

# Matrix Analysis and Applications

## Chapter 10: Advanced Topics

**Instructor: Kai Lu**

(<http://seit.sysu.edu.cn/teacher/1801>)

School of Electronics and Information Technology  
Sun Yat-sen University

December 21, 2020

# Table of Contents

- 1 Sparse Recovery
- 2 Low-Rank Matrix Factorization
- 3 Tensor Decomposition

# Table of Contents

1 Sparse Recovery

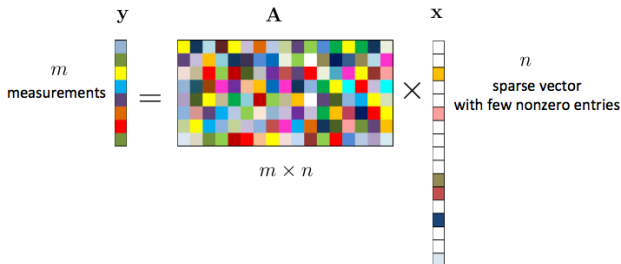
2 Low-Rank Matrix Factorization

3 Tensor Decomposition

# The Sparse Recovery Problem

**Problem:** given  $\mathbf{y} \in \mathbb{R}^m$  and  $\mathbf{A} \in \mathbb{R}^{m \times n}$  ( $m < n$ ), find a **sparsest**  $\mathbf{x} \in \mathbb{R}^n$  such that<sup>1</sup>

$$\mathbf{y} = \mathbf{A}\mathbf{x}. \quad (1)$$



- By sparsest, we mean that  $\mathbf{x}$  should have as many zero elements as possible.

<sup>1</sup>Michael Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, 2010.

# Sparse Recovery

Define

$$\|\mathbf{x}\|_0 \triangleq \sum_{i=1}^n \mathbf{1}\{x_i \neq 0\}, \quad (2)$$

which counts the number of nonzero entries of  $\mathbf{x}$ . Specifically,  $\mathbf{1}\{x \neq 0\}$  means that  $\mathbf{1}\{x \neq 0\} = 1$  if  $x \neq 0$ , and  $\mathbf{1}\{x \neq 0\} = 0$  if  $x = 0$ .

- $\|\mathbf{x}\|_0$  defined by (2) is commonly called the “ $\ell_0$ -norm” though strictly speaking it is not a norm.

# Sparse Recovery

Define

$$\|\mathbf{x}\|_0 \triangleq \sum_{i=1}^n \mathbf{1}\{x_i \neq 0\}, \quad (2)$$

which counts the number of nonzero entries of  $\mathbf{x}$ . Specifically,  $\mathbf{1}\{x \neq 0\}$  means that  $\mathbf{1}\{x \neq 0\} = 1$  if  $x \neq 0$ , and  $\mathbf{1}\{x \neq 0\} = 0$  if  $x = 0$ .

- $\|\mathbf{x}\|_0$  defined by (2) is commonly called the “ $\ell_0$ -norm” though strictly speaking it is not a norm.

**Minimum  $\ell_0$ -norm formulation:**

$$\mathcal{P}_0 : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad (3a)$$

$$\text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}. \quad (3b)$$

# Sparse Recovery

Define

$$\|\mathbf{x}\|_0 \triangleq \sum_{i=1}^n \mathbf{1}\{x_i \neq 0\}, \quad (2)$$

which counts the number of nonzero entries of  $\mathbf{x}$ . Specifically,  $\mathbf{1}\{x \neq 0\}$  means that  $\mathbf{1}\{x \neq 0\} = 1$  if  $x \neq 0$ , and  $\mathbf{1}\{x \neq 0\} = 0$  if  $x = 0$ .

- $\|\mathbf{x}\|_0$  defined by (2) is commonly called the “ $\ell_0$ -norm” though strictly speaking it is not a norm.

**Minimum  $\ell_0$ -norm formulation:**

$$\mathcal{P}_0 : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad (3a)$$

$$\text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}. \quad (3b)$$

## Question 1

Suppose that  $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}}$  (this is the model;  $\bar{\mathbf{x}}$  denotes the “true”  $\mathbf{x}$ ).

- (1) Does the minimum  $\ell_0$ -norm problem admit a solution that is exactly  $\bar{\mathbf{x}}$ ?
- (2) Also, is the solution unique?

# Sparse

**Spark:** the spark of  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , denoted by  $\text{spark}(\mathbf{A})$ , is the **smallest** number of **linearly dependent** columns of  $\mathbf{A}$ . Formally,

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{d} \neq \mathbf{0}} \|\mathbf{d}\|_0 \quad (4a)$$

$$\text{s.t. } \mathbf{A}\mathbf{d} = \mathbf{0}. \quad (4b)$$

In particular, if all the columns of  $\mathbf{A}$  are linearly independent,  $\text{spark}(\mathbf{A})$  is usually defined to be  $\infty$ .



# Sparse

**Spark:** the spark of  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , denoted by  $\text{spark}(\mathbf{A})$ , is the **smallest** number of **linearly dependent** columns of  $\mathbf{A}$ . Formally,

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{d} \neq \mathbf{0}} \|\mathbf{d}\|_0 \quad (4a)$$

$$\text{s.t. } \mathbf{A}\mathbf{d} = \mathbf{0}. \quad (4b)$$

In particular, if all the columns of  $\mathbf{A}$  are linearly independent,  $\text{spark}(\mathbf{A})$  is usually defined to be  $\infty$ .

- (1) Let  $\text{spark}(\mathbf{A}) = k$ . Then,  $k$  is the smallest number such that there exists a linearly dependent  $\{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_k}\}$  for some  $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$  (Assume that  $i_k \neq i_j$  for any  $k \neq j$ ).
- (2) Let  $\text{spark}(\mathbf{A}) = r + 1$ . Then, for any  $\{i_1, \dots, i_r\} \subseteq \{1, \dots, n\}$ ,  $\{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_r}\}$  is linearly independent. In other words, any collection of  $r$  columns of  $\mathbf{A}$  is linearly independent.

# Sparse

**Spark:** the spark of  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , denoted by  $\text{spark}(\mathbf{A})$ , is the **smallest** number of **linearly dependent** columns of  $\mathbf{A}$ . Formally,

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{d} \neq \mathbf{0}} \|\mathbf{d}\|_0 \quad (4a)$$

$$\text{s.t. } \mathbf{A}\mathbf{d} = \mathbf{0}. \quad (4b)$$

In particular, if all the columns of  $\mathbf{A}$  are linearly independent,  $\text{spark}(\mathbf{A})$  is usually defined to be  $\infty$ .

- (1) Let  $\text{spark}(\mathbf{A}) = k$ . Then,  $k$  is the smallest number such that there exists a linearly dependent  $\{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_k}\}$  for some  $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$  (Assume that  $i_k \neq i_j$  for any  $k \neq j$ ).
- (2) Let  $\text{spark}(\mathbf{A}) = r + 1$ . Then, for any  $\{i_1, \dots, i_r\} \subseteq \{1, \dots, n\}$ ,  $\{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_r}\}$  is linearly independent. In other words, any collection of  $r$  columns of  $\mathbf{A}$  is linearly independent.
- (3) **Comparison with rank:** If  $\text{rank}(\mathbf{A}) = j$ , then there exists a **linearly independent**  $\{\mathbf{a}_{i_1}, \dots, \mathbf{a}_{i_j}\}$  for some  $\{i_1, \dots, i_j\} \subseteq \{1, \dots, n\}$ .

# Sparse (cont'd)

Spark is stronger than rank. For example:

- (1) If any collection of  $m$  vectors in  $\{\mathbf{a}_1, \dots, \mathbf{a}_n\} \subseteq \mathbb{R}^m, n > m$ , is linearly independent, then

$$\text{spark}(\mathbf{A}) = m + 1, \quad \text{rank}(\mathbf{A}) = m.$$

- An example for this is Vandemonde matrices.

# Sparse (cont'd)

Spark is stronger than rank. For example:

- (1) If any collection of  $m$  vectors in  $\{\mathbf{a}_1, \dots, \mathbf{a}_n\} \subseteq \mathbb{R}^m, n > m$ , is linearly independent, then

$$\text{spark}(\mathbf{A}) = m + 1, \quad \text{rank}(\mathbf{A}) = m.$$

- An example for this is Vandemonde matrices.

- (2) Suppose  $\{\mathbf{v}_1, \dots, \mathbf{v}_r\} \in \mathbb{R}^n$  is linear independent, and let

$$\mathbf{A} = [\mathbf{v}_1, \dots, \mathbf{v}_r, \mathbf{v}_1].$$

Apparently,  $\text{rank}(\mathbf{A}) = r$ , yet  $\text{spark}(\mathbf{A}) = 2$ .

# Sparse (cont'd)

## Properties of spark:

Given  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with  $m \geq n$ , we have

- (1)  $\text{spark}(\mathbf{A}) \in \{1, 2, \dots, n\} \cup \{+\infty\}$ ,
- (2)  $\text{spark}(\mathbf{A}) = 1$  if and only if  $\mathbf{A}$  has a zero column,
- (3)  $\text{spark}(\mathbf{A}) = +\infty \iff \text{rank}(\mathbf{A}) = n$ ,
- (4) if  $\text{spark}(\mathbf{A}) \neq +\infty$ , then  $\text{spark}(\mathbf{A}) \leq \text{rank}(\mathbf{A}) + 1$ .

---

<sup>2</sup>D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via  $\ell_1$  minimization", *Proc. Natl. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.

# Sparse (cont'd)

## Properties of spark:

Given  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with  $m \geq n$ , we have

- (1)  $\text{spark}(\mathbf{A}) \in \{1, 2, \dots, n\} \cup \{+\infty\}$ ,
- (2)  $\text{spark}(\mathbf{A}) = 1$  if and only if  $\mathbf{A}$  has a zero column,
- (3)  $\text{spark}(\mathbf{A}) = +\infty \iff \text{rank}(\mathbf{A}) = n$ ,
- (4) if  $\text{spark}(\mathbf{A}) \neq +\infty$ , then  $\text{spark}(\mathbf{A}) \leq \text{rank}(\mathbf{A}) + 1$ .

## Remark 1

*The notion of spark of a matrix was introduced by Donoho and Elad.<sup>2</sup> It is strictly related to Compressed Sensing. The word “spark” comes from a verbal fusion of “sparse” and “rank”.*

---

<sup>2</sup>D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via  $\ell_1$  minimization”, *Proc. Natl. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.

