An Introduction to Sparse Coding, Sparse Sensing, and Optimization

Speaker: Wei-Lun Chao

Date: Nov. 23, 2011

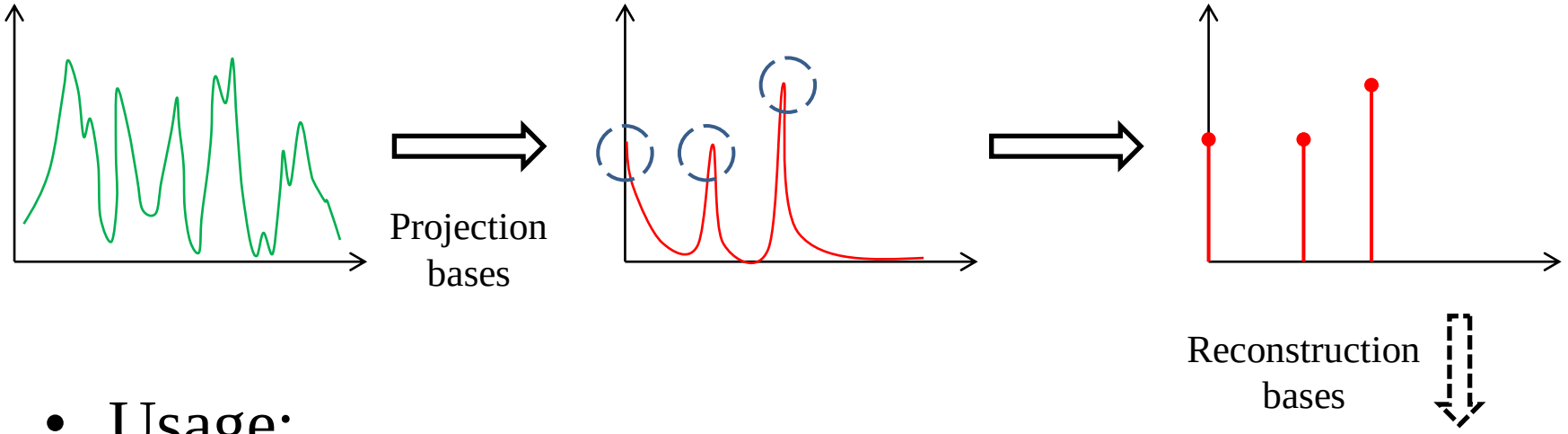
Outline

- Introduction
- The fundamental of optimization
- The idea of sparsity: coding V.S. sensing
- The solution
- The importance of dictionary
- Applications

Introduction

Introduction

- What is sparsity?



- Usage:
  - Compression
  - Analysis
  - Representation
  - Fast / sparse sensing

Introduction

- Why do we use Fourier transform and its modifications for image and acoustic compression?
  - Differentiability (theoretical)
  - Intrinsic sparsity (data-dependent)
  - Human perception (human-centric)
- Better bases for compression or representation?
  - Wavelets
  - How about data-dependent bases?
  - How about learning?

Introduction

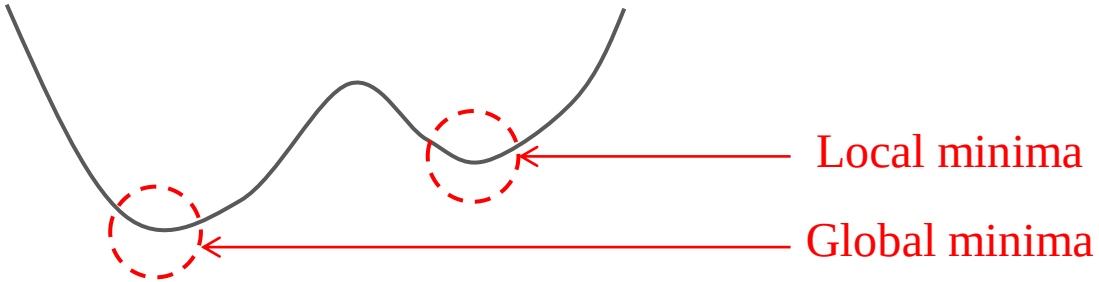
- Optimization
  - Frequently faced in algorithm design
  - Used to implement you creative idea
- Issue
  - What kinds of mathematical form and its corresponding optimization algorithms do guarantee the convergence to local or global optima?

The Fundamental of Optimization

A Warming-up Question

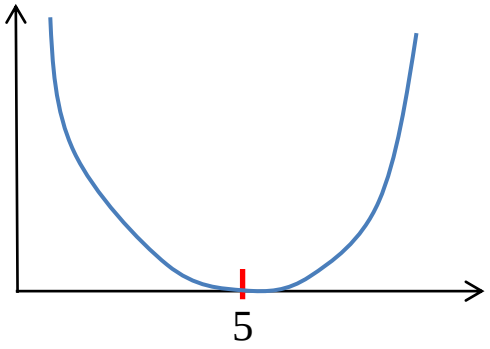
- How do you solve the following problems?

