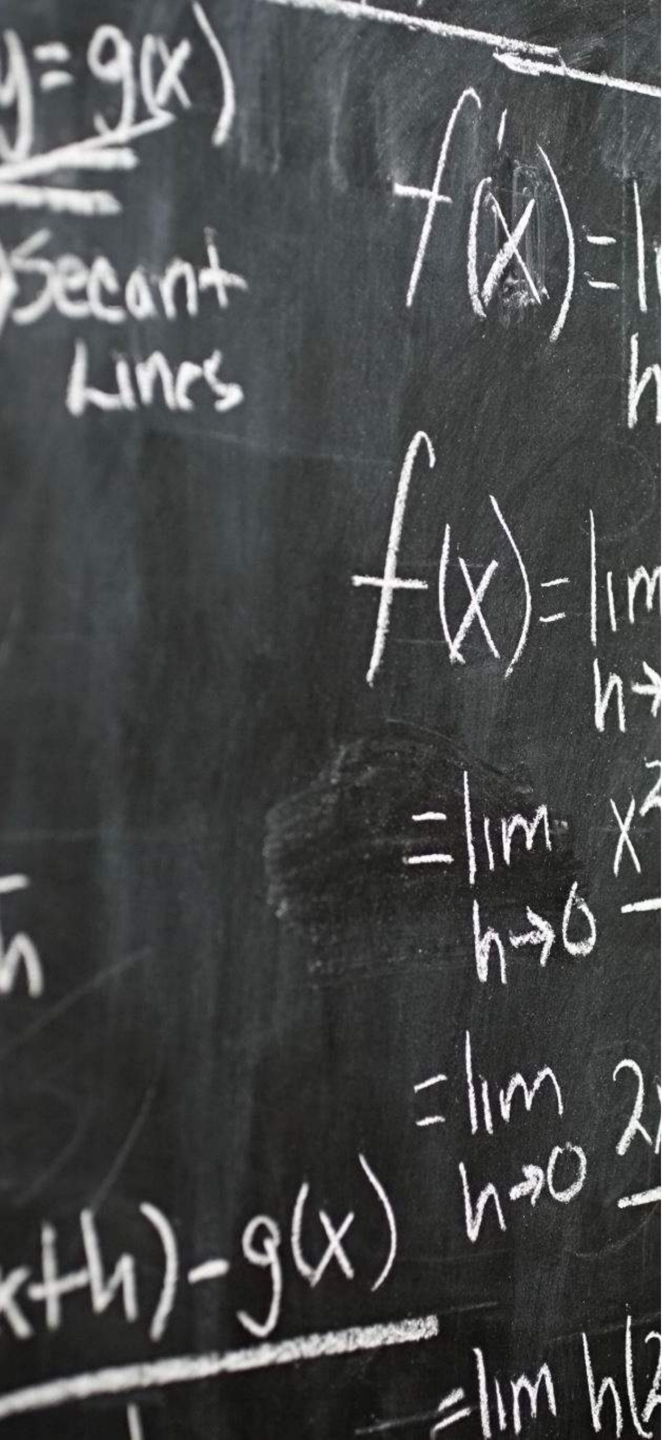
STA 5103: Selected Topics in Frontiers of Statistics III

Lecture 2: Math prerequisite

Prof. Chao Wang
Department of Statistics & Data Science,
Southern University of Science & Technology



Contents

- Convex analysis
- Singular value decomposition
- Inverse problems

Convex Analysis

A brief introduction

Convex set & Convex function

Convex Set

- A set C is convex if the line segment between any two points in C lies in C, i.e., if for any $x_1, x_2 \in C$ and any θ with $0 \leq \theta \leq 1$, we have

$$\theta x_1 + (1 - \theta) x_2 \in C$$

Convex function

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if $\mathbf{dom}\, f$ is a convex set and

$$f(\theta x + (1 - \theta) y) \leq \theta f(x) + (1 - \theta) f(y)$$

for all $x, y \in \mathbf{dom}\, f, 0 \leq \theta \leq 1$

1st /2nd conditions of convex function.

- First-order condition: differentiable f with convex domain is convex iff

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) \quad \text{for all } x, y \in \mathbf{dom}\, f$$

- Second-order conditions: for twice differentiable f with convex domain f is convex if and only if

$$\nabla^2 f(x) \succeq 0 \quad \text{for all } x \in \mathbf{dom}\, f$$

Relationship between convex set & convex function.

- α-sublevel set of $f : \mathbb{R}^n \to \mathbb{R}$:

$$C_\alpha = \{x \in \operatorname{dom} f \mid f(x) \leq \alpha\}$$

- sublevel sets of convex functions are convex (converse is false)

- epigraph of $f : \mathbb{R}^n \to \mathbb{R}$:

$$\operatorname{epi} f = \left\{(x, t) \in \mathbb{R}^{n+1} \mid x \in \operatorname{dom} f, f(x) \leq t\right\}$$

- f is convex if and only if epi f is a convex set

Convexity and global minimum

- Consider an optimization problem

$$\min f(x) \text{ s.t. } x \in \Omega,$$

where f is a convex function and Ω is a convex set. Then, any local minimum is also a global minimum.

Convexity and global minimum

- Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable convex function, and let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set. Consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C. \end{array}$$

  A vector x^* is optimal for this problem if and only if $x^* \in C$ and

$$\nabla f(x^*)^T (z - x^*) \geq 0 \text{ for all } z \in C.$$

- **(Interior Case or Unconstrained Case)** Let Ω be a subset of $\mathbb{R}^n$ and $f \in \mathcal{C}^1$ a real-value function on Ω. If x^* is a local minimizer of f over Ω and if x^* is an interior point, then $\nabla f(x^*) = 0$.

Optimality conditions in general cases

- **(First-Order Necessary Condition (FONC)).** Let Ω be a subset of $\mathbb{R}^n$ and $f \in \mathcal{C}^1$ a real-value function on Ω. If x^* is a local minimizer of f over Ω, then for any feasible direction d at x^*, we have

$$d^T \nabla f(x^*) \geq 0$$

- **(Interior Case or unconstrained case)** Let Ω be a subset of $\mathbb{R}^n$ and $f \in \mathcal{C}^1$ a real-value function on Ω. If x^* is a local minimizer of f over Ω and if x^* is an interior point, then $\nabla f(x^*) = 0$.