# Perfect Recovery Guarantee of $\ell_0$ -norm Minimization

## Theorem 1 (Uniqueness – Spark)

*Suppose that  $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}}$ . Then,  $\bar{\mathbf{x}}$  is the unique solution (i.e., global optimal) to the  $\ell_0$ -norm problem if*

$$\|\bar{\mathbf{x}}\|_0 < \frac{1}{2} \text{spark}(\mathbf{A}). \quad (5)$$

# Perfect Recovery Guarantee of $\ell_0$ -norm Minimization

## Theorem 1 (Uniqueness – Spark)

*Suppose that  $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}}$ . Then,  $\bar{\mathbf{x}}$  is the unique solution (i.e., global optimal) to the  $\ell_0$ -norm problem if*

$$\|\bar{\mathbf{x}}\|_0 < \frac{1}{2} \text{spark}(\mathbf{A}). \quad (5)$$

- **Implication:** if the true  $\bar{\mathbf{x}}$  is sufficiently sparse, then the minimum  $\ell_0$ -norm problem perfectly recovers that  $\bar{\mathbf{x}}$ .



# Perfect Recovery Guarantee of $\ell_0$ -norm Minimization

## Theorem 1 (Uniqueness – Spark)

*Suppose that  $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}}$ . Then,  $\bar{\mathbf{x}}$  is the unique solution (i.e., global optimal) to the  $\ell_0$ -norm problem if*

$$\|\bar{\mathbf{x}}\|_0 < \frac{1}{2} \text{spark}(\mathbf{A}). \quad (5)$$

- **Implication:** if the true  $\bar{\mathbf{x}}$  is sufficiently sparse, then the minimum  $\ell_0$ -norm problem perfectly recovers that  $\bar{\mathbf{x}}$ .

## Proof sketch (by contradiction):

- (1) Let  $\mathbf{x}^*$  be a solution to the minimum  $\ell_0$ -norm problem. Let  $\mathbf{e} \triangleq \bar{\mathbf{x}} - \mathbf{x}^*$ .
- (2)  $\mathbf{0} = \mathbf{A}\bar{\mathbf{x}} - \mathbf{A}\mathbf{x}^* = \mathbf{A}\mathbf{e} \implies \|\mathbf{e}\|_0 \leq \|\bar{\mathbf{x}}\|_0 + \|\mathbf{x}^*\|_0 \leq 2\|\bar{\mathbf{x}}\|_0$ .
- (3) Suppose  $\mathbf{e} \neq \mathbf{0}$ . Then,  $\mathbf{A}\mathbf{e} = \mathbf{0}$  and  $\|\mathbf{e}\|_0 \leq 2\|\bar{\mathbf{x}}\|_0 \implies \text{spark}(\mathbf{A}) \leq 2\|\bar{\mathbf{x}}\|_0$ .

# Mutual Coherence and $\ell_0$ -norm Recovery Guarantee

**Mutual Coherence:** the mutual coherence of  $\mathbf{A}$  is defined as (a.k.a. dictionary coherence)

$$\mu(\mathbf{A}) \triangleq \max_{j \neq k} \frac{|\mathbf{a}_j^T \mathbf{a}_k|}{\|\mathbf{a}_j\|_2 \|\mathbf{a}_k\|_2}. \quad (6)$$

It measures how similar the columns of  $\mathbf{A}$  are in the worst-case sense.

---

<sup>3</sup>M. Xia, Y.-C. Wu, and S. Aïssa, "Non-orthogonal opportunistic beamforming: performance analysis and implementation", *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1424-1433, April 2012.

# Mutual Coherence and $\ell_0$ -norm Recovery Guarantee

**Mutual Coherence:** the mutual coherence of  $\mathbf{A}$  is defined as (a.k.a. dictionary coherence)

$$\mu(\mathbf{A}) \triangleq \max_{j \neq k} \frac{|\mathbf{a}_j^T \mathbf{a}_k|}{\|\mathbf{a}_j\|_2 \|\mathbf{a}_k\|_2}. \quad (6)$$

It measures how similar the columns of  $\mathbf{A}$  are in the worst-case sense.

In particular, for full-rank matrices of size  $m \times n$ , the mutual coherence is bounded from below by

$$\mu \geq \sqrt{\frac{n-m}{m(n-1)}}. \quad (7)$$

The equality in (7) is obtained for a family of matrices named Grassmannian Frames, which finds application in beamforming design of MIMO systems.<sup>3</sup>

---

<sup>3</sup>M. Xia, Y.-C. Wu, and S. Aïssa, "Non-orthogonal opportunistic beamforming: performance analysis and implementation", *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1424-1433, April 2012.

# Mutual Coherence and $\ell_0$ -norm Recovery Guarantee

## Theorem 2 (Uniqueness – Mutual Coherence)

*Suppose that  $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}}$ . Then,  $\bar{\mathbf{x}}$  is the unique solution (i.e., global optimal) to the minimum  $\ell_0$ -norm problem if*

$$\|\bar{\mathbf{x}}\|_0 < \frac{1}{2} \left( 1 + \frac{1}{\mu(\mathbf{A})} \right). \quad (8)$$

- **Implication:** perfect recovery may depend on how incoherent  $\mathbf{A}$  is.

**Proof idea:** it can be shown that  $\text{spark}(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{A})}$ .

## Proof of Theorem 2 (cf. R6, Lemma 2.1, p. 26).

First, modify the matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  by normalizing its columns to be of unit  $\ell_2$ -norm, obtaining  $\tilde{\mathbf{A}}$ . This operation preserves both the spark and the mutual-coherence. The entries of the resulting Gram matrix<sup>a</sup>  $\mathbf{G} = \tilde{\mathbf{A}}^T \tilde{\mathbf{A}}$  satisfy the following properties:

$$\{G_{k,k} = 1 : 1 \leq k \leq n\} \text{ and } \{|G_{k,j}| \leq \mu(\mathbf{A}) : 1 \leq k, j \leq n, k \neq j\}. \quad (9)$$

Consider an arbitrary leading minor from  $\mathbf{G}$  of size  $p \times p$ , built by choosing a subgroup of  $p$  columns from  $\tilde{\mathbf{A}}$  and computing their sub-Gram matrix. From the **Gershgorin disk theorem**<sup>b</sup>, if this minor is diagonally dominant, i.e., if  $\sum_{j \neq i} |G_{i,j}| < |G_{i,i}|$  for every  $i$ , then this sub-matrix of  $\mathbf{G}$  is positive-definite, and so those  $p$  columns from  $\tilde{\mathbf{A}}$  are linearly independent. The condition  $1 > (p-1)\mu(\mathbf{A}) \implies p < 1 + 1/\mu(\mathbf{A})$  implies positive-definiteness of every  $p \times p$  minor. Thus,  $p = 1 + 1/\mu(\mathbf{A})$  is the smallest possible number of columns that might lead to linear dependence, and thus  $\text{spark}(\mathbf{A}) \geq 1 + 1/\mu(\mathbf{A})$ . □

---

<sup>a</sup>In linear algebra, the Gram matrix of a set of vectors  $\mathbf{v}_1, \dots, \mathbf{v}_n$  in an inner product space is the Hermitian matrix of inner products, whose entries are given by  $G_{ij} = \langle \mathbf{v}_i, \mathbf{v}_j \rangle$ . An important application is to compute linear independence: a set of vectors is linearly independent if and only if the determinant of the Gram matrix is non-zero.

<sup>b</sup>For a general (possible complex) matrix  $\mathbf{H}$  of size  $n \times n$ , Gershgorin's disks are the  $n$  disks formed by the centers  $H_{i,i}$  and radius  $\sum_{j \neq i} |H_{i,j}|$ . The theorem states that all eigenvalues of  $\mathbf{H}$  must lie within the union of these disks.

## Remark 2 (Comparison of Theorems 1 and 2)

Comparing Theorems 1 and 2, it is clear that they are parallel in form, but with different assumptions. In general, Theorem 1, which uses *spark*, is sharp and is far more powerful than Theorem 2, which uses the *mutual coherence* and so only a lower bound on spark. By recalling (7), it is clear that the mutual coherence of  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can never be smaller than  $1/\sqrt{m}$ , and therefore, the cardinality bound of Theorem 2 is never larger than  $\sqrt{m}/2$ . However, by virtue of (??), the spark can easily be as large as  $m$ , and Theorem 1 then gives a bound as large as  $m/2$ . As a result, Theorem 2 is also widely known as a weak version of Theorem 1.

# Sparse Recovery

## Question 2

*How should we solve the minimum  $\ell_0$ -norm problem*

$$\mathcal{P}_0 : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad (10a)$$

$$\text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}, \quad (10b)$$

*or can it be efficiently solved?*

# Sparse Recovery

## Question 2

*How should we solve the minimum  $\ell_0$ -norm problem*

$$\mathcal{P}_0 : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad (10a)$$

$$\text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}, \quad (10b)$$

*or can it be efficiently solved?*

- (1)  $\ell_0$ -norm minimization does not lead to a simple solution as in  $\ell_2$ -norm minimization.
- (2) The minimum  $\ell_0$ -norm problem is **NP-hard** in general.



# Sparse Recovery

## Question 2

*How should we solve the minimum  $\ell_0$ -norm problem*

$$\mathcal{P}_0 : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad (10a)$$

$$\text{s.t. } \mathbf{y} = \mathbf{Ax}, \quad (10b)$$

*or can it be efficiently solved?*

- (1)  $\ell_0$ -norm minimization does not lead to a simple solution as in  $\ell_2$ -norm minimization.
- (2) The minimum  $\ell_0$ -norm problem is **NP-hard** in general.

