Plot shows results for varying $p$ versus $\theta$. **The algorithm successes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

# Phase Transition of the MSP algorithm

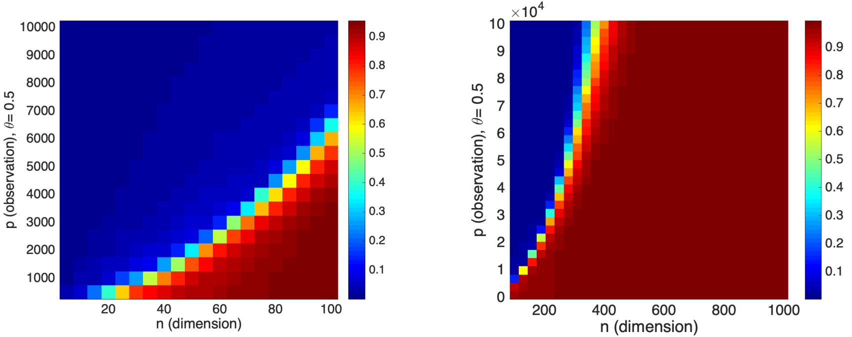
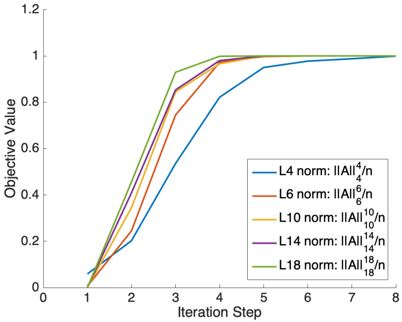



Figure: Phase transition plot of average normalized error $\left|1 - \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4/n\right|$ for 10 trials of MSP algorithm 1 with $\theta = 0.5$. Red area indicates large error and blue area small error, left: $n$ from 10 to 100 and $p$ from $10^3$ to $10^4$, right: changing $n$ 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 $\ell^{2k}$ Norm



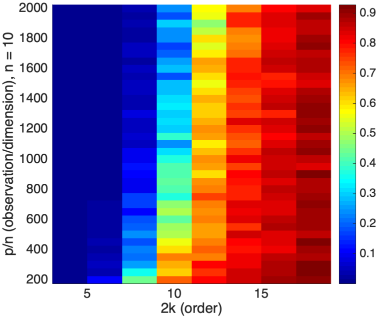




Figure: Experiments with different $\ell^{2k}$ norm. Left: Maximizing $\|\boldsymbol{A}\|_{2k}^{2k}$ for different order $k$. Right: Average normalized error of $\left|1 - \|\boldsymbol{A}\boldsymbol{D}_o\|_{2k}^{2k}/n\right|$ for maximizing $\|\boldsymbol{A}\boldsymbol{Y}\|_{2k}^{2k}$ for 20 trials, with $n = 10$, varying $k$ and $p$. $\ell^4$ **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 | $> 12$h | **0.35%** | **48.0s** |
| (e) | 13.07% | 5401.9s | N/A | $> 12$h | **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 $\left|1 - \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4 / n\right|$. 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]

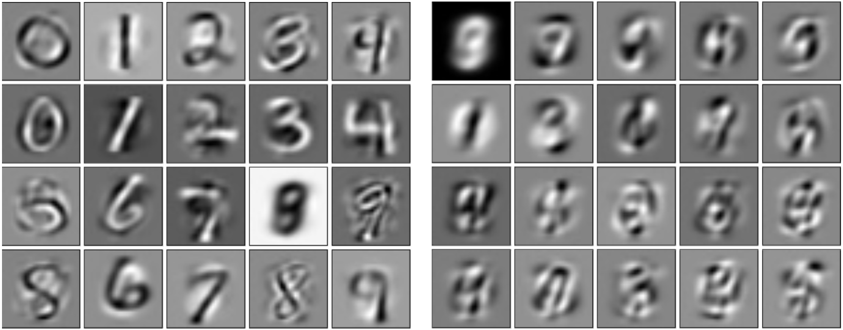
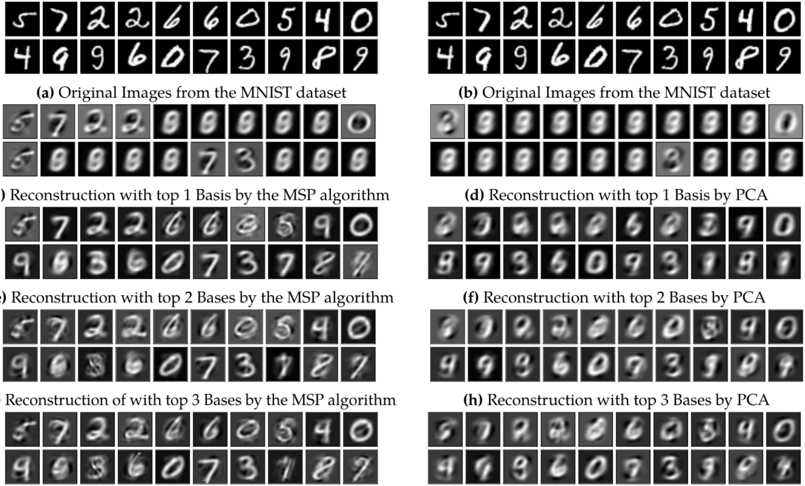