Lagrangian

- Standard form problem (not necessarily convex)

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& h_i(x) = 0, \quad i = 1, \ldots, p
\end{array}
$$

variable $x \in \mathbf{R}^n$, domain $\mathcal{D}$, optimal value $p^\star$

- Lagrangian: $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$, with dom $L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p$,

$$
L(x, \lambda, v) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x)
$$

- weighted sum of objective and constraint functions
- λ_i is Lagrange multiplier associated with $f_i(x) \leq 0$
- v_i is Lagrange multiplier associated with $h_i(x) = 0$

Lagrangian.

- Lagrange dual function: $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$,

$$
\begin{aligned}
g(\lambda, v) &= \inf_{x \in \mathcal{D}} L(x, \lambda, v) \\
&= \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x) \right)
\end{aligned}
$$

  - a concave function of λ, v
  - can be $-\infty$ for some λ, v; this defines the domain of g

- Lagrange dual problem

$$
\begin{array}{ll}
\text{maximize} & g(\lambda, v) \\
\text{subject to} & \lambda \geq 0
\end{array}
$$

Karush-Kuhn-Tucker (KKT)

1. (primal feasibility) $f_i(x) \leq 0$ for $i = 1, \ldots, m$ and $h_i(x) = 0$ for $i = 1, \ldots, p$

2. (dual feasibility) $\lambda \geq 0$

3. $\lambda_i f_i(x) = 0$ for $i = 1, \ldots, m$

4. the gradient of the Lagrangian with respect to x vanishes:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} v_i \nabla h_i(x) = 0$$

- if the problem is convex optimization and its Langurange function is differentiable with respect to x, then x is optimal if and only if there exist λ, v such that $1 - 4$ are satisfied

Subgradient & Subdifferential

- g is a subgradient of a convex function f at $x \in \operatorname{dom} f$ if

$$f(y) \geq f(x) + g^T(y - x) \text{ for all } y \in \operatorname{dom} f$$

- subdifferential $\partial f(x)$ of f at x is the set of all subgradients:

$$\begin{aligned} \partial f(x) &= \{g | g^T(y - x) \leq f(y) - f(x), \forall y \in \operatorname{dom} f\} \\ &= \bigcap_{z \in \operatorname{dom} f} \{g \mid f(z) \geq f(x) + g^T(z - x)\} \end{aligned}$$

Subgradient & Subdifferential

- The optimality conditions can be adjusted for the non-differential convex functions via the subgradient concept.

- Karush-Kuhn-Tucker conditions:
  if strong duality holds, then $x^\star, \lambda^\star$ are primal, dual optimal if and only if

  (a) $x^\star$ is primal feasible

  (b) $\lambda^\star \geq 0$

  (c) $\lambda_i^\star f_i(x^\star) = 0$ for $i = 1, \ldots, m$

  (d) $x^\star$ is a minimizer of $L(x, \lambda^\star) = f_0(x) + \sum_{i=1}^m \lambda_i^\star f_i(x)$
  :

$$0 \in \partial f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \partial f_i(x^\star)$$

Lipschitz continuity & Strongly convexity

- A function h is called Lipschitz continuous with Lipschitz constant L, if

$$|h(x) - h(y)| \leq L\|x - y\|, \forall x, y \in \mathrm{dom}\, h.$$

- f is strongly convex with parameter $\mu > 0$ if $f(x) - \frac{\mu}{2}\|x\|_2^2$ is convex

Lipschitz continuity & Strongly convexity

Lipschitz-continuous with parameter $L > 0$

- quadratic upper bound $f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|x - y\|_2^2, x, y \in \mathrm{dom}\, f$
- If $\mathrm{dom}\, f = \mathbb{R}^n$ and f has a minimizer x^*, then

$$\frac{1}{2L}\|\nabla f(x)\|_2^2 \leq f(x) - f\left(x^*\right) \leq \frac{L}{2}\left\|x - x^*\right\|_2^2$$

Strongly convex with parameter $\mu > 0$

- First-order condition $f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|x - y\|_2^2, \forall x, y \in \mathrm{dom}\, f$
- Second-order condition $\nabla^2 f(x) \succeq \mu I, \forall x \in \mathrm{dom}\, f$
- If $\mathrm{dom}\, f = \mathbb{R}^n$, then f has a minimizer x^*, and

$$\frac{\mu}{2}\left\|x - x^*\right\|_2^2 \leq f(x) - f\left(x^*\right) \leq \frac{1}{2\mu}\|\nabla f(x)\|_2^2$$

Lipschitz continuity and subgradient

- A convex function f is Lipschitz continuous with constant $G > 0$:

$$|f(x) - f(y)| \leq G\|x - y\|_2, \forall x, y$$

this is equivalent to

$$\|g\|_2 \leq G, \forall g \in \partial f(x), \forall x$$

Gradient descent method

- Gradient Descent

$$x^{(k+1)} = x^{(k)} - t^{(k)} \nabla f\left(x^{(k)}\right)$$

- Stepsize Choices
  - exact line search: $t^{(k)} = \operatorname{argmin}_t f\left(x^{(k)} - t\nabla f\left(x^{(k)}\right)\right)$
  - fixed: $t^{(k)}$ constant
  - backtracking line search (most practical)

- Assume that f convex and differentiable, with $\operatorname{dom}(f) = \mathbb{R}^n$, and additionally that ∇f is Lipschitz continuous with constant $L > 0$ Gradient descent with fixed step size $t \leq 1/L$ satisfies

$$f\left(x^{(k)}\right) - f^* \leq \frac{\left\|x^{(0)} - x^*\right\|_2^2}{2tk}$$

and same result holds for backtracking, with t replaced by β/L

Subgradient method

- Subgradient method: choose $x^{(0)}$ and repeat

$$x^{(k)} = x^{(k-1)} - t_k g^{(k-1)}, k = 1, 2, \ldots$$

$g^{(k-1)}$ is any subgradient of f at $x^{(k-1)}$ step size rules:

  - fixed step: t_k constant
  - fixed length: $t_k \left\|g^{(k-1)}\right\|_2$ constant (i.e., $\left\|x^{(k)} - x^{(k-1)}\right\|_2$ constant)
  - diminishing: $t_k \to 0, \sum_{k=1}^{\infty} t_k = \infty$

- For a fixed step size t, subgradient method satisfies

$$\lim_{k\to\infty} f\left(x_{\text{best}}^{(k)}\right) \leq f^\star + G^2 t/2$$

Proximal gradient method.