## Remark 3 (NP-hard and time complexity)

*Given any  $\mathbf{y}$  and  $\mathbf{A}$ , the problem is unlikely to be exactly solvable in polynomial time, i.e., in a time complexity of  $\mathcal{O}(n^p)$  for any  $p > 0$ . For more about time complexity, please refer to Wikipedia.*

# Tackling the Minimum $\ell_0$ -Norm Problem

The minimum  $\ell_0$ -norm problem can be exactly solved by **exhaustive search**:

- 1: **for** all  $\mathcal{I} \subseteq \{1, 2, \dots, n\}$  **do**
- 2:   **if**  $\mathbf{y} = \mathbf{A}_{\mathcal{I}}\bar{\mathbf{x}}$  holds for some  $\bar{\mathbf{x}} \in \mathbb{R}^{|\mathcal{I}|}$  **then**
- 3:     record  $(\bar{\mathbf{x}}, \mathcal{I})$  as one of the candidate solutions
- 4:   **end if**
- 5:   output the candidate solution  $(\bar{\mathbf{x}}, \mathcal{I})$  that has the smallest  $|\mathcal{I}|$
- 6: **end for**

# Tackling the Minimum $\ell_0$ -Norm Problem

The minimum  $\ell_0$ -norm problem can be exactly solved by **exhaustive search**:

- 1: **for** all  $\mathcal{I} \subseteq \{1, 2, \dots, n\}$  **do**
- 2:     **if**  $\mathbf{y} = \mathbf{A}_{\mathcal{I}}\bar{\mathbf{x}}$  holds for some  $\bar{\mathbf{x}} \in \mathbb{R}^{|\mathcal{I}|}$  **then**
- 3:         record  $(\bar{\mathbf{x}}, \mathcal{I})$  as one of the candidate solutions
- 4:     **end if**
- 5:     output the candidate solution  $(\bar{\mathbf{x}}, \mathcal{I})$  that has the smallest  $|\mathcal{I}|$
- 6: **end for**

## Remark 4

- (1) *For example, for  $n = 3$ , we test 7 combinations:*

$$\mathcal{I} = \{1\}, \mathcal{I} = \{2\}, \mathcal{I} = \{3\}, \mathcal{I} = \{1, 2\}, \mathcal{I} = \{2, 3\}, \mathcal{I} = \{1, 3\}, \mathcal{I} = \{1, 2, 3\}.$$

- (2) *Preferable for small  $n$ , yet too expensive for large  $n$ .*
- (3) *To be specific, the computational complexity is  $\mathcal{O}(mn^k k^2)$  flops where  $k$  denotes the number of non-zero elements in  $\mathbf{x}$ .*
- (4) *How about trying less (approximation)?*

Consider an approximation called the **orthogonal matching pursuit (OMP)**.

- [illegible]

**Note:** The computational complexity of OMP algorithm is  $\mathcal{O}(kmn)$  flops.

# Perfect Recovery Guarantee of Greedy Pursuit

When is OMP equivalent to  $\ell_0$ -norm minimization?

- Many theoretically provable conditions, including variants of OMP, like the pure (PGA), the orthogonal (OGA), the relaxed (RGA), and the weak (WGA) greedy algorithm, and many other greedy approximation algorithms.
- Let's have a taste of it by considering a simple version.

# Perfect Recovery Guarantee of Greedy Pursuit

When is OMP equivalent to  $\ell_0$ -norm minimization?

- Many theoretically provable conditions, including variants of OMP, like the pure (PGA), the orthogonal (OGA), the relaxed (RGA), and the weak (WGA) greedy algorithm, and many other greedy approximation algorithms.
- Let's have a taste of it by considering a simple version.

## Theorem 3

Suppose that  $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}}$ . Then, OMP recovers  $\bar{\mathbf{x}}$  if

$$\|\bar{\mathbf{x}}\|_0 < \frac{1}{2} \left( 1 + \frac{1}{\mu(\mathbf{A})} \right). \quad (11)$$

- The sufficient recovery condition is the same as that of Theorem 2 (though not that of Theorem 1)

**Proof idea:** show that OMP is guaranteed to pick a correct column at every stage.

# Convex Relaxation

Another approximation approach is to replace  $\|\mathbf{x}\|_0$  by a convex function:

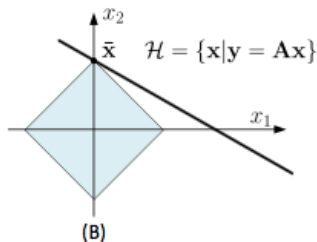
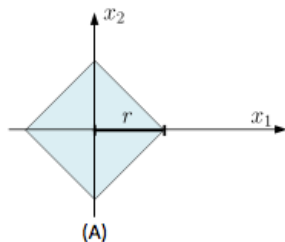
$$\mathcal{P}_1 : \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad (12a)$$

$$\text{s.t. } \mathbf{y} = \mathbf{A}\mathbf{x}. \quad (12b)$$

## Remark 5

- (1) *The above optimization problem is also known as **basis pursuit** in the literature;*
- (2) *It is convex and a linear programming, which can easily solved by using e.g., CVX or Matlab (with the built-in function LINPROG);*
- (3) *No closed-form solution as in the minimum  $\ell_2$ -norm problem, but can be efficiently solved via custom-designed optimization algorithms.*

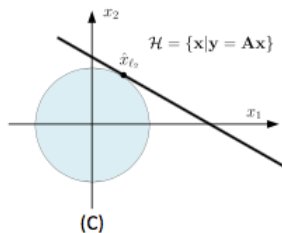
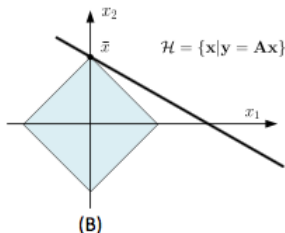
# Illustration of $\ell_1$ -Norm Geometry



- (1) Fig. A on the left shows the  $\ell_1$ -norm ball of radius  $r$  in  $\mathbb{R}^2$ . Note that the  $\ell_1$ -norm ball is “pointy” along the axes.
- (2) Fig. B on the right shows the  $\ell_1$ -norm recovery solution. The point  $\bar{x}$  is a “sparse” vector; the line  $\mathcal{H}$  is the set of all  $x$  that satisfy  $y = Ax$ .



# Illustration of $\ell_1$ -Norm Geometry (cont'd)



- (3) The  $\ell_1$ -norm recovery problem is to pick out a point in  $\mathcal{H}$  that has the minimum  $\ell_1$ -norm. We can see that  $\bar{x}$  in Fig. B on the left is such a point.
- (4) Fig. C on the right shows the geometry when  $\ell_2$ -norm is used. We can see that the solution  $\hat{x}$  may not be sparse.

# Recovery Guarantee of $\ell_1$ -Norm Minimization

When is  $\ell_1$ -norm minimization equivalent to  $\ell_0$ -norm minimization?

- Again, many theoretically provable conditions for  $\ell_1$ -norm minimization, such as the restricted isometry property condition, null space property, ...
- We sample a simple result here.

# Recovery Guarantee of $\ell_1$ -Norm Minimization

When is  $\ell_1$ -norm minimization equivalent to  $\ell_0$ -norm minimization?

- Again, many theoretically provable conditions for  $\ell_1$ -norm minimization, such as the restricted isometry property condition, null space property, ...
- We sample a simple result here.

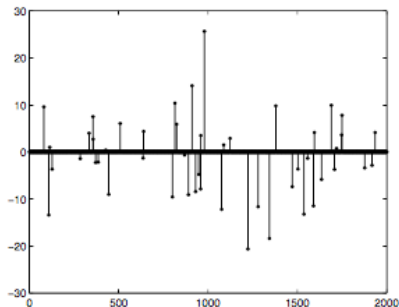
## Theorem 4

*Suppose that  $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}}$ . Then,  $\bar{\mathbf{x}}$  is the unique solution to the minimum  $\ell_1$ -norm problem if*

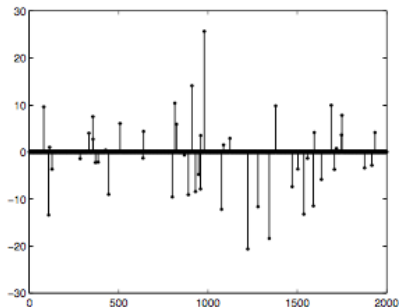
$$\|\bar{\mathbf{x}}\|_0 < \frac{1}{2} \left( 1 + \frac{1}{\mu(\mathbf{A})} \right). \quad (13)$$

# Toy Demonstration: Sparse Signal Reconstruction

- (1) Sparse vector  $\mathbf{x} \in \mathbb{R}^n$  with  $n = 2000$  and  $\|\mathbf{x}\|_0 = 50$ .
- (2)  $m = 400$  noise-free observations of  $\mathbf{y} = \mathbf{A}\mathbf{x}$ ,  $a_{ij}$  is randomly generated.

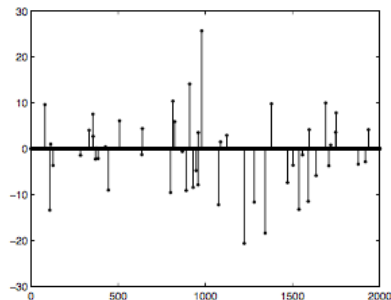


(a) Sparse source signal

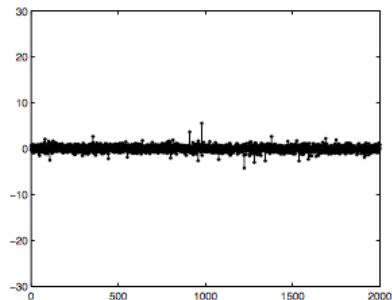


(b) Recovery by 1-norm minimization

## Toy Demonstration (cont'd)



(c) Sparse source signal



(d) Recovery by 2-norm minimization

# Application: Compressive Sensing (CS)

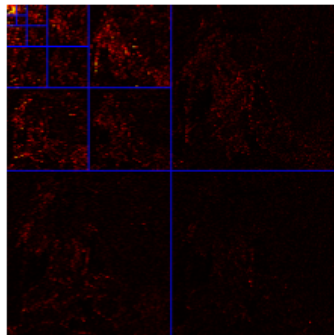
Consider a signal  $\tilde{x} \in \mathbb{R}^n$  that has a sparse representation  $x \in \mathbb{R}^n$  in the domain of  $\Psi \in \mathbb{R}^{n \times n}$  (e.g., FFT and wavelet), i.e.,

$$\tilde{x} = \Psi x, \quad (14)$$

where  $x$  is sparse.



(a) The pirate image  $\tilde{x}$



(b) The wavelet transform  $x$

# Application (cont'd)

- (1) To acquire  $x$ , we use a sensing matrix  $\Phi \in \mathbb{R}^{m \times n}$  to observe  $x$

$$y = \Phi \tilde{x} = \Phi \Psi x. \quad (15)$$

Here, we have  $m \ll n$ , i.e., very few observations compared to the dimension of  $x$ .

- (2) Such a  $y$  will be good for compression, transmission and storage.  
 (3)  $\tilde{x}$  is recovered by recovering  $x$  (by recalling  $\tilde{x} = \Psi x$ ):

$$\min_x \|x\|_0 \quad (16a)$$

$$\text{s.t. } y = Ax, \quad (16b)$$

where  $A \triangleq \Phi \Psi$ .

---

<sup>4</sup>Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, May 2012.

# Application (cont'd)

- (1) To acquire  $x$ , we use a sensing matrix  $\Phi \in \mathbb{R}^{m \times n}$  to observe  $x$

$$y = \Phi \tilde{x} = \Phi \Psi x. \quad (15)$$

Here, we have  $m \ll n$ , i.e., very few observations compared to the dimension of  $x$ .

- (2) Such a  $y$  will be good for compression, transmission and storage.