(1)



(2) $\min_{w} f(w) = (w-5)^2$ ⟹

(a) Plot



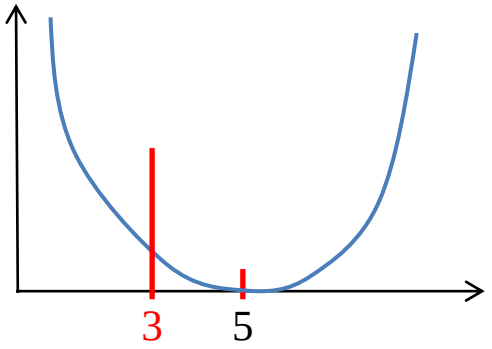
(b) Take derivatives, check = 0

An Advanced Question

- How about the following questions?

(3) $\min_{w} \sum_{n=1}^{N} f_n(w)$ ⟹ (a) Plot? (b) Take derivative = 0?

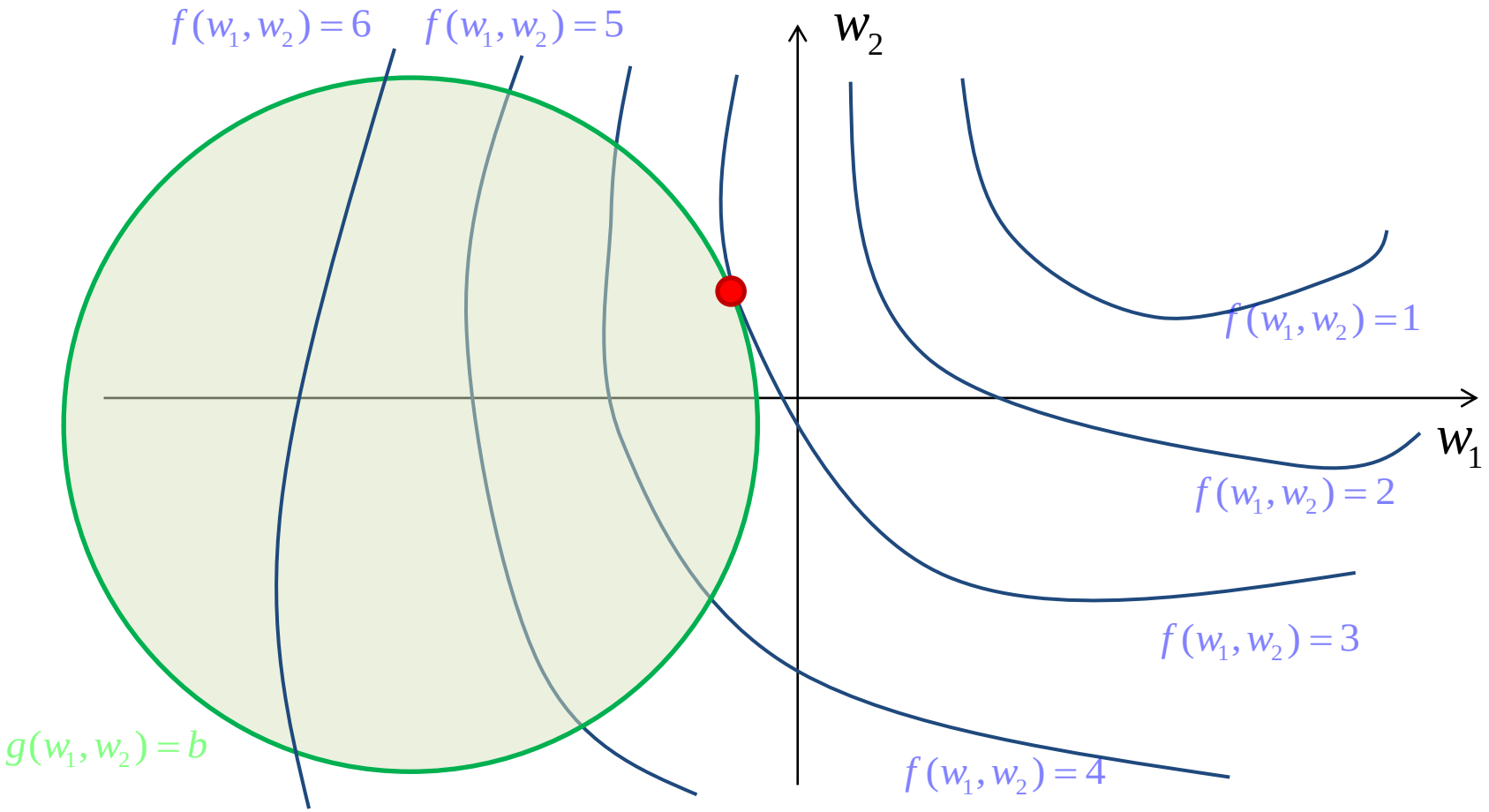
(4) $\min_{w} f(w) = (w-5)^2$ s.t. $w \leq 3$ ⟹ Derivative?