- Consider:

$$\min f(x) = g(x) + h(x)$$

  - g convex, differentiable, $\operatorname{dom} g = \mathbb{R}^n$
  - h convex, but non-differentiable

- The proximal gradient method:

$$x^{(k)} = \underset{x}{\operatorname{argmin}} \frac{1}{2t_k} \left\| x - \left(x^{(k-1)} - t_k \nabla g\left(x^{(k-1)}\right)\right)\right\|^2 + h(x)$$

or equivalently, $x^{(k)} = \operatorname{prox}_{t_k h}\left(x^{(k-1)} - t_k \nabla g\left(x^{(k-1)}\right)\right)$,
$t_k > 0$ is step size, constant or determined by line search

Proximal mapping

- the proximal mapping (prox-operator) of a convex function h is defined as

$$\mathrm{prox}_h(x) = \underset{u}{\mathrm{argmin}} \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right)$$

- If $u = \mathrm{prox}_h(x), v = \mathrm{prox}_h(y)$, then

$$(u - v)^\top (x - y) \geq \|u - v\|_2^2$$

  prox $_h$ is firmly nonexpansive

- By Cauchy-Schwarz inequality

$$\|\mathrm{prox}_h(x) - \mathrm{prox}_h(y)\|_2 \leq \|x - y\|_2$$

  prox_h is nonexpansive, or Lipschitz continuous with constant 1

Two famous proximal mappings

- Euclidean norm: $f(x) = \|x\|_2$

$$\mathrm{prox}_{tf}(x) = \begin{cases} (1 - t/\|x\|_2)\, x \text{ if } \|x\|_2 \geq t \\ 0 \text{ otherwise} \end{cases}$$

- $f(x) = \|x\|_1$:

$$\mathrm{prox}_{tf}(x)_i = \begin{cases} x_i - t & \text{if } x_i \geq t \\ 0 & \text{if } |x_i| \leq t \\ x_i + t & \text{if } x_i \leq -t \end{cases}$$

This is also called "soft-threshold" (shrinkage) operation: $S_t(x) := \mathrm{prox}_{tf}(x)$.

FISTA

- Consider: $\min f(x) = g(x) + h(x)$
  - g convex, differentiable, with dom $g = \mathbb{R}^n$
  - h closed, convex, with inexpensive prox th operator
- Algorithm: choose any $x^{(0)} = x^{(-1)}$; for $k \geq 1$, repeat

$$
\begin{aligned}
y &= x^{(k-1)} + \tfrac{k-1}{k+1}\left(x^{(k-1)} - x^{k-2}\right) \\
x^{(k)} &= \mathrm{prox}_{th}(y - t\nabla g(y))
\end{aligned}
$$

  - step size t_k fixed or by line search
  - acronym stands for "Fast Iterative Shrinkage-Thresholding Algorithm"

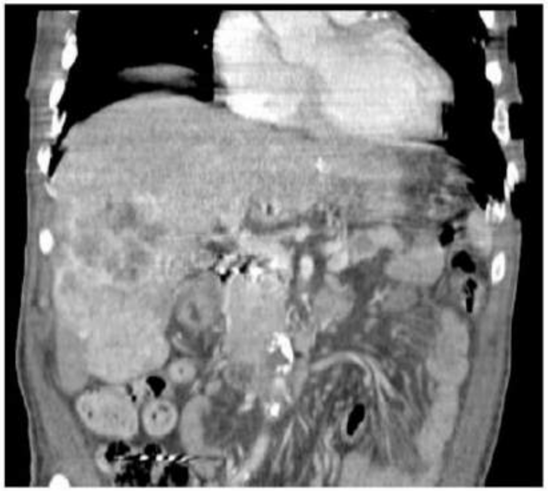
Example

Optimization-based Iterative Reconstruction Methods

Iterative reconstruction is based on optimization problems and algorithms

Discrete imaging model:

$$Ax = b$$



Typical CT images:

- Regions of homogeneous tissue.
- Separated by sharp boundaries.

Reconstruction by regularization:

$$x^\star = \operatorname*{argmin}_x \underbrace{\mathcal{D}(Ax, b)}_{\text{data fidelity}} + \lambda \cdot \underbrace{\mathcal{R}(x)}_{\text{regularizer}}$$

Most basic optimization problem (no regularization):

$$u^\star = \arg\min_u \|Au - b\|_2^2 = \sum_i \left((Au)_i - b_i\right)^2$$

Regularized reconstruction – example Tikhonov

Data fidelity

Regulariser

$$u^\star = \arg\min_u \left\{ \|Au - b\|_2^2 + \alpha^2 \|Lu\|_2^2 \right\}$$

Minimiser:
Solution image

Unknown
image TBD

Regularisation
parameter

- Balance between fitting data and penalizing ”large values of *Lu*”
- Different choices of *L*:
  - Identity operator – make pixel values small
  - Finite difference gradient operator – make neighbor pixels similar
- No one best regulariser - different image types need different regularisers!