- (3)  $\tilde{x}$  is recovered by recovering  $x$  (by recalling  $\tilde{x} = \Psi x$ ):

$$\min_x \|x\|_0 \quad (16a)$$

$$\text{s.t. } y = Ax, \quad (16b)$$

where  $A \triangleq \Phi \Psi$ .

## Remark 6 (Compressive Sensing)

For more details on Compressive Sensing, please refer to the book<sup>4</sup> edited by Prof. Yonina Eldar.

<sup>4</sup>Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, May 2012.



# Variations

When  $\mathbf{y}$  is contaminated by noise, or when  $\mathbf{y} = \mathbf{A}\mathbf{x}$  does not exactly hold, some variations of the minimum  $\ell_1$ -norm problem can be considered.

- (1) **Basis pursuit denoising**: given  $\epsilon > 0$ , solve

$$\mathcal{P}_3: \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad (17a)$$

$$\text{s.t. } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon. \quad (17b)$$

- (2)  **$\ell_1$ -norm regularized LS**: given  $\lambda > 0$ , solve

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1. \quad (18)$$

- (3) **LASSO** (least absolute shrinkage and selection operator)<sup>5</sup>: given  $\tau > 0$ , solve

$$\mathcal{P}_4: \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \quad (19a)$$

$$\text{s.t. } \|\mathbf{x}\|_1 \leq \tau. \quad (19b)$$

---

<sup>5</sup>Robert Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267-288, 1996.

# Variations (cont'd)

- (4) When outliers exist in  $\mathbf{y}$ , i.e., some elements of  $\mathbf{y}$  are badly corrupted, we also want  $\mathbf{r} = \mathbf{y} - \mathbf{A}\mathbf{x}$  to be sparse. A outliers-robust formulation:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_1 + \lambda \|\mathbf{x}\|_1. \quad (20)$$

- (5) It may also happen that  $\mathbf{D}\mathbf{x}$  is sparse for some transformation  $\mathbf{D}$ . So,

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{D}\mathbf{x}\|_1. \quad (21)$$

- (6) **Take-home point:** if you want something sparse, try to minimize its  $\ell_1$ -norm:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_1 + \lambda \|\mathbf{D}\mathbf{x}\|_1. \quad (22)$$

# Application: Total Variation-based Denoising

## (1) Scenario:

- estimate  $\mathbf{x} \in \mathbb{R}^n$  from a noisy measurement  $\mathbf{x}_{\text{cor}} = \mathbf{x} + \boldsymbol{\nu}$ .
- $\mathbf{x}$  is known to be piecewise linear, i.e., for most  $i$  we have

$$x_i - x_{i-1} = x_{i+1} - x_i \iff -x_{i-1} + 2x_i - x_{i+1} = 0.$$

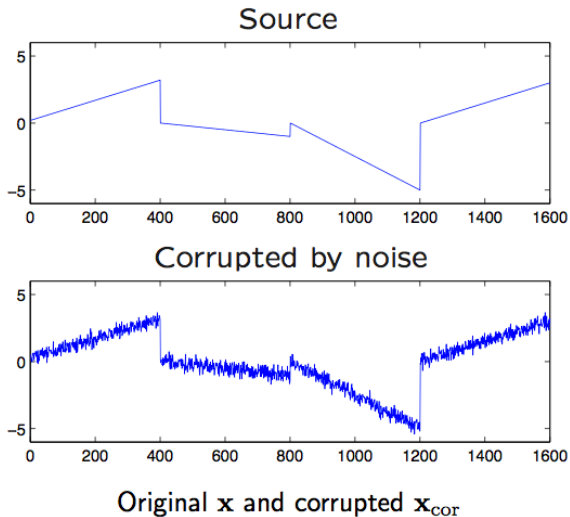
- equivalently,  $D\mathbf{x}$  is sparse, where

$$D = \begin{bmatrix} -1 & 2 & 1 & 0 & \cdots \\ 0 & -1 & 2 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & -1 & 2 & 1 \end{bmatrix}.$$

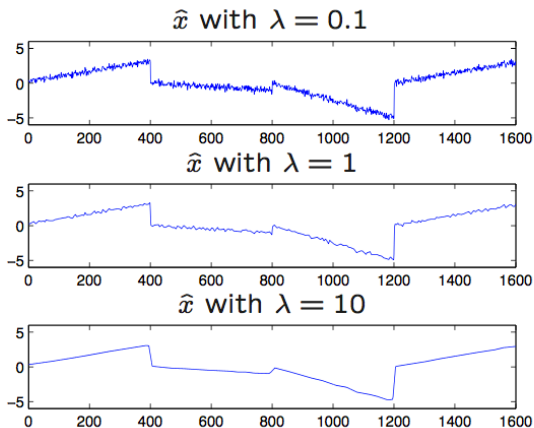
(2) Problem formulation:  $\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}_{\text{cor}} - \mathbf{x}\|_2 + \lambda \|D\mathbf{x}\|_0$ .

(3) Heuristic: change  $\|D\mathbf{x}\|_0$  to  $\|D\mathbf{x}\|_1$ .

## Application (cont'd)

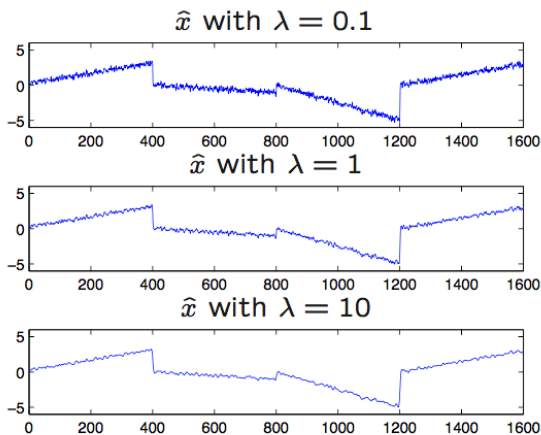


## Application (cont'd)



Denoised signals with different  $\lambda$ 's and by  $\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}_{\text{cor}} - \mathbf{x}\|_2 + \lambda \|\mathbf{D}\mathbf{x}\|_1$ .

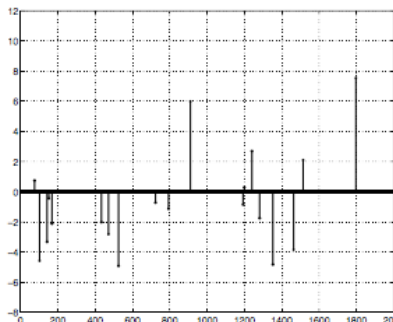
## Application (cont'd)



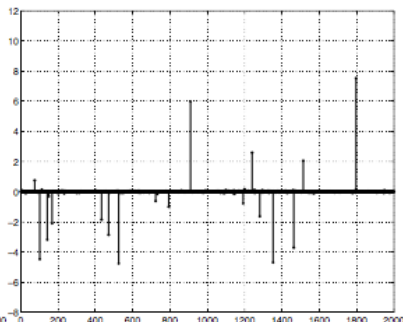
Denoised signals with different  $\lambda$ 's and by  $\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}_{\text{cor}} - \mathbf{x}\|_2 + \lambda \|\mathbf{D}\mathbf{x}\|_2$ .

# Toy Demonstration: Noisy sparse signal reconstruction

- (1) Sparse signal  $\mathbf{x} \in \mathbb{R}^n$  with  $n = 2000$  and  $\|\mathbf{x}\|_0 = 20$ .
- (2)  $m = 400$  noisy observations of  $\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\nu}$ , both  $a_{ij}$  and  $\nu_i$  are randomly generated.
- (3) 1-norm regularized LS  $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1$  is used, with  $\lambda = 0.1$ .

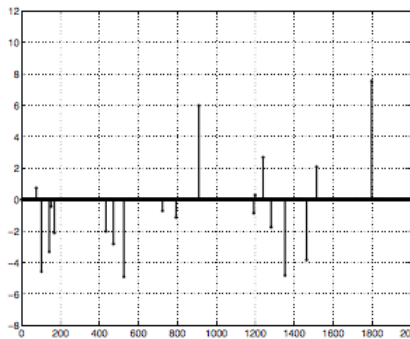


(a) Sparse source signal

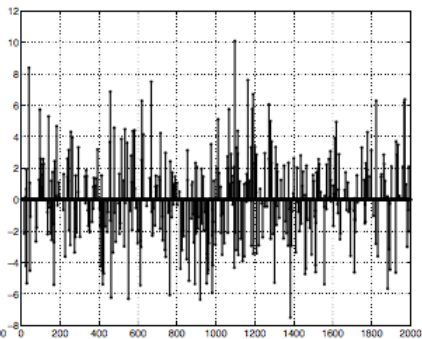


(b) 1-norm regularized LS estimate

## Toy Demonstration (cont'd)



(c) Sparse source signal



(d) LS estimate

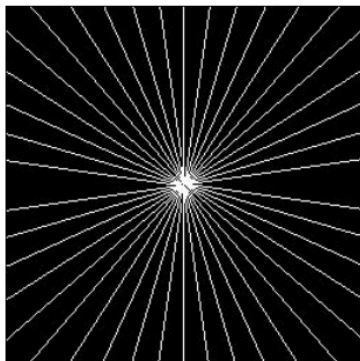


# Application: Magnetic Resonance Imaging (MRI)

Problem: MRI image reconstruction.



(a)

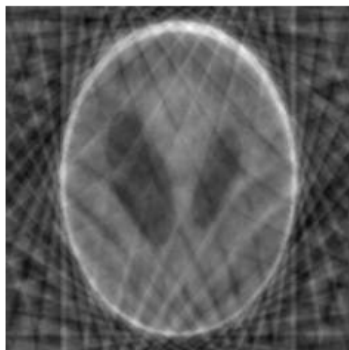


(b)

Fig. (a) shows the original test image. Fig. (b) shows the sampling region in the frequency domain. Fourier coefficients are sampled along 22 approximately radial lines.

Source: [R4]

## Application (cont'd)



(c)



(d)

Fig. (c) is the recovery by filling the unobserved Fourier coefficients to zero. Fig. (d) is the recovery by minimizing the total variations. Source: [R4]

# Toy Demonstration: Data Fitting

- (1) Recap: a noisy polynomial model:

$$f(x) = a_1 + a_2x + a_3x^2 + \cdots + a_nx^{n-1}, \quad (23)$$

$$y_i = f(x_i) + v_i, \quad i = 1, 2, \cdots, m, \quad (24)$$

where  $a_1, \cdots, a_n$  are the polynomial coefficients and are unknown;  $v_i$  is noise.