Figure: Bases learned from the MNIST dataset. Left: Some selected "meaningful" bases learned through MSP; Right: Top bases learned through PCA.

# 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 of 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]

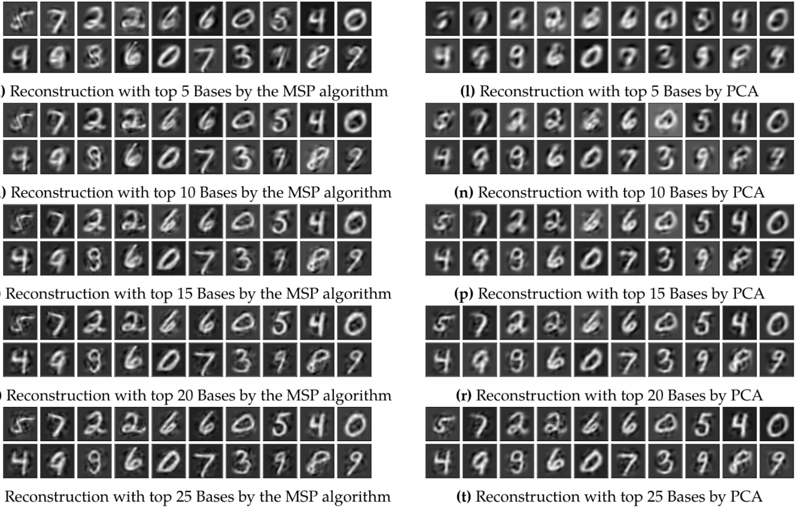



Figure: Reconstruction result comparison between MSP and PCA using different number of bases.

# Generalization to Stiefel Manifold [ZMZM20]

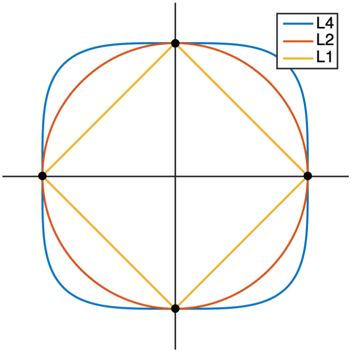



Figure: $\ell^1$-,$\ell^2$-, and $\ell^4$-spheres in $\mathbb{R}^2$

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

$$\max_{\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})} \frac{1}{4} \|\boldsymbol{A}\boldsymbol{Y}\|_4^4, \tag{22}$$

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

# Generalization to Stiefel Manifold [ZMZM20]

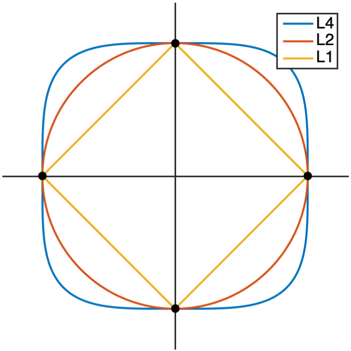



Figure: $\ell^1$-,$\ell^2$-, and $\ell^4$-spheres in $\mathbb{R}^2$

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

$$\max_{\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})} \frac{1}{4} \|\boldsymbol{A}\boldsymbol{Y}\|_4^4, \qquad (22)$$

where the orthogonality constraint $\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})$ 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 $k (1 \le k \le n)$ bases?**