Regularized reconstruction – example Tikhonov

$$u^\star = \arg\min_u \left\{ \|Au - b\|_2^2 + \alpha^2 \|Lu\|_2^2 \right\}$$

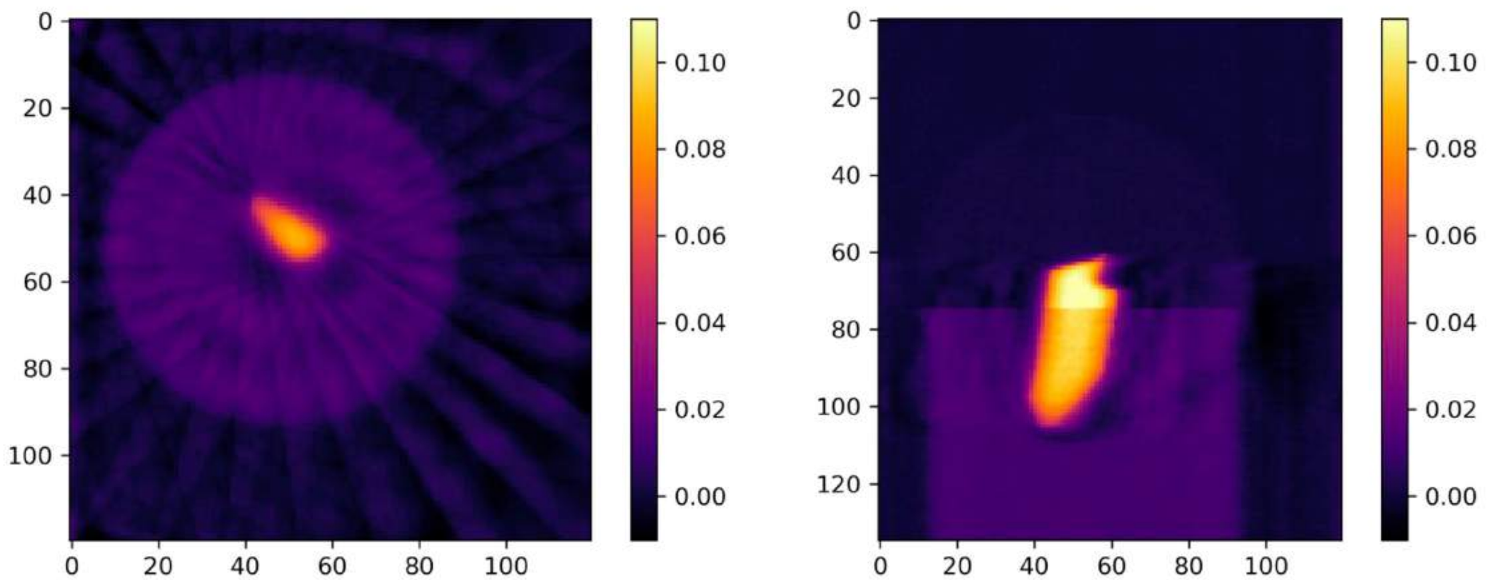
Data fidelity

Regulariser

Minimiser:
Solution image

Unknown
image TBD

Regularisation
parameter



How to solve Tikhonov problem

$$\min_u \|Au - b\|^2 + \alpha^2 \|Lu\|^2$$

$$\downarrow$$

$$\min_u \left\| \begin{pmatrix} A \\ \alpha L \end{pmatrix} u - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|^2$$

$$\downarrow$$

$$\min_u \|\tilde{A}u - \tilde{b}\|^2$$ **with** $$\tilde{A} = \begin{pmatrix} A \\ \alpha L \end{pmatrix} \qquad \tilde{b} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

$$\downarrow$$

CGLS with $\tilde{A}$ and $\tilde{b}$

Gradient descent algorithm (when differentiable)

More general optimization problem

$$\arg\min_u f(u)$$

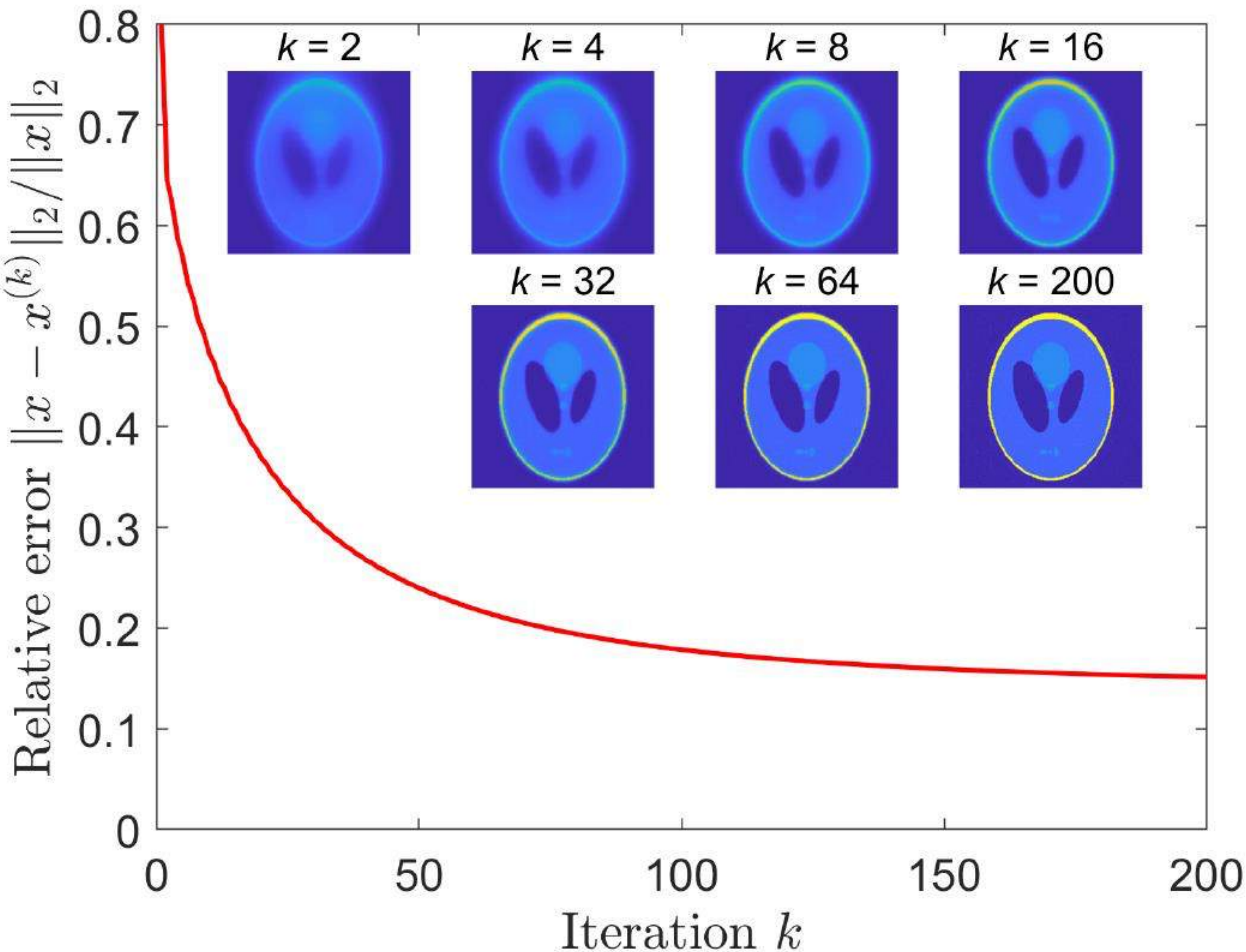
Gradient (*f* must be differentiable):

$$\nabla f(u) = \begin{pmatrix} \frac{\partial f(u)}{\partial u_1} \\ \frac{\partial f(u)}{\partial u_2} \\ \vdots \\ \frac{\partial f(u)}{\partial u_n} \end{pmatrix}$$

Gradient descent algorithm:

$$u^{(k+1)} = u^{(k)} - t_k \nabla f(u^{(k)}), \quad k = 0, 1, 2, ...$$

Gradient descent algorithm example: least squares minimization (Shepp-Logan)