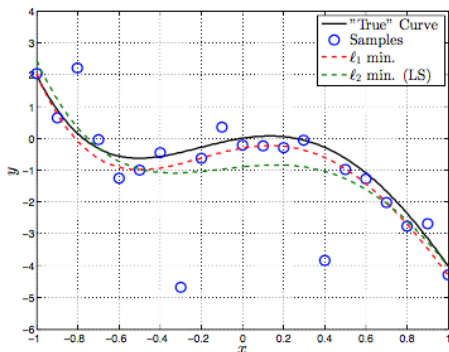
- (2) Problem: determine  $f(x)$  by estimating  $a = [a_1, \cdots, a_n]^T$  from the data set.
- (3) Generally, we want  $\mathbf{X}\mathbf{a}$  to be close to  $\mathbf{y}$ , where

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^{n-1} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

- (4) We also consider that outliers may exist in  $\mathbf{y}$  and that the guessed model order is **overestimated**.

# Toy Demonstration (cont'd)

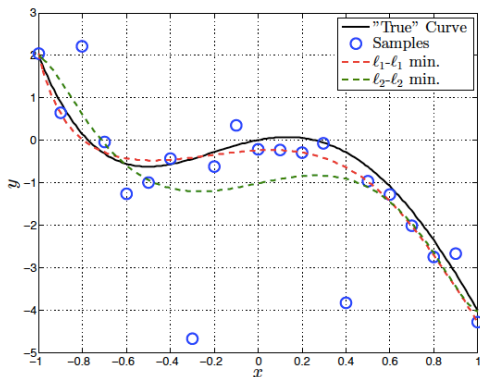
As a few data points deviate from the nominal data points, we use the  $\ell_1$ -norm minimization formulation  $\min_{\mathbf{a}} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|_1$ .



- "True" curve – the true  $f(x)$ ; the true model order is  $n = 4$ .
- $\ell_1$ -min and  $\ell_2$ -min – the estimated  $f(x)$ , with exact model order  $n = 4$ .

# Toy Demonstration (cont'd)

When the guessed model order is overestimated, we also want the coefficient vector  $\mathbf{a}$  to be sparse. We tried the  $\ell_1 - \ell_1$  formulation  $\min_{\mathbf{a}} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|_1 + \lambda \|\mathbf{a}\|_1$ .



- "True" curve – the true  $f(x)$ ; the true model order is  $n = 4$ .
- $\ell_1 - \ell_1$ -min and  $\ell_2 - \ell_2$ -min – the estimated  $f(x)$ , with  $\lambda = 0.1$  and a guessed model order  $n = 18$ .

# An Algorithm for $\ell_2 - \ell_1$ Minimization

Consider the  $\ell_2 - \ell_1$  minimization problem

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 + \lambda \|\mathbf{x}\|_1. \quad (25)$$

- As mentioned, the problem is convex and there are many optimization algorithms designed for it.
- Aim: get a flavor of one particular algorithm – that is sufficiently “matrix”.

# An Algorithm for $\ell_2 - \ell_1$ Minimization (cont'd)

To get some insights, consider the (plain old) LS problem

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2. \quad (26)$$

Now, suppose we don't want to solve the normal equation which requires  $\mathcal{O}(n^3)$ . Here's a trick: observe that for a given  $\bar{\mathbf{x}}$ ,

$$\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \underbrace{\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + (c\|\mathbf{x} - \bar{\mathbf{x}}\|_2^2 - \|\mathbf{A}\mathbf{x} - \mathbf{A}\bar{\mathbf{x}}\|_2^2)}_{\triangleq g(\mathbf{x}, \bar{\mathbf{x}})}, \quad (27)$$

where  $c > \lambda_{\max}(\mathbf{A}^T \mathbf{A})$ . It can be verified that

$$g(\mathbf{x}, \bar{\mathbf{x}}) = c \left\| \mathbf{x} - \left( \frac{1}{c} \mathbf{A}^T (\mathbf{y} - \mathbf{A}\bar{\mathbf{x}}) + \bar{\mathbf{x}} \right) \right\|_2^2 + \text{const.} \quad (28)$$

Also,

$$\arg \min_{\mathbf{x}} g(\mathbf{x}, \bar{\mathbf{x}}) = \frac{1}{c} \mathbf{A}^T (\mathbf{y} - \mathbf{A}\bar{\mathbf{x}}) + \bar{\mathbf{x}}. \quad (29)$$

# An Algorithm for $\ell_2 - \ell_1$ Minimization (cont'd)

An iterative algorithm for LS: given an initial point  $\mathbf{x}^{(0)}$ , do

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} g(\mathbf{x}, \mathbf{x}^{(k)}) \\ &= \frac{1}{c} \mathbf{A}^T (\mathbf{y} - \mathbf{A} \mathbf{x}^{(k)}) + \mathbf{x}^{(k)}, \quad k = 1, 2, \dots\end{aligned}\tag{30}$$



# An Algorithm for $\ell_2 - \ell_1$ Minimization (cont'd)

An iterative algorithm for LS: given an initial point  $\mathbf{x}^{(0)}$ , do

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} g(\mathbf{x}, \mathbf{x}^{(k)}) \\ &= \frac{1}{c} \mathbf{A}^T (\mathbf{y} - \mathbf{A} \mathbf{x}^{(k)}) + \mathbf{x}^{(k)}, \quad k = 1, 2, \dots\end{aligned}\quad (30)$$

- Pros: simple per-iteration update, may run under large-scale problem scenarios.

# An Algorithm for $\ell_2 - \ell_1$ Minimization (cont'd)

An iterative algorithm for LS: given an initial point  $\mathbf{x}^{(0)}$ , do

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} g(\mathbf{x}, \mathbf{x}^{(k)}) \\ &= \frac{1}{c} \mathbf{A}^T (\mathbf{y} - \mathbf{A} \mathbf{x}^{(k)}) + \mathbf{x}^{(k)}, \quad k = 1, 2, \dots\end{aligned}\quad (30)$$

- Pros: simple per-iteration update, may run under large-scale problem scenarios.
- **Question:** does it converge to the desired LS solution?

# An Algorithm for $\ell_2 - \ell_1$ Minimization (cont'd)

The above example is an instance of **majorization minimization (MM)**.

- (1) Consider a general optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}).$$

Suppose the problem is hard to directly minimize.

- (2) Let's introduce a **surrogate function** of  $f(\mathbf{x})$  that is easy to solve and satisfies

$$\mathbf{g}(\mathbf{x}, \bar{\mathbf{x}}) \geq f(\mathbf{x}) \text{ and } \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{x}}) = f(\bar{\mathbf{x}}), \quad (31)$$

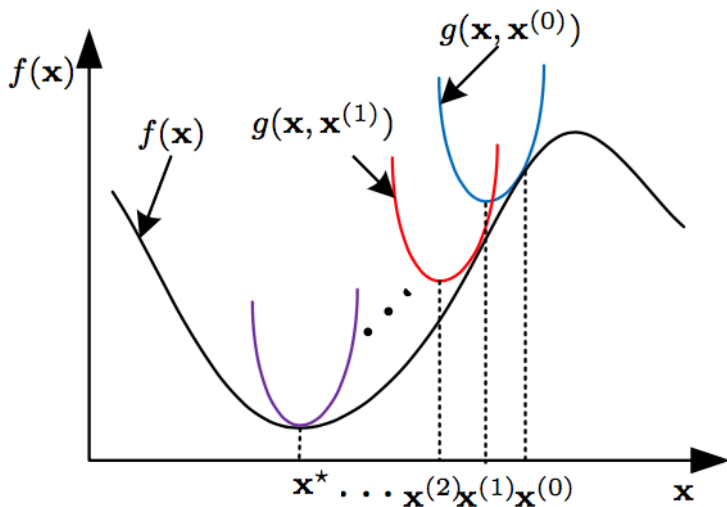
for all  $\mathbf{x}$  and  $\bar{\mathbf{x}}$ .

- (3) MM update:

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x}} \mathbf{g}(\mathbf{x}, \mathbf{x}^{(k)}), \quad k = 1, 2, \dots \quad (32)$$

- (4) Under some conditions,  $\mathbf{x}^{(k)}$  is guaranteed to converge to an optimal solution.

# An Algorithm for $\ell_2 - \ell_1$ Minimization (cont'd)



# An Algorithm for $\ell_2 - \ell_1$ Minimization (Finally)

Consider the  $\ell_2 - \ell_1$  minimization objective

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1, \quad (33)$$

and define the surrogate function

$$\mathbf{g}(\mathbf{x}, \mathbf{x}^{(k)}) \triangleq f(\mathbf{x}) + \left( \frac{c}{2} \left\| \mathbf{x} - \mathbf{x}^{(k)} \right\|_2^2 - \frac{1}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}^{(k)} \right\|_2^2 \right) \quad (34)$$

$$= \frac{c}{2} \left\| \mathbf{x} - \hat{\mathbf{x}}^{(k)} \right\|_2^2 + \lambda \|\mathbf{x}\|_1 + \text{const.}, \quad (35)$$

where  $c > \lambda_{\max}(\mathbf{A}^T \mathbf{A})$  and  $\hat{\mathbf{x}}^{(k)} = \frac{1}{c} \mathbf{A}^T (\mathbf{y} - \mathbf{A}\mathbf{x}^{(k)}) + \mathbf{x}^{(k)}$ .

It is shown that the minimizer  $\mathbf{x}^{(k+1)}$  of  $\mathbf{g}(\mathbf{x}, \mathbf{x}^{(k)})$  is simply

$$x_i^{(k+1)} = \text{soft}(\hat{x}_i^{(k)}, \lambda/c), \quad i = 1, \dots, n \quad (36)$$

where  $\text{soft}(z, \sigma) \triangleq \text{sign} z \times \max\{|z| - \sigma, 0\}$  denotes a **soft-thresholding** operation.

# Table of Contents

1 Sparse Recovery

2 Low-Rank Matrix Factorization

3 Tensor Decomposition

# Preliminaries – Alternating Optimization

- (1) Consider an optimization problem

$$\min_{\mathbf{a}, \mathbf{b}, \mathbf{c}} f(\mathbf{a}, \mathbf{b}, \mathbf{c}). \quad (37)$$

- (2) The problem may be difficult to solve if we try to optimize all the optimization variables  $\mathbf{a}$ ,  $\mathbf{b}$  and  $\mathbf{c}$  **simultaneously**.
- (3) But it may be easier if we just optimize one optimization variable while fixing the other two. This leads to the following **alternating** optimization algorithm.