# Generalization to Stiefel Manifold

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

$$\max_{\boldsymbol{W}} \frac{1}{4} \|\boldsymbol{W}^* \boldsymbol{Y}\|_4^4 \quad \text{s.t.} \quad \boldsymbol{W} \in \mathsf{St}(k, n; \mathbb{R}) \subset \mathbb{R}^{n \times k}. \tag{23}$$

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

$$\boldsymbol{W}_{t+1} = \mathcal{P}_{\mathsf{St}(k,n;\mathbb{R})} \left[ \nabla_{\boldsymbol{W}} \phi(\boldsymbol{W}_t) \right] = \boldsymbol{U}_t \boldsymbol{V}_t^*, \tag{24}$$

where $\boldsymbol{U}_t \boldsymbol{\Sigma}_t \boldsymbol{V}_t^* = \mathsf{SVD}[\boldsymbol{Y}(\boldsymbol{Y}^* \boldsymbol{W}_t)^{\circ 3}]$.

[5] For any $1 \le k \le n$, $\mathsf{St}(k, n; \mathbb{R}) \doteq \{\boldsymbol{W} \in \mathbb{R}^{n \times k} : \boldsymbol{W}^* \boldsymbol{W} = \boldsymbol{I}_k\}$.

# Relation with Geometric Interpretation of PCA

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

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

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

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

- $\ell^4$-Norm maximization

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

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

# Relation with Statistical Interpretation of PCA

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

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

# Relation with ICA and $4^{\text{th}}$ Order Moment

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

$$\mathrm{kurt}(\boldsymbol{w}^*\boldsymbol{y}) = \mathbb{E}[\boldsymbol{w}^*\boldsymbol{y}]^4 - 3\,\|\boldsymbol{w}\|_2^4 \tag{25}$$

can identify one independent component of $\boldsymbol{y}$.

# Relation with ICA and $4^{\text{th}}$ Order Moment

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

$$\operatorname{kurt}(\boldsymbol{w}^* \boldsymbol{y}) = \mathbb{E}[\boldsymbol{w}^* \boldsymbol{y}]^4 - 3 \|\boldsymbol{w}\|_2^4 \tag{25}$$

can identify one independent component of $\boldsymbol{y}$.

## Importance of $4^{\text{th}}$ Order Statistics

- The $4^{\text{th}}$ 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:
$$\max_{w \in \mathbb{S}^{n-1}} \varphi(w) \doteq \frac{1}{2} \|\boldsymbol{w}^* \boldsymbol{Y}\|_2^2$$
  - Algorithm:
$$\boldsymbol{w}_{t+1} = \mathcal{P}_{\mathbb{S}^{n-1}}[\nabla_{\boldsymbol{w}} \varphi(\boldsymbol{w}_t)] = \frac{\boldsymbol{Y}\boldsymbol{Y}^* \boldsymbol{w}_t}{\|\boldsymbol{Y}\boldsymbol{Y}^* \boldsymbol{w}_t\|_2}$$
- **ICA**
  - Optimization:
$$\max_{\boldsymbol{w} \in \mathbb{S}^{n-1}} \psi(\boldsymbol{w}) \doteq \frac{1}{4} \mathrm{kurt}[\boldsymbol{w}^* \boldsymbol{y}] = \frac{1}{4} \mathbb{E}\left[\boldsymbol{w}^* \boldsymbol{y}\right]^4 - \frac{3}{4} \|\boldsymbol{w}\|_2^4$$
  - Algorithm:
$$\boldsymbol{w}_{t+1} = \mathcal{P}_{\mathbb{S}^{n-1}}\left[\nabla_{\boldsymbol{w}} \psi(\boldsymbol{w}_t)\right] = \frac{\mathbb{E}\left[\boldsymbol{y}\left(\boldsymbol{y}^* \boldsymbol{w}_t\right)^3\right] - 3\|\boldsymbol{w}_t\|_2^2 \boldsymbol{w}_t}{\left\|\mathbb{E}\left[\boldsymbol{y}\left(\boldsymbol{y}^* \boldsymbol{w}_t\right)^3\right] - 3\|\boldsymbol{w}_t\|_2^2 \boldsymbol{w}_t\right\|_2}$$
- **DL**
  - Optimization:
$$\max_{\boldsymbol{W} \in \mathsf{St}(k,n;\mathbb{R})} \phi(\boldsymbol{W}) \doteq \frac{1}{4} \|\boldsymbol{W}^* \boldsymbol{Y}\|_4^4$$
  - Algorithm:
$$\boldsymbol{W}_{t+1} = \mathcal{P}_{\mathsf{St}(k,n;\mathbb{R})}\left[\nabla_{\boldsymbol{W}} \phi(\boldsymbol{W}_t)\right] = \boldsymbol{U}_t \boldsymbol{V}_t^*,$$