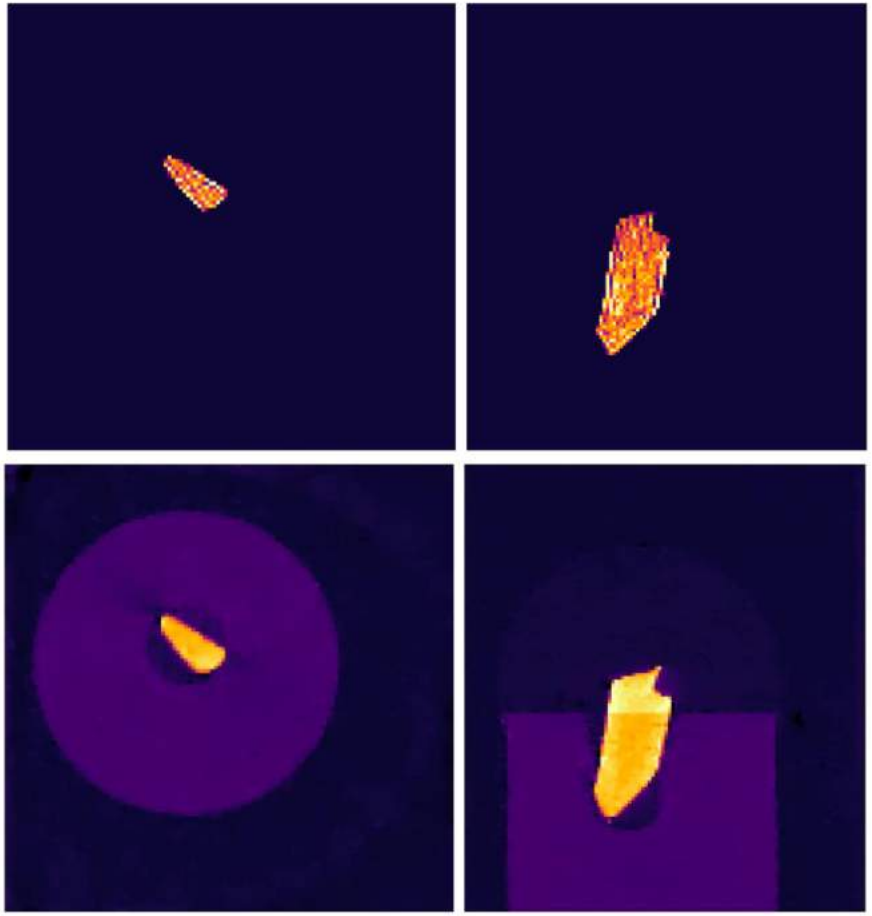
Sparsity and total variation regularization

L1-norm regularization:

$$\|u\|_1 = \sum_j |u_j|$$

Total variation regularization:

$$\sum_j \|D_j u\|_2$$



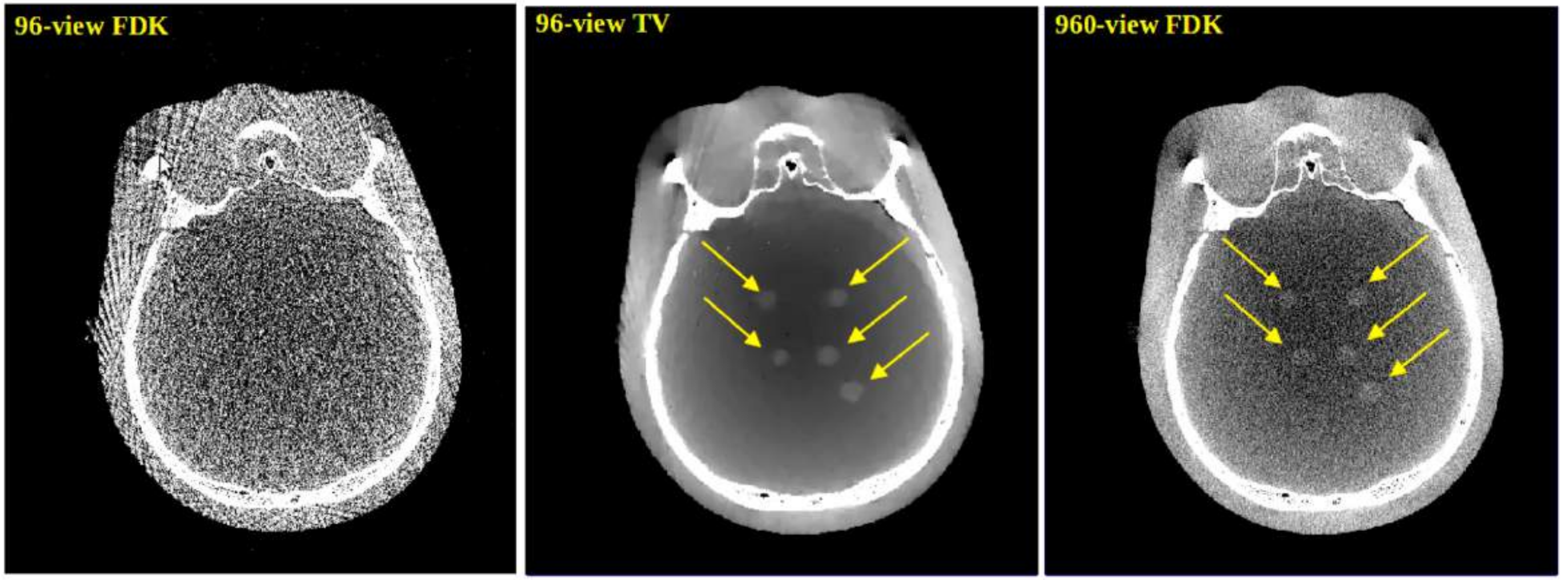
TV reconstruction example, physical head phantom, cone-beam X-ray CT

Total variation: Homogeneous regions with sharp boundaries.

$$x^\star = \underset{x}{\operatorname{argmin}} \left\{ \|Ax - b\|_2^2 + \alpha \|x\|_{\mathsf{TV}} \right\}$$

$$\|x\|_{\mathsf{TV}} = \sum_j \|D_j x\|_2, \quad D_j \text{ finite diff. gradient at voxel } j.$$