(5) $\min_{\boldsymbol{w}} \sum_{n=1}^{N} f_n(\boldsymbol{w})$ s.t. $w_i \leq b_i$ ⟹ How to do?

Illustration

- 2-D case: $\min_{w_1, w_2} f(w_1, w_2),\ \text{s.t. } g(w_1, w_2) \le b$



How to Solve?

- Thanks to……
  - Lagrange multiplier
  - Linear programming, quadratic programming, and recently, convex optimization
- Standard form:

$$\min_{\boldsymbol{w}} f_0(\boldsymbol{w})$$

$$\text{s.t. } h_i(\boldsymbol{w}) \le b_i,\ i = 1, ..., m$$

$$\text{s.t. } g_i(\boldsymbol{w}) = c_i,\ i = 1, ..., n$$

Fallacy

- A quadratic programming problem with constraints

$$\min_{x} \left(A\boldsymbol{x} - \boldsymbol{b} \right)^2$$

The importance of each food

Personal nutrient need

$$\begin{bmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots\cdots & \boldsymbol{a}_N \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} | \\ b \\ | \end{bmatrix}$$

Nutrient content of each food

$$x_i \geq 0$$

(1) Take derivative (x)

$$\boldsymbol{x} = (A^T A)^{-1} A^T \boldsymbol{b}$$

choose x_i with $x_i \geq 0$

(2) Quadratic programming (o)

$$\arg\min_{x} \left(A\boldsymbol{x} - b \right)^2, \ \text{s.t.}\ x_i \geq 0$$

(3) Sparse coding (o)

The Idea of Sparsity

What is Sparsity?

- Think about a problem?

Assume full rank, $N > d$

$$\min_{x} \left(Ax - b \right)^2$$

$$\underbrace{\begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots\cdots & a_N \\ | & | & & | \end{bmatrix}}_{N} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} | \\ b \\ | \end{bmatrix} \in R^d$$

$\Rightarrow$

Many x can achieve

$$\min_{x} \left(Ax - b \right)^2 = 0$$

$\Downarrow$

Which do you want?

$\Leftarrow$

Choose the x with the least nonzero component

$$\arg\min_{x} \|x\|_0 \text{, s.t. } \left(Ax - b \right)^2 = 0$$

Why Sparsity?

- The more concise, the more better
- In some domain, there naturally exists a sparse latent vector that controls the data we saw. (ex. MRI, music)

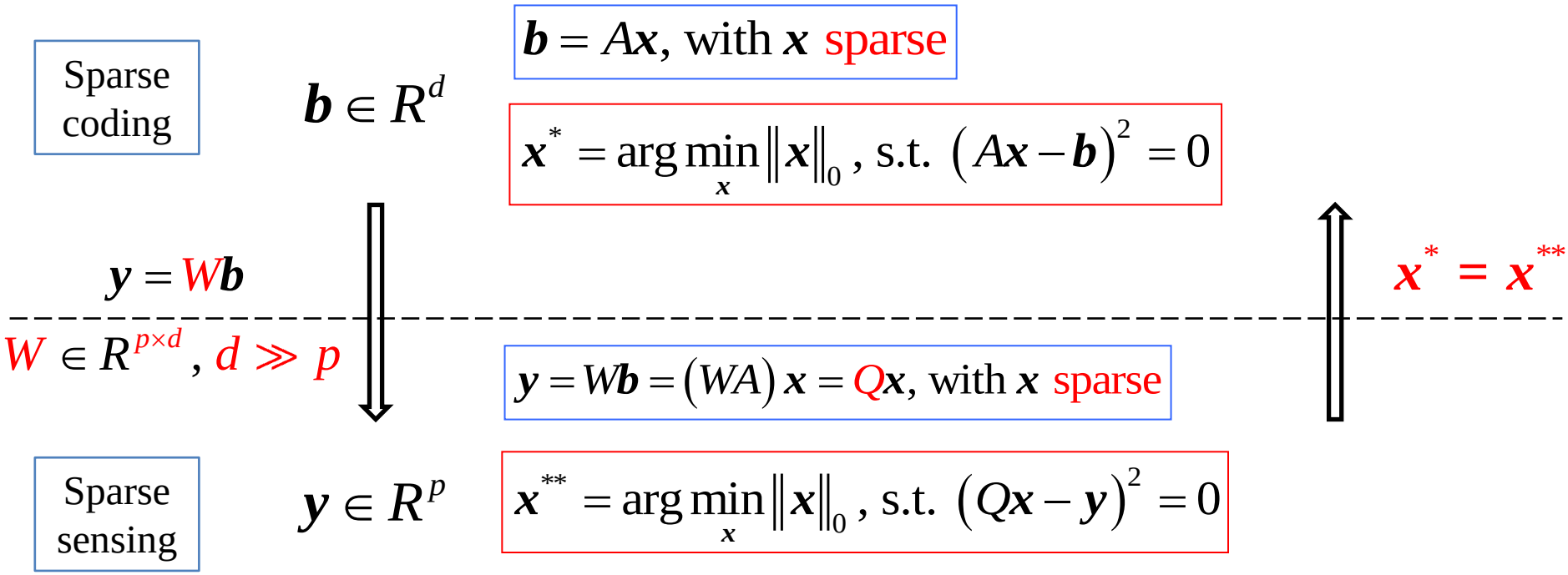
$$\begin{bmatrix} | \\ \boldsymbol{b} \\ | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots\cdots & \boldsymbol{a}_d \\ | & | & & | \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ x_i \\ \vdots \\ x_j \\ \vdots \\ 0 \end{bmatrix} + (\text{noise})$$