---

```

1 initialize  $\mathbf{a} = \mathbf{a}^{(0)}$ ,  $\mathbf{b} = \mathbf{b}^{(0)}$ ,  $\mathbf{c} = \mathbf{c}^{(0)}$ , and  $k = 0$ ;
2 repeat
3    $\mathbf{a}^{(k+1)} = \arg \min_{\mathbf{a}} f(\mathbf{a}, \mathbf{b}^{(k)}, \mathbf{c}^{(k)});$ 
4    $\mathbf{b}^{(k+1)} = \arg \min_{\mathbf{b}} f(\mathbf{a}^{(k+1)}, \mathbf{b}, \mathbf{c}^{(k)});$ 
5    $\mathbf{c}^{(k+1)} = \arg \min_{\mathbf{c}} f(\mathbf{a}^{(k+1)}, \mathbf{b}^{(k+1)}, \mathbf{c});$ 
6    $k = k + 1$ ;
7 until some stopping conditions are satisfied;
```

---

# Preliminaries – Matrix Factorization

**Problem:** factor  $\mathbf{Y} \in \mathbb{R}^{m \times n}$  as  $\mathbf{Y} = \mathbf{AB}$  (either exactly or approximately), where  $\mathbf{A} \in \mathbb{R}^{m \times r}$ ,  $\mathbf{B} \in \mathbb{R}^{r \times n}$ , and  $r \ll \min\{m, n\}$ .

A formulation:

$$\min_{\mathbf{A}, \mathbf{B}} \|\mathbf{Y} - \mathbf{AB}\|_F^2. \quad (38)$$

$$\mathbf{Y} \approx \mathbf{A} \times \mathbf{B}$$

- Applications: dimensionality reduction, extracting meaningful feature from data, low-rank modeling, and so forth.<sup>6</sup>

<sup>6</sup>Ivan Markovsky, *Low-Rank Approximation: Algorithms, Implementation, Applications*, 2nd ed., Springer, 2018.



# Preliminaries – Matrix Factorization (cont'd)

- (1) The problem given by (38) is same as the low-rank matrix approximation problem; SVD is the solution.
- (2) Despite that, let's take a look at alternating optimization: for  $k = 1, 2, \dots$

$$\mathbf{A}^{(k)} = \arg \min_{\mathbf{A}} \left\| \mathbf{Y} - \mathbf{A} \mathbf{B}^{(k-1)} \right\|_F = \mathbf{Y} \left( \mathbf{B}^{(k-1)} \right)^\dagger, \quad (39)$$

$$\mathbf{B}^{(k)} = \arg \min_{\mathbf{B}} \left\| \mathbf{Y} - \mathbf{A}^{(k)} \mathbf{B} \right\|_F = \left( \mathbf{A}^{(k)} \right)^\dagger \mathbf{Y}. \quad (40)$$

- Easy, just do LS iteratively.
  - Convergence even to a locally optimal solution is a tricky theoretical subject (heavy and skipped here), though practically works.
- (3) The **alternating optimization** concept is important as we proceed to harder low-rank problems.

# Low-Rank Matrix Completion

**Application:** recommendation engines

- In 2009, Netflix awarded \$1 million to a team that performed best in recommending new movies to users based on their previous preference [R2].

Let  $\mathbf{Z}$  be a preference matrix, where  $z_{ij}$  records how user  $i$  likes movie  $j$ .

$$\mathbf{Z} = \begin{matrix} & \text{movies } (j) \\ \begin{matrix} \text{users } (i) \\ \left[ \begin{array}{cccccc} 2 & 3 & 1 & ? & ? & 5 & 5 \\ 1 & ? & 4 & 2 & ? & ? & ? \\ ? & 3 & 1 & ? & 2 & 2 & 2 \\ ? & ? & ? & 3 & ? & 1 & 5 \end{array} \right] \end{matrix} \end{matrix}$$

- (1) Some entries  $z_{ij}$  are missing, since no one watches all movies.
- (2) Aim: guess the unknown  $z_{ij}$  from the known ones.
- (3)  $\mathbf{Z}$  is assumed to be of low rank, as researches show that only a few factors affect users' preferences.

# Low-Rank Matrix Completion (cont'd)

**A low-rank matrix completion formulation:**

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{Y}} \|\mathbf{Y} - \mathbf{AB}\|_F^2 \quad (41a)$$

$$\text{s.t. } y_{ij} = z_{ij}, \quad \forall (i, j) \in \Omega, \quad (41b)$$

where  $\mathbf{A} \in \mathbb{R}^{m \times r}$ ,  $\mathbf{B} \in \mathbb{R}^{r \times n}$ ,  $\mathbf{Y} \in \mathbb{R}^{m \times m}$ , and  $\Omega$  is an index set indicating the *available* elements of  $\mathbf{Z}$ .

# Low-Rank Matrix Completion (cont'd)

**A low-rank matrix completion formulation:**

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{Y}} \|\mathbf{Y} - \mathbf{AB}\|_F^2 \quad (41a)$$

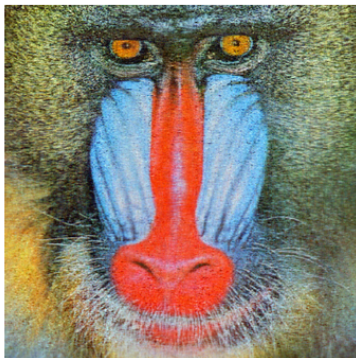
$$\text{s.t. } y_{ij} = z_{ij}, \quad \forall (i, j) \in \Omega, \quad (41b)$$

where  $\mathbf{A} \in \mathbb{R}^{m \times r}$ ,  $\mathbf{B} \in \mathbb{R}^{r \times n}$ ,  $\mathbf{Y} \in \mathbb{R}^{m \times m}$ , and  $\Omega$  is an index set indicating the *available* elements of  $\mathbf{Z}$ .

- (1) SVD does NOT work since we don't have the full  $\mathbf{Z}$ !
- (2) We can do alternating optimization w.r.t.  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{Y}$ .
  - Optimization w.r.t.  $\mathbf{A}$  is simply LS  $\mathbf{A}^* = \mathbf{Y}\mathbf{B}^\dagger$ .
  - Optimization w.r.t.  $\mathbf{B}$  is also LS  $\mathbf{B}^* = \mathbf{A}^\dagger\mathbf{Y}$ .
  - Optimization w.r.t.  $\mathbf{Y}$  is simply

$$y_{ij}^* = \begin{cases} [\mathbf{A}^* \mathbf{B}^*]_{i,j}, & \text{for } (i, j) \notin \Omega; \\ z_{ij}, & \text{for } (i, j) \in \Omega. \end{cases}$$

# Toy Demonstration of Low-Rank Matrix Completion



**Left:** An incomplete image with 40% missing pixels.

**Right:** the low-rank matrix completion result ( $r = 120$ ).

# Nonnegative Matrix Factorization (NMF)

Consider again the low-rank factorization problem

$$\min_{\mathbf{A} \in \mathbb{R}^{m \times r}, \mathbf{B} \in \mathbb{R}^{r \times n}} \|\mathbf{Y} - \mathbf{AB}\|_F^2. \quad (42)$$

The solution is not unique: if  $(\mathbf{A}^*, \mathbf{B}^*)$  is a solution to the above problem, then  $(\mathbf{A}^* \mathbf{Q}^T, \mathbf{Q} \mathbf{B}^*)$  for any orthogonal  $\mathbf{Q}$  is also a solution.

# Nonnegative Matrix Factorization (NMF)

Consider again the low-rank factorization problem

$$\min_{\mathbf{A} \in \mathbb{R}^{m \times r}, \mathbf{B} \in \mathbb{R}^{r \times n}} \|\mathbf{Y} - \mathbf{AB}\|_F^2. \quad (42)$$

The solution is not unique: if  $(\mathbf{A}^*, \mathbf{B}^*)$  is a solution to the above problem, then  $(\mathbf{A}^* \mathbf{Q}^T, \mathbf{Q} \mathbf{B}^*)$  for any orthogonal  $\mathbf{Q}$  is also a solution.

**Nonnegative Matrix Factorization (NMF):**

$$\min_{\mathbf{A} \in \mathbb{R}^{m \times r}, \mathbf{B} \in \mathbb{R}^{r \times n}} \|\mathbf{Y} - \mathbf{AB}\|_F^2, \text{ s.t. } \mathbf{A} \geq \mathbf{0}, \mathbf{B} \geq \mathbf{0}, \quad (43)$$

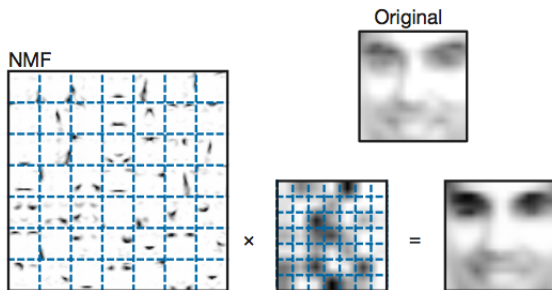
where  $\mathbf{X} \geq \mathbf{0}$  means that  $\mathbf{X}$  is element-wise non-negative.

- (1) Found to be able to extract meaningful features (by empirical studies).
- (2) Under some conditions, the NMF solution is provably unique.
- (3) Numerous applications, e.g., in machine learning, signal processing, remote sensing.

# NMF Examples

## 1 Image Processing [R3]:

- $A \geq 0$  constraints the basis elements to be nonnegative.
- $B \geq 0$  imposes an additive reconstruction.

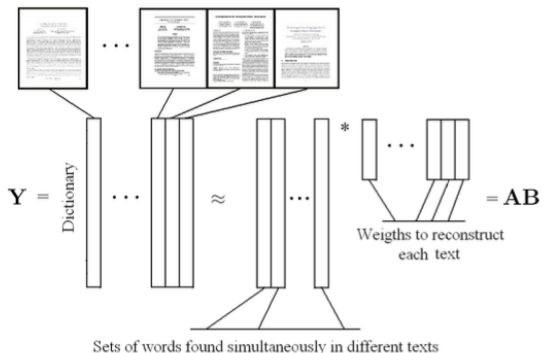


- The basis elements extract facial features such as eyes, nose and lips.



# Examples (cont'd)

## 2 Text Mining:



- Basis elements allow us to recover different topics;
- Weights allow us to assign each text to its corresponding topics.