TV is an example of **sparsity-regularized reconstruction.**

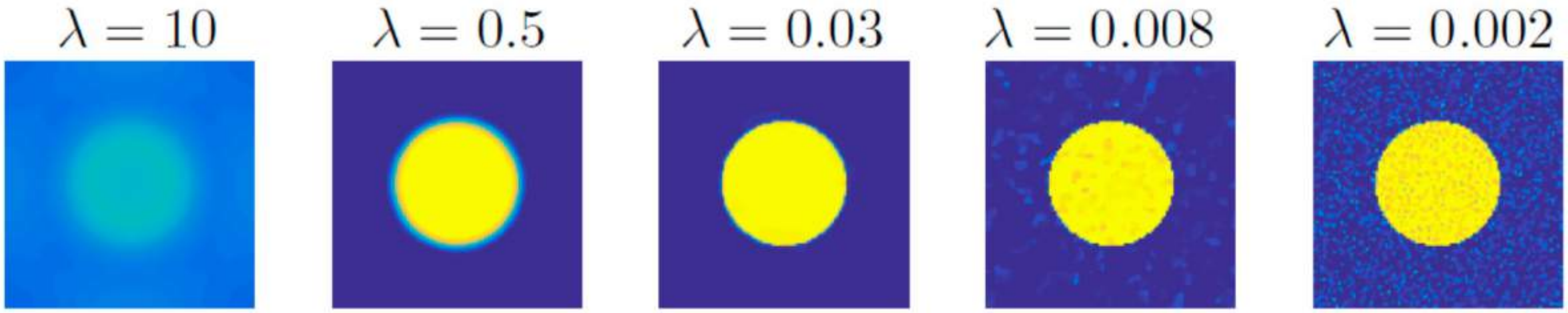


[Bian et al. 2010, Phys. Med. Biol. **55**, 6575–6599]. Courtesy: X. Pan, U. Chicago.

Effect of regularization parameter

Total variation regularization:

$$\min_u \; \|Au - b\|_2^2 + \lambda \cdot \mathrm{TV}(u)$$

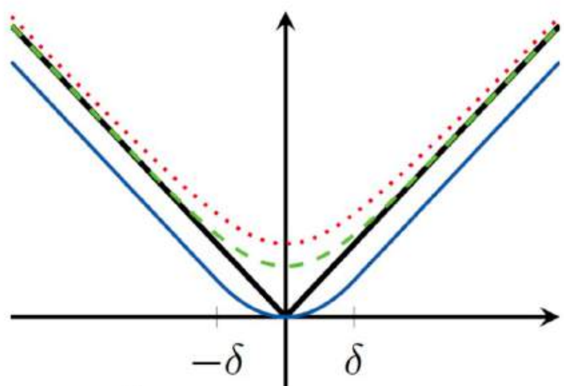


- Large λ: Almost only effect of regularizer. TV $\to$ Constant.
- Small λ: Almost just least-squares solution.
- Best trade-off?

How to solve TV optimization problem?

- TV is NOT smooth, i.e., NOT differentiable – due to coupling of *x* and *y* derivatives under a square-root:

$$\mathrm{TV}(u) = \|Du\|_{2,1} = \sum_{i,j} \left(\sqrt{(D_y u)^2 + (D_x u)^2} \right)_{i,j}$$



- We cannot use gradient descent etc.

- One approach is to *smooth the problem:*

$$\mathrm{TV}_\delta(u) = \sum_{i,j} \left(\sqrt{(D_y u)^2 + (D_x u)^2 + \delta^2} \right)_{i,j}$$

FISTA: Fast Iterative Shrinkage Thresholding Algorithm

$$\boldsymbol{x}^{\star} = \arg\min_{\boldsymbol{x}} \{\mathcal{F}(\boldsymbol{x}) + \beta\,\mathcal{G}(\boldsymbol{x})\}$$

Input: $\boldsymbol{b}$, $\boldsymbol{x}^{[0]}$, β, S, L

Output: $\boldsymbol{x}^{[S]}$

$\boldsymbol{y}^{[1]} = \boldsymbol{x}^{[0]}$, $t^{[1]} = 1$

for all $s = 1, \ldots, S$ **do**

1: $\boldsymbol{u}^{[s]} = \boldsymbol{y}^{[s]} - L^{-1}\nabla\mathcal{F}(\boldsymbol{y}^{[s]})$

2: $\boldsymbol{x}^{[s]} = \mathrm{prox}_{\beta/L}[\mathcal{G}](\boldsymbol{u}^{[s]})$

3: $t^{[s+1]} = \left(1 + \sqrt{1 + 4(t^{[s]})^2}\right)/2$

4: $\boldsymbol{y}^{[s+1]} = \boldsymbol{x}^{[s]} + (t^{[s]} - 1)/t^{[s+1]} \cdot (\boldsymbol{x}^{[s]} - \boldsymbol{x}^{[s-1]})$

end for

Proximal mapping

- Defined through a minimisation problem

$$\mathrm{prox}_{\beta/L}[\mathcal{G}](\boldsymbol{v}) = \arg\min_{\boldsymbol{u}} \left\{ \frac{\beta}{L}\mathcal{G}(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{v}\|_2^2 \right\}$$

- FISTA is useful when proximal mapping above has simple closed-form solution or can be efficiently computed numerically.

- Simple closed-form examples for $\mathcal{G}$:
  - Constraint to convex set: Proximal mapping is projection.
  - L1-norm: Proximal mapping is soft-thresholding.

- Proximal mapping for TV can be computed numerically.

Singular Value Decomposition

A brief introduction

SVD

Singular Value Decomposition (SVD)

Let g be an image (can be general $m \times n$ image).

Assume $g^T g$ is of rank r. Then g can be written as

$$g = U \Lambda^{1/2} V^T$$

where $U \in M_{m \times m}$ and $V \in M_{n \times n}$ are orthogonal matrices ($UU^T = U^T U = I$ and $VV^T = V^T V = I$) and $\Lambda^{1/2}$ is a diagonal $n \times n$ matrix.

An image can be decomposed as:

$$g = U \Lambda^{1/2} V^T = \sum_{i=1}^{r} \lambda_i^{1/2} \vec{u}_i \vec{v}_i^T .$$

Eigen-image

SVD

- Let $B \in M_{n\times n}$ be a real symmetric matrix. Then, there exist n orthonormal eigenvectors $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ with corresponding eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ such that

$$B = \begin{bmatrix} | & & | \\ \vec{v}_1 & \cdots & \vec{v}_n \\ | & & | \end{bmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} \begin{bmatrix} - & \vec{v}_1^T & - \\ - & \vec{v}_2^T & - \\ & \vdots & \\ - & \vec{v}_n^T & - \end{bmatrix}.$$

  Now, note that $gg^T \in M_{m\times m}$ and $g^T g \in M_{n\times n}$ are symmetric. Thus, there exist n pairwise orthonormal eigenvectors $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ of $g^T g$.