A *k*-sparse domain means that each ***b*** can be constructed by a ***x*** vector with at most *k* nonzero element

- In some domain, samples from the same class have the sparse property.
- The domain can be learned.

Sparse Sensing VS. Sparse Coding

- Assume that: We have $A \in R^{d \times N}$, $N > d$. Now an observation $\boldsymbol{b} \in R^{d}$ comes in



Note: p is based on the sparsity of the data (on k)

Sparse Sensing

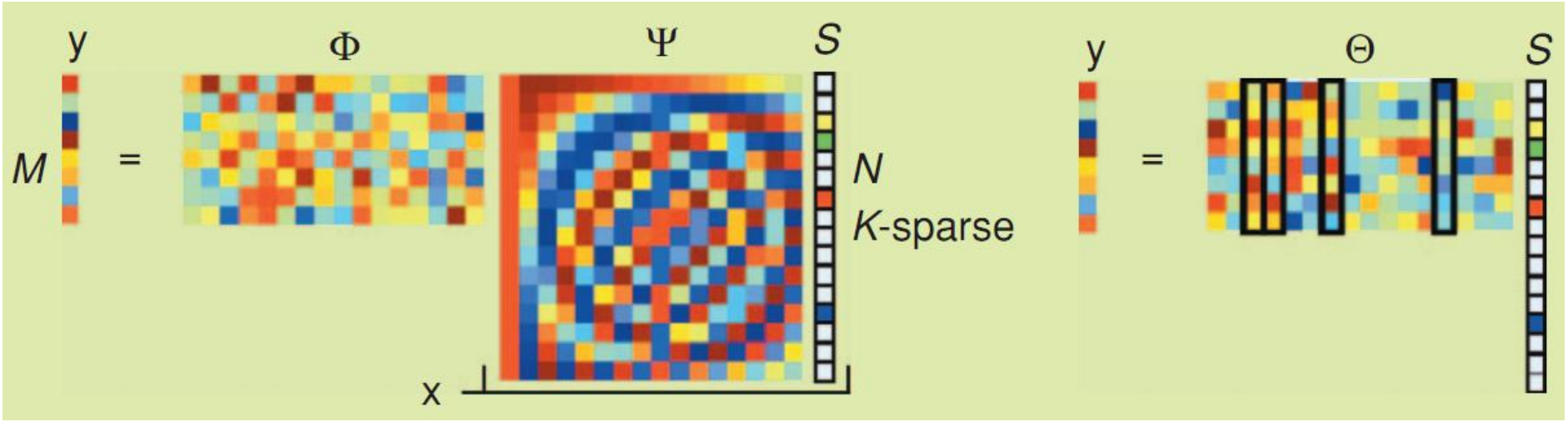
$\boldsymbol{b} \in R^d$ $\boxed{\boldsymbol{b} = A\boldsymbol{x}, \text{ with } \boldsymbol{x} \text{ sparse}}$

$\boldsymbol{y} = W\boldsymbol{b}$

$W \in R^{p \times d}, d \gg p$

$\boldsymbol{x}^* = \boldsymbol{x}^{**}$

$\boldsymbol{y} \in R^p$ $\boxed{\boldsymbol{y} = W\boldsymbol{b} = (WA)\boldsymbol{x} = Q\boldsymbol{x}, \text{ with } \boldsymbol{x} \text{ sparse}}$



Sparse Sensing VS. Sparse Coding

- Sparse sensing (compressed sensing):
  - It spends much time or money to get ***b***, so get ***y*** first then recover ***b***
- Sparse coding (sparse representation):
  - Believe that there exists the sparse property in the data, otherwise sparse representation means nothing.
  - ***x*** is used to be the feature of ***b***
  - ***x*** can be used to efficiently store ***b*** and reconstruct ***b***

The Solution

How to Get The Sparse Solution?