## NMF (cont'd)

- (1) NMF is **NP-hard** in general.
- (2) **A practical way to go**: alternating optimization w.r.t.  $\mathbf{A}$  and  $\mathbf{B}$ .
- (3) For example, for a given  $\mathbf{A}$ , the  $i^{\text{th}}$  column of  $\mathbf{B}$  is updated by solving a nonnegative LS (NLS) problem:

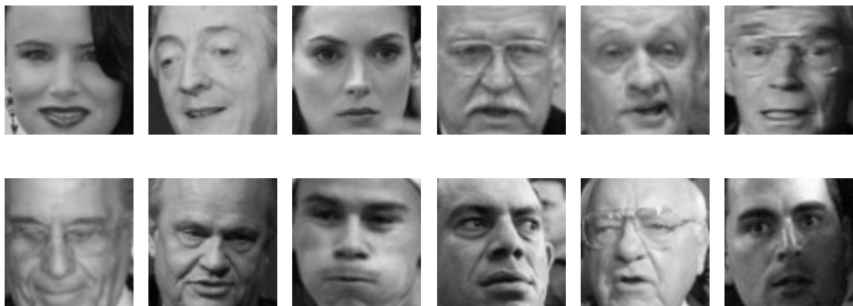
$$\min_{\mathbf{b}_i \in \mathbb{R}^r} \|\mathbf{y}_i - \mathbf{A}\mathbf{b}_i\|_2^2, \text{ s.t. } \mathbf{b}_i \geq \mathbf{0}. \quad (44)$$

NLS is convex, and there are many NLS algorithms, e.g., projected gradient

$$\mathbf{b}_i^{(j+1)} = \left[ \mathbf{b}_i^{(j)} + \alpha^{(j)} \mathbf{A}^T \left( \mathbf{y}_i - \mathbf{A}\mathbf{b}_i^{(j)} \right) \right]^+, \quad (45)$$

where  $\alpha^{(j)} > 0$  is some step size,  $[\cdot]^+$  is the elementwise projection onto negative numbers.

# Toy Demonstration of NMF



A face image dataset. Image size =  $101 \times 101$ ; number of face images  $n = 13232$ . Each  $\mathbf{x}_i$ ,  $i = 1, \dots, n$ , is the vectorization of one face image, yielding  $m = 101 \times 101 = 10201$ .

## Toy Demonstration of NMF: NMF-Extracted Features

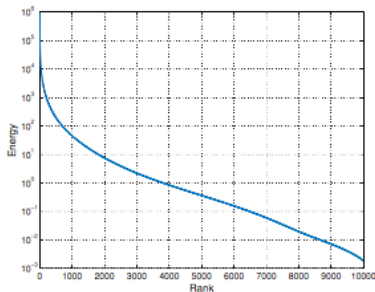


NMF settings:  $r = 49$ , Lee-Seung multiplicative update with 5000 iterations.

# Toy Demonstration of NMF: Comparison with PCA



Mean face

1st principal left  
singular vector2nd principal left  
singular vector3th principal left  
singular vectorlast principal left  
singular vector

Energy Concentration

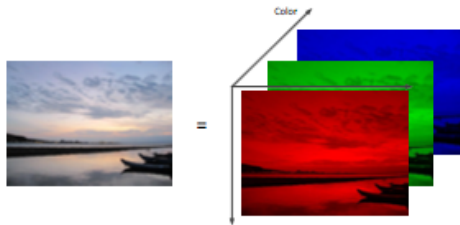
# Table of Contents

- 1 Sparse Recovery
- 2 Low-Rank Matrix Factorization
- 3 Tensor Decomposition

# Tensor

A **tensor** is a multi-way numerical array. An  $N$ -way tensor is denoted by  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$  and its entries by  $x_{i_1, i_2, \dots, i_N}$ .

- (1) Natural extension of matrices (which are two-way tensors).
- (2) Example: a color picture is a 3-way tensor, video 4-way.



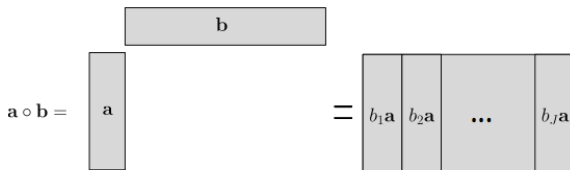
- (3) Applications: blind signal separation, chemometrics, data mining,  $\dots$
- (4) **Focus:** decomposition for 3-way tensors  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  (sufficiently complicated).

# Outer Product

The **outer product** of  $\mathbf{a} \in \mathbb{R}^I$  and  $\mathbf{b} \in \mathbb{R}^J$  is an  $I \times J$  matrix, given by

$$\mathbf{a} \circ \mathbf{b} = \mathbf{a}\mathbf{b}^T = [b_1\mathbf{a}, b_2\mathbf{a}, \dots, b_J\mathbf{a}], \quad (46)$$

where “ $\circ$ ” denotes the outer product operator.





# Outer Product (cont'd)

The outer product of a matrix  $\mathbf{A} \in \mathbb{R}^{I \times J}$  and a vector  $\mathbf{c} \in \mathbb{R}^K$ , denoted by  $\mathbf{A} \circ \mathbf{c}$ , is a three-way  $I \times J \times K$  tensor that takes the following form:

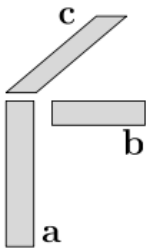
$$\mathbf{A} \circ \mathbf{c} = \begin{array}{c} \text{c} \\ \text{A} \end{array} = \begin{array}{c} c_K \mathbf{A} \\ \vdots \\ c_2 \mathbf{A} \\ c_1 \mathbf{A} \end{array}$$

Specifically, if we let  $\mathcal{X} = \mathbf{A} \circ \mathbf{c}$ , then

$$\mathcal{X}(:, :, k) = c_k \mathbf{A}, \quad k = 1, \dots, K. \quad (47)$$

## Outer Product (cont'd)

Tensors in the form of  $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$  are called rank-1 tensors.

$$\mathbf{a} \circ \mathbf{b} \circ \mathbf{c} =$$


# Tensor Decomposition [R5]

**Problem:** decompose  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  as

$$x_{i,j,k} = \sum_{r=1}^R a_{ir} b_{jr} c_{kr}, \quad (48)$$

for some  $a_{ir}$ ,  $b_{jr}$  and  $c_{kr}$ ,  $i = 1, \dots, I$ ,  $j = 1, \dots, J$ ,  $k = 1, \dots, K$ ,  $r = 1, \dots, R$ , or equivalently,

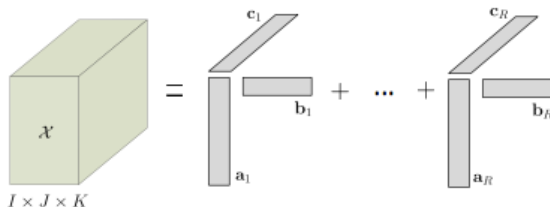
$$\mathcal{X} = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad (49)$$

where  $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_R] \in \mathbb{R}^{I \times R}$ ,  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_R] \in \mathbb{R}^{J \times R}$ , and  $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_R] \in \mathbb{R}^{K \times R}$ .

# Tensor Decomposition (cont'd)

Recall the tensor decomposition expression:

$$\mathcal{X} = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad (50)$$



- (1) A sum of rank-1 tensors.
- (2) The smallest  $R$  that satisfies (50) is called the **rank** of the tensor.
- (3) Many names: tensor rank decomposition, canonical polyadic decomposition (CPD), parallel factor analysis (PARAFAC), CANDECOMP.
- (4) It may be regarded as **a generalization of the matrix singular value decomposition (SVD) to tensors.**

# Uniqueness of Tensor Decomposition

- The low-rank matrix factorization problem  $\mathbf{X} = \mathbf{A}\mathbf{B}$  is not unique.
- We also have many matrix decompositions: SVD, QR,  $\dots$

## Theorem 5

Let  $(\mathbf{A}, \mathbf{B}, \mathbf{C})$  be a factor for  $\mathcal{X} = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ . If

$$\text{spark}(\mathbf{A}) + \text{spark}(\mathbf{B}) + \text{spark}(\mathbf{C}) \geq 2R + 5, \quad (51)$$

then  $(\mathbf{A}, \mathbf{B}, \mathbf{C})$  is the unique tensor decomposition factor for  $\mathcal{X}$  up to a common column permutation and scaling.

**Implication:** under some mild conditions with the sparks of  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$ , low-rank tensor decomposition is essentially unique.

# Slabs/Slices

A **slab** (a.k.a. slice) of a tensor is a matrix obtained by fixing one index of the tensor.

Horizontal slabs:  $\left\{ \mathbf{X}_i^{(1)} = \mathcal{X}(i, :, :) \right\}_{i=1}^I$

Lateral slabs:  $\left\{ \mathbf{X}_j^{(2)} = \mathcal{X}(:, j, :) \right\}_{j=1}^J$

Frontal slabs:  $\left\{ \mathbf{X}_k^{(3)} = \mathcal{X}(:, :, k) \right\}_{k=1}^K$

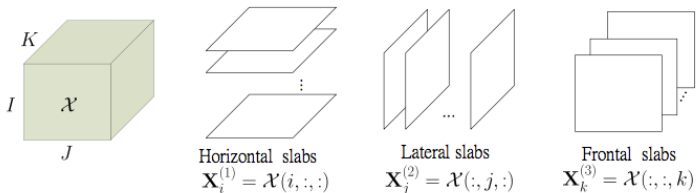
# Slabs/Slices

A **slab** (a.k.a. slice) of a tensor is a matrix obtained by fixing one index of the tensor.

Horizontal slabs:  $\left\{ \mathbf{X}_i^{(1)} = \mathcal{X}(i, :, :) \right\}_{i=1}^I$

Lateral slabs:  $\left\{ \mathbf{X}_j^{(2)} = \mathcal{X}(:, j, :) \right\}_{j=1}^J$

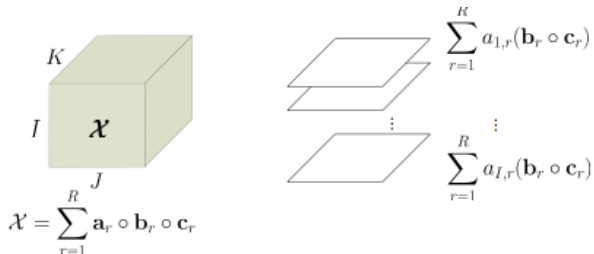
Frontal slabs:  $\left\{ \mathbf{X}_k^{(3)} = \mathcal{X}(:, :, k) \right\}_{k=1}^K$



# PARAFAC Formulation

Consider the horizontal slabs as an example.

$$\mathbf{X}_i^{(1)} = \sum_{r=1}^R a_{i,r} (\mathbf{b}_r \circ \mathbf{c}_r) = \sum_{r=1}^R a_{i,r} \mathbf{b}_r \mathbf{c}_r^T. \quad (52)$$



We can write

$$\mathbf{X}_i^{(1)} = \mathbf{B} \mathbf{D}_{\mathbf{A}(i,:)} \mathbf{C}^T, \quad (53)$$

where  $\mathbf{D}_{\mathbf{A}(i,:)} \triangleq \text{diag}(\mathbf{A}(i, :))$ .