  where $\boldsymbol{U}_t \boldsymbol{\Sigma}_t \boldsymbol{V}_t^* = \mathsf{SVD}[\boldsymbol{Y}(\boldsymbol{Y}^* \boldsymbol{W})^{\circ 3}]$.

# Gradient-Based Fixed Point Algorithms

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

$$x_{t+1} = g(x_t) = x_t - \frac{f(x_t)}{f'(x_t)}. \qquad (26)$$

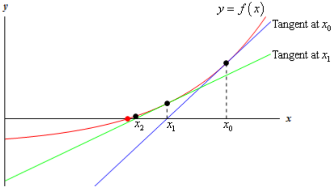
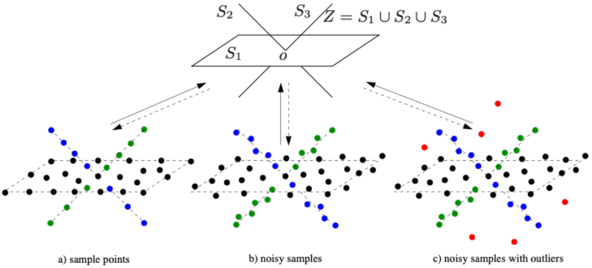


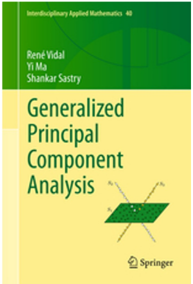

Table: PCA (Power iteration), ICA (FastICA), and DL (MSP) all are fixed-point algorithms based on **projected gradient descent**.

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

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

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

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

# Imperfect Measurements Type II: Outliers

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

[6] When $\tau p$ is not an integer, $\tau p$ is rounded to the closest integer.

# Imperfect Measurements Type II: Outliers

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

## Proposition (Objective with Outliers)

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

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

*where $C_\theta$ is a constant depending on $\theta$.*

[6] When $\tau p$ is not an integer, $\tau p$ is rounded to the closest integer.

# Imperfect Measurements Type III: Corruptions

**Measurements with Sparse Corruptions:** $\boldsymbol{Y}_C := \boldsymbol{Y} + \sigma \boldsymbol{B} \circ \boldsymbol{S}$, where $\sigma > 0$ controls the scale of corrupting entries, $\boldsymbol{B} \in \mathbb{R}^{n \times p}$ is a Bernoulli matrix with $b_{i,j} \sim_{iid} \mathrm{Ber}(\beta)$, where $\beta \in (0,1)$ controls the ratio of the sparse corruptions, and entries $s_{i,j}$ of $\boldsymbol{S} \in \mathbb{R}^{n \times p}$ 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:** $\boldsymbol{Y}_C := \boldsymbol{Y} + \sigma\boldsymbol{B} \circ \boldsymbol{S}$, where $\sigma > 0$ controls the scale of corrupting entries, $\boldsymbol{B} \in \mathbb{R}^{n\times p}$ 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 $\boldsymbol{S} \in \mathbb{R}^{n\times p}$ 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 $\boldsymbol{X}_o \in \mathbb{R}^{n\times p}$, $x_{i,j} \sim_{iid}$ BG$(\theta)$, $\boldsymbol{D}_o \in \mathsf{O}(n;\mathbb{R})$ is any orthogonal matrix and $\boldsymbol{Y} = \boldsymbol{D}_o\boldsymbol{X}_o$. For any orthogonal matrix $\boldsymbol{W} \in \mathsf{O}(n;\mathbb{R})$ and any random Bernoulli matrix $\boldsymbol{B} \in \mathbb{R}^{n\times p}, b_{i,j} \sim_{iid} Ber(\beta)$ independent of $\boldsymbol{X}_o$, let $\boldsymbol{Y}_C = \boldsymbol{Y} + \sigma\boldsymbol{B} \circ \boldsymbol{S}$ denote the data with sparse corruptions, and $\boldsymbol{S} \in \mathbb{R}^{n\times p}$ is defined as (68). Then the expectation of $\|\boldsymbol{W}^*\boldsymbol{Y}_C\|_4^4$ is:*