- There is no algorithm other than exhaustively searching to solve:

$$\boldsymbol{x}^{*} = \arg\min_{\boldsymbol{x}} \left\| \boldsymbol{x} \right\|_0 , \text{ s.t. } \left(A\boldsymbol{x} - \boldsymbol{b} \right)^2 = 0$$

- While in some situations (ex. special form of A), the solution of l_1 minimization approaches the one of l_0 minimization

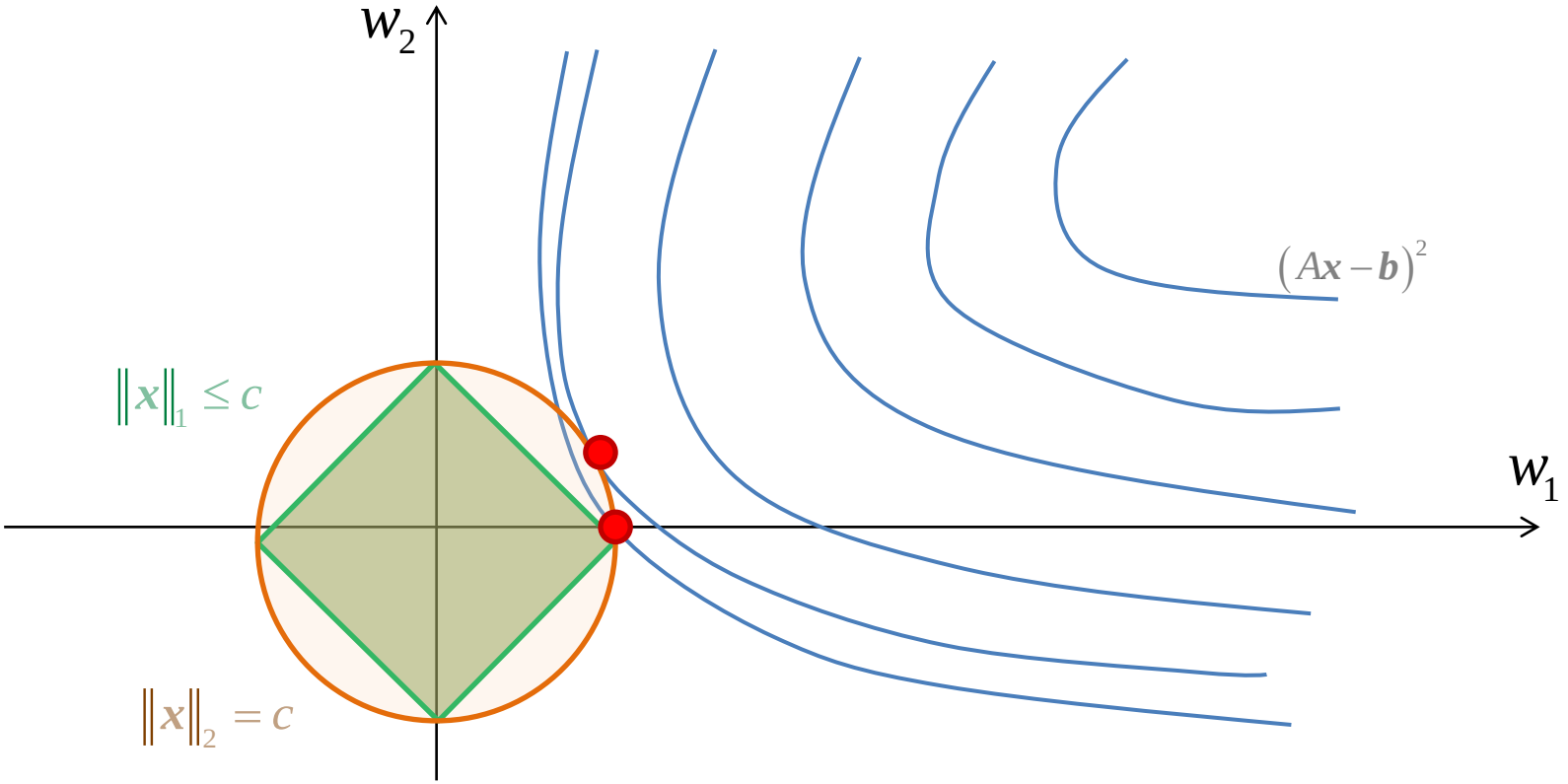
$$\boldsymbol{x}^{***} = \arg\min_{\boldsymbol{x}} \left\| \boldsymbol{x} \right\|_1 = \sum_{n=1}^{N} \left| x^{(n)} \right| , \text{ s.t. } \left(A\boldsymbol{x} - \boldsymbol{b} \right)^2 = 0$$

$$\boldsymbol{x}^{***} = \boldsymbol{x}^{*}$$

Why l_1?