- For any k with $0 \leq k \leq r$, we define

$$g_k = \sum_{j=1}^{k} \sigma_j \vec{u}_j \vec{v}_j^T$$

  where g_k is called a rank- k approximation of g.

Rank-k approximation

- The Frobenius norm (F-norm) given by

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2},$$

where a_{ij} is the i-th row, j-th column entry of A. Let $\mathbf{a_i}$ be the j-th column of A. We have

$$\|A\|_F = \sqrt{\sum_{j=1}^{n} \|a_j\|_2^2} = \sqrt{\mathrm{tr}\,(A^*A)} = \sqrt{\mathrm{tr}\,(AA^*)},$$

where $\mathrm{tr}(\cdot)$ is the trace of the matrix in the argument.

- Let $f = \sum_{j=1}^{r} \sigma_j \vec{u}_j \vec{v}_j^T$ be the SVD of an $M \times N$ image f. For any k with $k < r$ and $f_k = \sum_{j=1}^{k} \sigma_j \vec{u}_j \vec{v}_j^T$, we have

$$\|f - f_k\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2.$$

Example of SVD decomposition of an image

Example 2.1: SVD decomposition of an image

Show the different stages of the SVD of the following image:

$$g = \begin{pmatrix} 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 100 & 100 & 100 & 255 & 255 \\ 255 & 255 & 100 & 150 & 150 & 150 & 100 & 255 \\ 255 & 255 & 100 & 150 & 200 & 150 & 100 & 255 \\ 255 & 255 & 100 & 150 & 150 & 150 & 100 & 255 \\ 255 & 255 & 255 & 100 & 100 & 100 & 255 & 255 \\ 255 & 255 & 255 & 255 & 50 & 255 & 255 & 255 \\ 50 & 50 & 50 & 50 & 255 & 255 & 255 & 255 \end{pmatrix}$$

Example of SVD decomposition of an image

Example 2.1: SVD decomposition of an image

The image looks like:

Example of SVD decomposition of an image

Example 2.1: SVD decomposition of an image

Consider the eigenvalues of:

$$gg^T = \begin{pmatrix} 520200 & 401625 & 360825 & 373575 & 360825 & 401625 & 467925 & 311100 \\ 401625 & 355125 & 291075 & 296075 & 291075 & 355125 & 381125 & 224300 \\ 360825 & 291075 & 282575 & 290075 & 282575 & 291075 & 330075 & 205025 \\ 373575 & 296075 & 290075 & 300075 & 290075 & 296075 & 332575 & 217775 \\ 360825 & 291075 & 282575 & 290075 & 282575 & 291075 & 330075 & 205025 \\ 401625 & 355125 & 291075 & 296075 & 291075 & 355125 & 381125 & 224300 \\ 467925 & 381125 & 330075 & 332575 & 330075 & 381125 & 457675 & 258825 \\ 311100 & 224300 & 205025 & 217775 & 205025 & 224300 & 258825 & 270100 \end{pmatrix}$$

Eigenvalues are:

2593416.500 111621.508 71738.313 34790.875
11882.712 0.009 0.001 0.000

We take first 5 eigenvalues!!

Example of SVD decomposition of an image

Example 2.1: SVD decomposition of an image

The corresponding first five eigenvectors are:

$$\begin{pmatrix} 0.441 & -0.167 & -0.080 & -0.388 & 0.764 \\ 0.359 & 0.252 & -0.328 & 0.446 & 0.040 \\ 0.321 & 0.086 & 0.440 & 0.034 & -0.201 \\ 0.329 & 0.003 & 0.503 & 0.093 & 0.107 \\ 0.321 & 0.086 & 0.440 & 0.035 & -0.202 \\ 0.359 & 0.252 & -0.328 & 0.446 & 0.040 \\ 0.407 & 0.173 & -0.341 & -0.630 & -0.504 \\ 0.261 & -0.895 & -0.150 & 0.209 & -0.256 \end{pmatrix}$$

$$\mathbf{u}_1 \quad \mathbf{u}_2 \quad \mathbf{u}_3 \quad \mathbf{u}_4 \quad \mathbf{u}_5$$

Example of SVD decomposition of an image

Example 2.1: SVD decomposition of an image
The corresponding first five eigenvectors are:

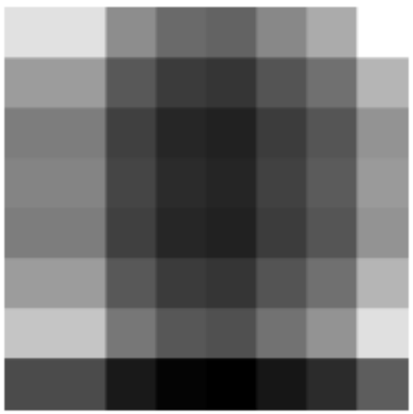
$\mathbf{v}_i$ can be computed by $g^T \mathbf{u}_i$

$$\begin{pmatrix} 0.410 & 0.389 & 0.264 & 0.106 & -0.012 \\ 0.410 & 0.389 & 0.264 & 0.106 & -0.012 \\ 0.316 & 0.308 & -0.537 & -0.029 & 0.408 \\ 0.277 & 0.100 & 0.101 & -0.727 & 0.158 \\ 0.269 & -0.555 & 0.341 & 0.220 & 0.675 \\ 0.311 & -0.449 & -0.014 & -0.497 & -0.323 \\ 0.349 & -0.241 & -0.651 & 0.200 & -0.074 \\ 0.443 & -0.160 & 0.149 & 0.336 & -0.493 \end{pmatrix}$$

Example of SVD decomposition of an image

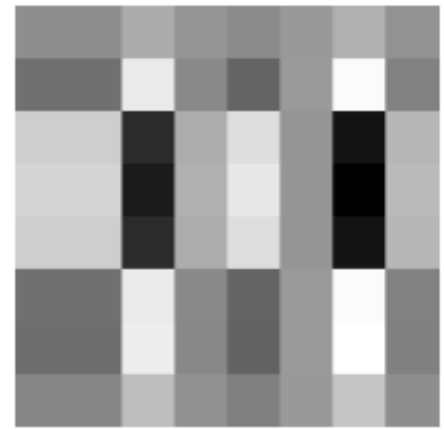
Example 2.1: SVD decomposition of an image