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

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

# Numerical Experiments I

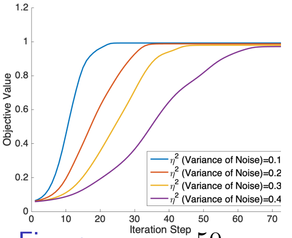



Figure: $n = 50, p = 20,000, \theta = 0.3$, varying $\eta^2$ from $0.1$ to $0.4$

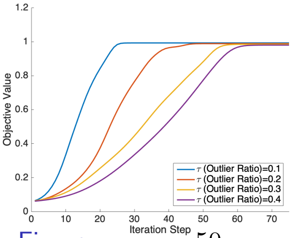



Figure: $n = 50, p = 20,000, \theta = 0.3$, varying $\tau$ from $0.1$ to $0.4$

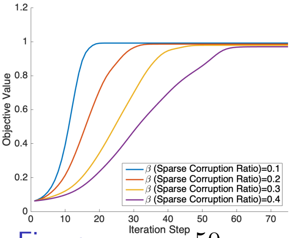



Figure: $n = 50, p = 20,000, \theta = 0.3$, $\sigma = 1$, varying $\beta$ from $0.1$ to $0.4$

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

# Numerical Experiments II

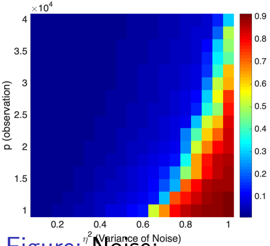



Figure: Noise:
$n = 50, \theta = 0.3$

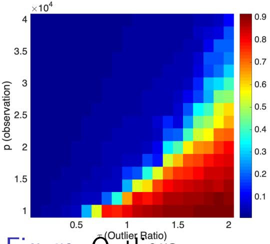



Figure: Outliers:
$n = 50, \theta = 0.3$

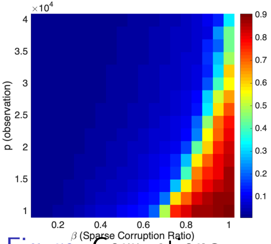
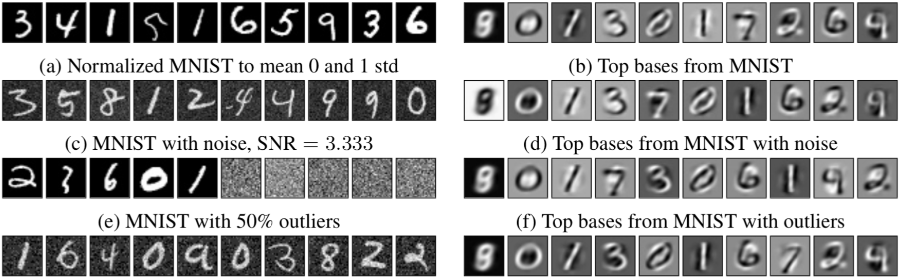



Figure: Corruptions:
$n = 50, \theta = 0.3$

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

# Real Image Data: MNIST



(a) Normalized MNIST to mean 0 and 1 std

(b) Top bases from MNIST

(c) MNIST with noise, SNR = 3.333

(d) Top bases from MNIST with noise

(e) MNIST with 50% outliers

(f) Top bases from MNIST with outliers

(g) MNIST with 50% sparse corruptions

(h) Top bases from MNIST with sparse corruptions

Figure: Top Bases learned from imperfect measurements of MNIST.

# Real Image Data: Single Image

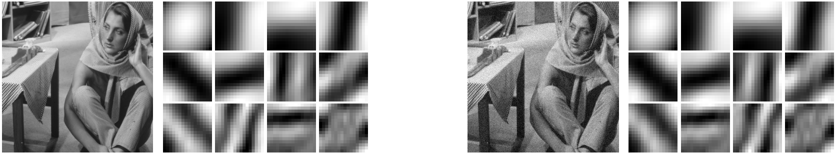


Figure: The top 12 bases learned from all $16 \times 16$ patches of Barbara, both with (right) and without (left) Gaussian noise. The noisy image is produced by adding Gaussian noise to the clean image, resulting in SNR of $5.87$.