- Question 1: Why l_1 can result in a sparse solution?

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{, s.t. } (A\boldsymbol{x}-\boldsymbol{b})^2 = 0 \quad \rightarrow \quad \arg\min_{\boldsymbol{x}} (A\boldsymbol{x}-\boldsymbol{b})^2 \text{, s.t. } \|\boldsymbol{x}\|_1 \leq c$$



Why l_1?

- Question 2: Why the sparse solution achieved by l_1 minimization approaches the one of l_0 minimization?
  - ➢ This is a matter or Mathematics
  - ➢ No matter how, sparse representation based on l_1 minimization has been widely used for pattern recognition.
  - ➢ In addition, if one doesn't care about using the sparse solution for representation (feature), it seems OK if these two solutions are not the same.

$$\boldsymbol{b} = A\boldsymbol{x}^{***}$$

$$\boldsymbol{b} = A\boldsymbol{x}^{*}$$

Noise

- Sometimes, the data is observed with nose

$$\begin{bmatrix} | \\ \boldsymbol{b} \\ | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots\cdots & \boldsymbol{a}_d \\ | & | & & | \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ x_i^* \\ \vdots \\ x_i^* \\ \vdots \\ 0 \end{bmatrix} \Longrightarrow \boldsymbol{b} = \boldsymbol{b} + noise \xrightarrow{l_0\,(l_1)\ \text{minimization}} \boldsymbol{x} \approx \boldsymbol{x}\,???$$

- The answer seems to be negative

$$\boldsymbol{b} = A\boldsymbol{x}^*, \boldsymbol{y}^* = \arg\min_{\boldsymbol{y}} \|\boldsymbol{y}\|_1, \text{ s.t. } \left(A\boldsymbol{y} - noise\right)^2 = 0$$

$$\boldsymbol{y}^* + \boldsymbol{x}^* \neq \boldsymbol{x}^*, \text{ and } \boldsymbol{y}^* \text{ is usually not sparse}$$

$$\boldsymbol{x} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1, \text{ s.t. } \left(A\boldsymbol{x} - \boldsymbol{b}\right)^2 = 0 \text{ is neither equal to } \boldsymbol{y}^* + \boldsymbol{x}^* \text{ nor to } \boldsymbol{x}^*$$

possibly not sparse

Noise

- Several ways to overcome this:

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{, s.t. } \left(A\boldsymbol{x}-\boldsymbol{b}\right)^2 = 0 \;\rightarrow\; \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{, s.t. } \|A\boldsymbol{x}-\boldsymbol{b}\|_2 \le c$$

$$\rightarrow\; \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{, s.t. } \|A\boldsymbol{x}-\boldsymbol{b}\|_1 \le c$$

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{, s.t. } \left(A\boldsymbol{x}-\boldsymbol{b}\right)^2 = 0 \;\rightarrow\; \arg\min_{\boldsymbol{z}} \|\boldsymbol{z}\|_1 \text{, s.t. } \left(\left[A \mid I\right]\boldsymbol{z}-\boldsymbol{b}\right)^2 = 0, \text{ where } \boldsymbol{z} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{t} \end{bmatrix}$$

- What is the difference between:

$$\|A\boldsymbol{x}-\boldsymbol{b}\|_2 \le c \text{ and } \|A\boldsymbol{x}-\boldsymbol{b}\|_1 \le c$$

Equivalent form

- You may also see several forms for the problem:

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{, s.t. } \|A\boldsymbol{x}-\boldsymbol{b}\|_1 \le c \quad \rightarrow \quad \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 + \lambda \|A\boldsymbol{x}-\boldsymbol{b}\|_1$$

$$\rightarrow \quad \arg\min_{\boldsymbol{x}} \|A\boldsymbol{x}-\boldsymbol{b}\|_1 \text{, s.t. } \|\boldsymbol{x}\| \le d$$

- These equivalent forms are derived from Lagrange multiplier
- There have been several publications aiming at how solving the l_1 minimization problem.

The Importance of Dictionary

Dictionary generation

- If the preceding sections, we generally assume that the (over-complete) bases A is existed and known
- However in practice, we usually need to build it:
  - Wavelet + Fourier + Haar + ……
  - Learning based on data
- How to learn?