# PARAFAC Formulation (cont'd)

**Khatri-Rao (KR) product:** the KR product of  $\mathbf{A} \in \mathbb{R}^{I \times R}$  and  $\mathbf{B} \in \mathbb{R}^{J \times R}$  is defined as

$$\mathbf{A} \odot \mathbf{B} \triangleq \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_R \otimes \mathbf{b}_R \end{bmatrix}, \quad (54)$$

where  $\otimes$  denotes Kronecker product (see Chap. 9). In essence, the KR product is the “column-wise” Kronecker product.

# PARAFAC Formulation (cont'd)

**Khatri-Rao (KR) product:** the KR product of  $\mathbf{A} \in \mathbb{R}^{I \times R}$  and  $\mathbf{B} \in \mathbb{R}^{J \times R}$  is defined as

$$\mathbf{A} \odot \mathbf{B} \triangleq [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \cdots \quad \mathbf{a}_R \otimes \mathbf{b}_R], \quad (54)$$

where  $\otimes$  denotes Kronecker product (see Chap. 9). In essence, the KR product is the “column-wise” Kronecker product.

**A key KR property:** let  $\mathbf{D} \triangleq \text{diag}(\mathbf{d})$ . We have

$$\text{vec}(\mathbf{A}\mathbf{D}\mathbf{B}^T) = (\mathbf{B} \odot \mathbf{A})\mathbf{d}. \quad (55)$$

**Proof.**

Recalling the horizontal slabs expression  $\mathbf{X}_i^{(1)} = \mathbf{B}\mathbf{D}_{\mathbf{A}(i,:)}\mathbf{C}^T$ , where  $\mathbf{D}_{\mathbf{A}(i,:)} \triangleq \text{diag}(\mathbf{A}(i,:))$ , it can be readily rewritten as

$$\text{vec}(\mathbf{X}_i^{(1)}) = (\mathbf{C} \odot \mathbf{B})\mathbf{A}^T(i,:), \quad (56)$$

by virtue of the vectorization property. □

# PARAFAC Formulation (cont'd)

**Khatri-Rao (KR) product:** the KR product of  $\mathbf{A} \in \mathbb{R}^{I \times R}$  and  $\mathbf{B} \in \mathbb{R}^{J \times R}$  is defined as

$$\mathbf{A} \odot \mathbf{B} \triangleq [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \cdots \quad \mathbf{a}_R \otimes \mathbf{b}_R], \quad (54)$$

where  $\otimes$  denotes Kronecker product (see Chap. 9). In essence, the KR product is the “column-wise” Kronecker product.

**A key KR property:** let  $\mathbf{D} \triangleq \text{diag}(\mathbf{d})$ . We have

$$\text{vec}(\mathbf{A} \mathbf{D} \mathbf{B}^T) = (\mathbf{B} \odot \mathbf{A}) \mathbf{d}. \quad (55)$$

**Proof.**

Recalling the horizontal slabs expression  $\mathbf{X}_i^{(1)} = \mathbf{B} \mathbf{D}_{\mathbf{A}(i,:)} \mathbf{C}^T$ , where  $\mathbf{D}_{\mathbf{A}(i,:)} \triangleq \text{diag}(\mathbf{A}(i,:))$ , it can be readily rewritten as

$$\text{vec}(\mathbf{X}_i^{(1)}) = (\mathbf{C} \odot \mathbf{B}) \mathbf{A}^T(i,:), \quad (56)$$

by virtue of the vectorization property. □

Roughly speaking, (56) implies

$$\mathbf{A}^T(i,:) = (\mathbf{C} \odot \mathbf{B})^\dagger \text{vec}(\mathbf{X}_i^{(1)}). \quad (57)$$

# PARAFAC Formulation (cont'd)

By the trick above, we can write:

$$\begin{aligned} \text{Horizontal slabs: } \mathbf{X}_i^{(1)} &= \mathcal{X}(i, :, :) = \mathbf{B} \mathbf{D}_{\mathbf{A}(i, :)} \mathbf{C}^T \\ \text{vec} \left( \mathbf{X}_i^{(1)} \right) &= (\mathbf{C} \odot \mathbf{B}) \mathbf{A}^T(i, :) \end{aligned}$$

$$\begin{aligned} \text{Lateral slabs: } \mathbf{X}_j^{(2)} &= \mathcal{X}(:, j, :) = \mathbf{C} \mathbf{D}_{\mathbf{B}(j, :)} \mathbf{A}^T \\ \text{vec} \left( \mathbf{X}_j^{(2)} \right) &= (\mathbf{A} \odot \mathbf{C}) \mathbf{B}^T(j, :) \end{aligned}$$

$$\begin{aligned} \text{Frontal slabs: } \mathbf{X}_k^{(3)} &= \mathcal{X}(:, :, k) = \mathbf{A} \mathbf{D}_{\mathbf{C}(k, :)} \mathbf{B}^T \\ \text{vec} \left( \mathbf{X}_k^{(3)} \right) &= (\mathbf{B} \odot \mathbf{A}) \mathbf{C}^T(k, :) \end{aligned}$$

**Observation:**

- (1) Fixing  $\mathbf{B}$  and  $\mathbf{C}$ , solving for  $\mathbf{A}$  is a linear system problem.
- (2) Fixing  $\mathbf{A}$  and  $\mathbf{C}$ , solving for  $\mathbf{B}$  is a linear system problem.
- (3) Fixing  $\mathbf{A}$  and  $\mathbf{B}$ , solving for  $\mathbf{C}$  is a linear system problem.

# PARAFAC Formulation (cont'd)

A tensor decomposition formulation: given  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  and  $R > 0$ , solve

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \left\| \mathcal{X} - \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right\|_F^2. \quad (58)$$

- (1) NP-hard in general.
- (2) Can be readily handled by alternating optimization w.r.t.  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$ , e.g., optimization w.r.t.  $\mathbf{A}$  while fixing  $\mathbf{B}$  and  $\mathbf{C}$  is

$$\min_{\mathbf{A}} \sum_{i=1}^I \left\| \mathbf{X}_i^{(1)} - \mathbf{B} \mathbf{D}_{\mathbf{A}(i,:)} \mathbf{C}^T \right\|_F^2 = \min_{\mathbf{A}} \sum_{i=1}^I \left\| \text{vec} \left( \mathbf{X}_i^{(1)} \right) - (\mathbf{C} \odot \mathbf{B}) \mathbf{A}^T(i,:) \right\|_2^2, \quad (59)$$

and a solution is

$$(\mathbf{A}^*(i,:))^T = (\mathbf{C} \circ \mathbf{B})^\dagger \text{vec} \left( \mathbf{X}_i^{(1)} \right), \quad i = 1, \dots, I. \quad (60)$$

# Applications

## Application I: Blind PARAFAC receivers for DS-CDMA systems.<sup>7</sup>

The output of ant.  $k$ , for symbol  $n$  and chip  $p$ :

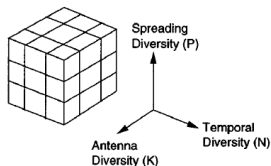


Figure 1: Datacube in diversity space.

$$x_{k,n,p} = \sum_{m=0}^M \alpha(k, m) c_m(p) s_m(n), \quad (61)$$

where

- $\alpha(k, m)$ : channel fading btw user  $m$  and ant.  $k$ ;
- $c_m(p)$ :  $p^{\text{th}}$  chip of the spreading code of user  $m$ ;
- $s_m(n)$ :  $n^{\text{th}}$  symbol transmitted by user  $m$ .

<sup>7</sup>N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA systems," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 810–823 March 2000.

# Applications (cont'd)

**Application II:** Tensor space-time-frequency coding with semi-blind receivers for MIMO wireless communication systems.<sup>8</sup>

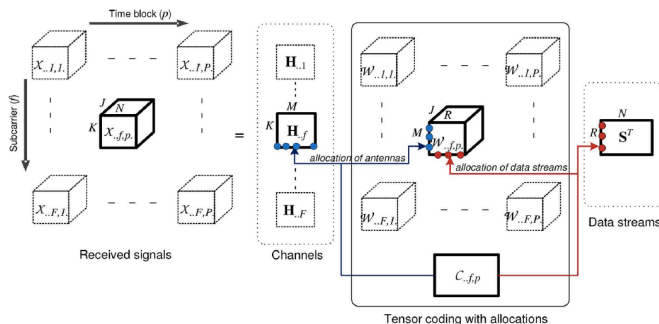


Figure 2: Block-diagram of the tensor space-time-frequency coding system.

**System Parameters:**  $m$  ( $k$ ): Tx (Rx) antenna index;  $r$ : data stream index;  $p$ : time slot (data block) index;  $n$ : symbol period index;  $j$ : chip period index;  $f$ : sub-carrier index.

<sup>8</sup>G. Favier and A. L. F. de Almeida, "Tensor space-time-frequency coding with semi-blind receivers for MIMO wireless communication systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5987–6002, Nov. 2014.

# Applications (cont'd)

**Application III:** Tensor-based joint channel and symbol estimation for two-way MIMO relaying systems.<sup>9</sup>

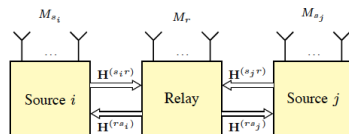


Figure 3: Two-way model with a pair of sources  $i$  and  $j$ .

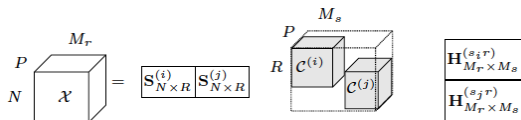


Figure 4: 3-D illustration of a block Tucker-2 tensor.

<sup>9</sup>W. C. Freitas, G. Favier and A. L. F. de Almeida, "Tensor-based joint channel and symbol estimation for two-way MIMO relaying systems," *IEEE Signal Processing Letters*, 2018, [Online] Available:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8565909>



# References

- [R1] Wotao Yin, *Sparse Optimization*, Course Handouts, UCLA, available online at <http://www.math.ucla.edu/~wotaoyin/summer2013/lectures.html>
- [R2] ACM SIGKDD and Netflix. *Proceedings of KDD Cup and Workshop*, 2007. Proceedings available online at <http://www.cs.uic.edu/liub/KDD-cup-2007/proceedings.html>
- [R3] D. D. Lee and H. S. Seung. “Learning the parts of objects by non-negative matrix factorization,” *Nature*, 1999.
- [R4] E. J. Candes, J. Romberg, and T. Tao. “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, Feb. 2006, pp. 489–509.
- [R5] Tamara G. Kolda and Brett W. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500.
- [R6] E. Michael, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, 2010, p. 26.

**Thank you  
for your attention!**



**Kai Lu**

**E-mail:** lukai86@mail.sysu.edu.cn

**Web:** <http://seit.sysu.edu.cn/teacher/1801>