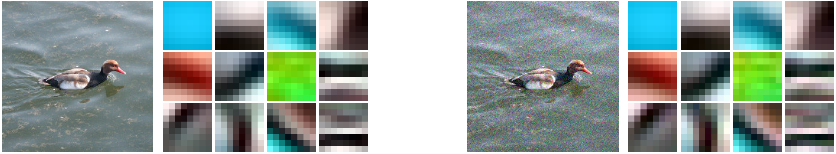


Figure: The top 12 bases learned from all $8 \times 8 \times 3$ color patches of the clean and noisy image, respectively. Here, the SNR of the noisy image is 6.56.

# Real Image Data: Single Image

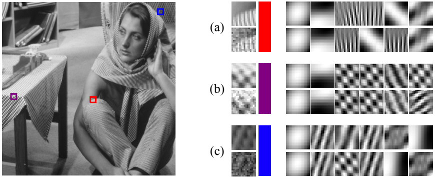



Figure: Representations of three $16 \times 16$ patches from Barbara w/ and w/o noise. Each selected patch is visualized, both w/ and w/o noise, and the top 6 corresponding bases are shown.

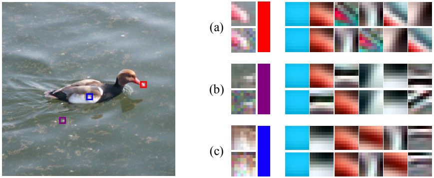
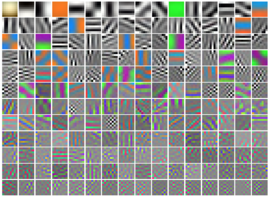
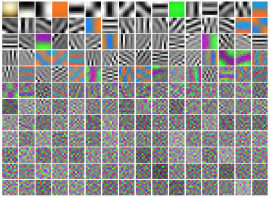



Figure: Representations of three $8 \times 8 \times 3$ patches from duck w/ and w/o noise. Each selected patch is visualized, both w/ and w/o noise, and the top 6 corresponding bases are shown.

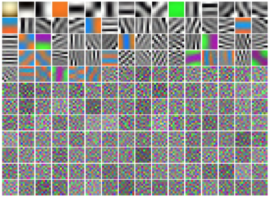
# Real Image Data: CIFAR-10



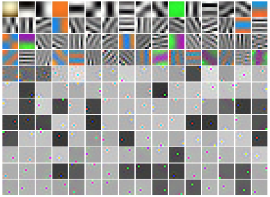
(a) Clean Patches



(b) Patches with Noise (SNR = 6.23)



(c) Patches with Outliers



(d) Patches with Sparse Corruptions

Figure: All $8 \times 8 \times 3 = 192$ bases learned from $100,000$ random $8 \times 8$ 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.

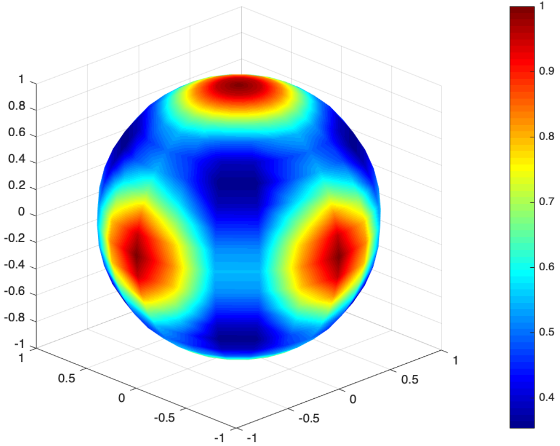
# Reference

**High-Dimensional Data Analysis with Low-Dimensional Models**

John Wright and Yi Ma
Cambridge University Press, 2022.

Pre-production copy at: https://book-wright-ma.github.io

- Github:
  https://book-wright-ma.github.io/Book-WM-20210422.pdf
- Dropbox:
  https://www.dropbox.com/s/0jm9qhm0t5v21gq/Book-WM-20210422.pdf?dl=0
- Chinese Version (available upon request).




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 L4-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, 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 L4 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 $\ell^4$-norm maximization over the orthogonal group.
*arXiv preprint arXiv:1906.02435*, 2019.