Given a training set $\left\{\boldsymbol{b}^{(i)} \in R^d\right\}_{i=1}^{N}$, form B as $B = \left[\boldsymbol{b}^{(1)}\ \boldsymbol{b}^{(2)}\ \ldots\ldots\ \boldsymbol{b}^{(N)}\right]$

$$\left(A^*,\ X^*\right) = \arg\min_{A,\ X} \left\|B - AX\right\|_F^2 + \lambda \left\|X\right\|_1, \text{ where } X\left[\boldsymbol{x}^{(1)}\ \boldsymbol{x}^{(2)}\ \ldots\ldots\ \boldsymbol{x}^{(N)}\right]$$

- May result in over-fitting

Applications

Back to the problem we have

- A quadratic programming problem with constraints

$$\min_{x} \left(A\boldsymbol{x} - \boldsymbol{b} \right)^2$$

The importance of each food

Personal nutrient need

$$\begin{bmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots\cdots & \boldsymbol{a}_N \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} | \\ b \\ | \end{bmatrix}$$

Nutrient content of each food

$$x_i \geq 0$$

(1) Take derivative (x)

$$\boldsymbol{x} = (A^T A)^{-1} A^T \boldsymbol{b}$$

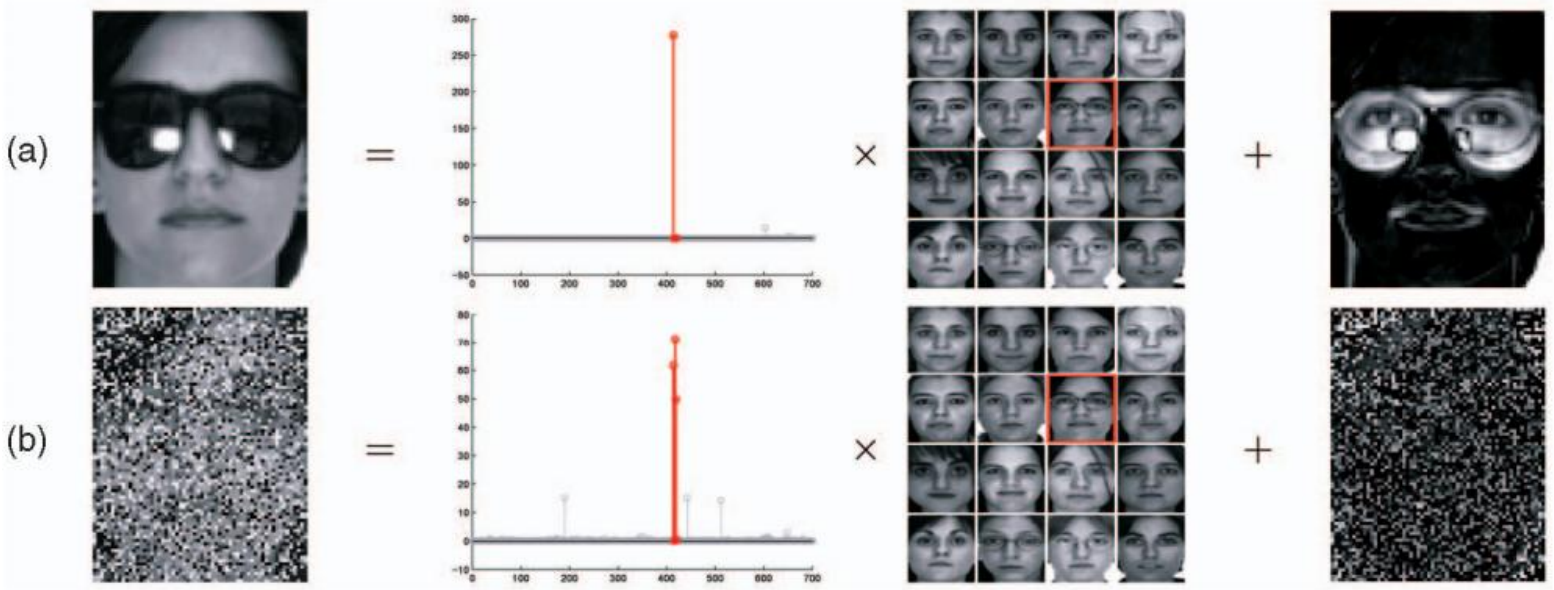
choose x_i with $x_i \geq 0$

(2) Quadratic programming (o)

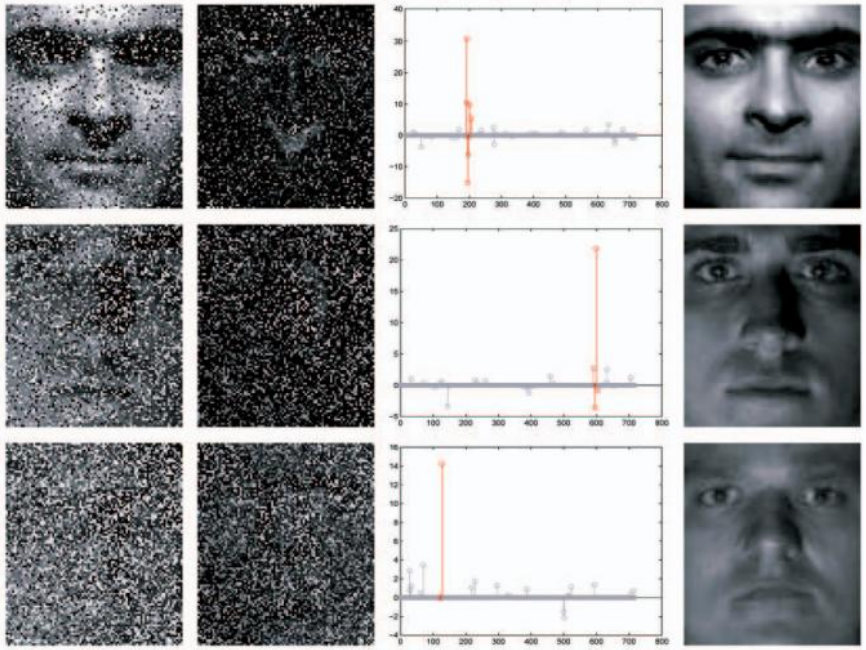
$$\arg\min_{x} \left(A\boldsymbol{x} - b \right)^2, \ \text{s.t.}\ x_i \geq 0$$

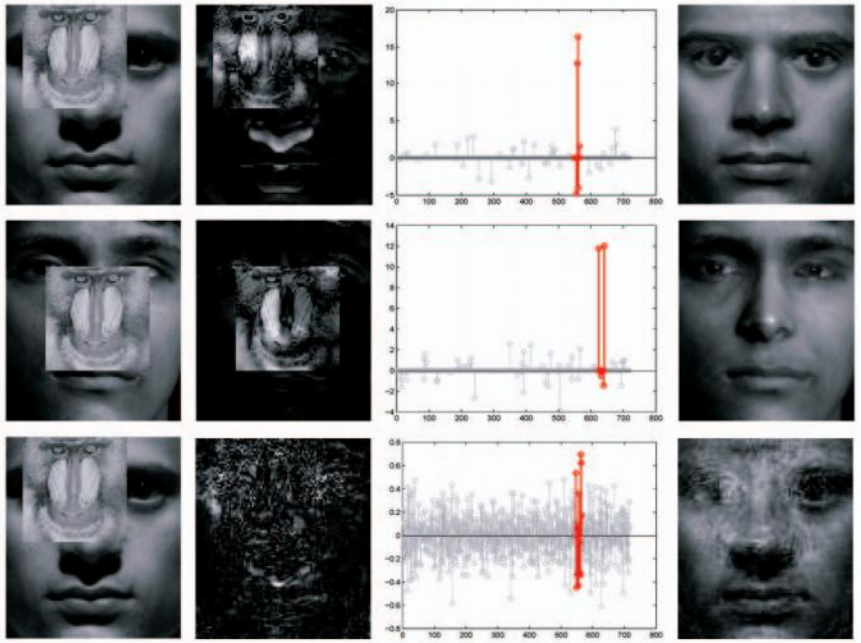
(3) Sparse coding (o)

Face Recognition (1)



Face Recognition (2)





An important issue

- When using sparse representation as a way of feature extraction, you may wonder, even if there exists the sparsity property in the data, does sparse feature really come up with better results? Does it contain any semantic meaning?
- Successful areas:
  - Face recognition
  - Digit recognition
  - Object recognition (with carful design):

    Ex. K-means → Sparse representation

De-noising

Learn a patch dictionary. For each patch, compute the sparse representation then use it to reconstruct the patch.

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 + \lambda \|A\boldsymbol{x} - \boldsymbol{b}\|_1$$

$$\boldsymbol{b} = A\boldsymbol{x}^*$$

Detection based on reconstruction

Learn a patch dictionary for a specific object. For each patch in the image, compute the sparse representation and use it to reconstruct the image. Check the error for each patch, and identify those with small error as detected object.

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 + \lambda \|A\boldsymbol{x} - \boldsymbol{b}\|_1$$

$$\boldsymbol{b} = A\boldsymbol{x}^*$$

check $\|\boldsymbol{b} - \boldsymbol{b}\|_2^2$

Maybe not over-complete

Other cases: Foreground-background detection, pedestrian detection, ……

Conclusion

What you should know

- What is the form of standard optimization?
- What is sparsity?
- What is sparse coding and sparse sensing?
- What kind of optimization method to solve it?
- Try to use it !!

Thank you for listening