Compute the five eigenimages:



i=1

i=2



i=3

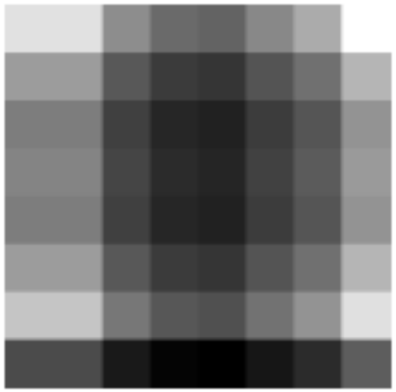
i=4

i=5

Example of SVD decompsition of an image

Example 2.1: SVD decomposition of an image

Compute the five eigen decomposition



a) k=1

b) k=2

c) k=3

d) k=4

e) k=5

Original

Example of SVD decomposition of an image

Example 2.1: SVD decomposition of an image

Error in the reconstruction:

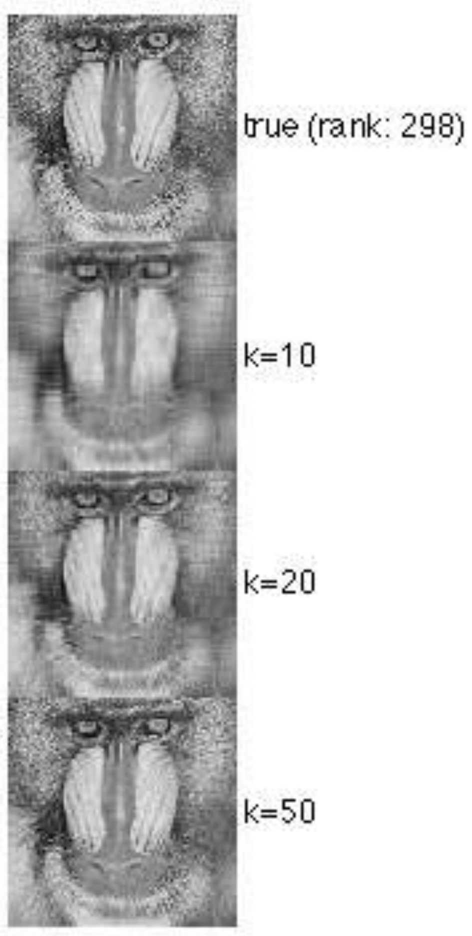
$$\sum_{all\ pixels} (reconstructed\ pixel - original\ pixel)^2$$

$$\begin{array}{lll}
Square\ error\ for\ image\ a: & 230033.32 & (\lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 = 230033.41) \\
Square\ error\ for\ image\ b: & 118412.02 & (\lambda_3 + \lambda_4 + \lambda_5 = 118411.90) \\
Square\ error\ for\ image\ c: & 46673.53 & (\lambda_4 + \lambda_5 = 46673.59) \\
Square\ error\ for\ image\ d: & 11882.65 & (\lambda_5 = 11882.71) \\
Square\ error\ for\ image\ e: & 0 &
\end{array}$$

Small error, 0.01!

Example of SVD decomposition of an image

SVD decomposition of an image



Work well for images with patterns

Work well for simple images

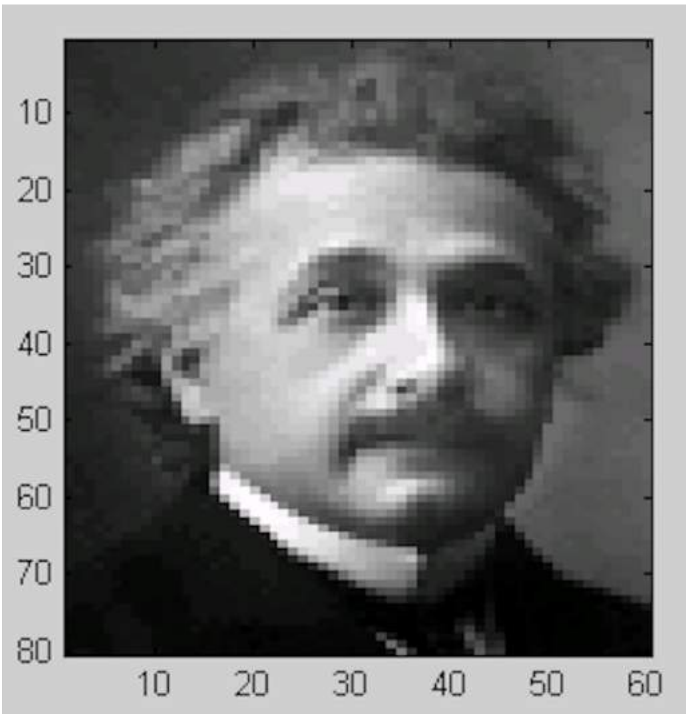
Example of SVD decompsition of an image

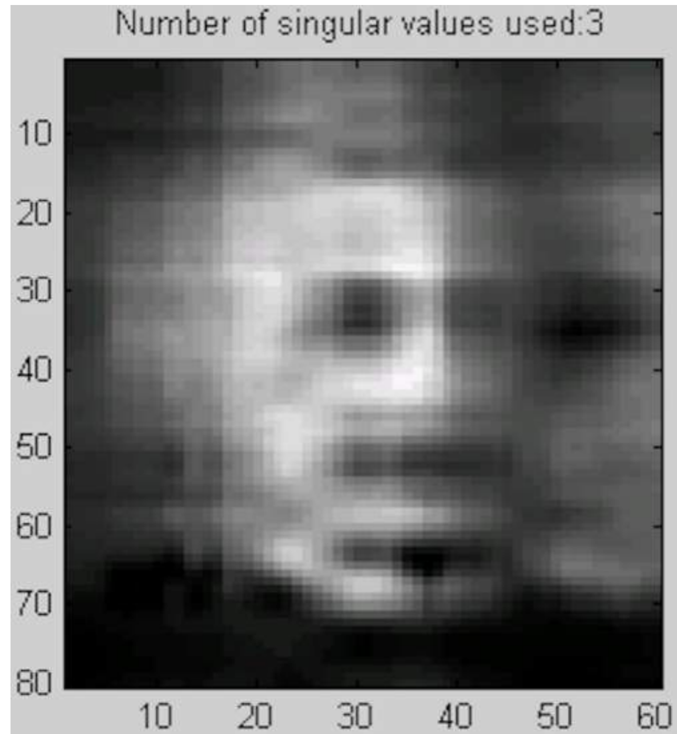
SVD decomposition of an image

Low rank approximation can capture key (big) object

Example of SVD decompsition of an image

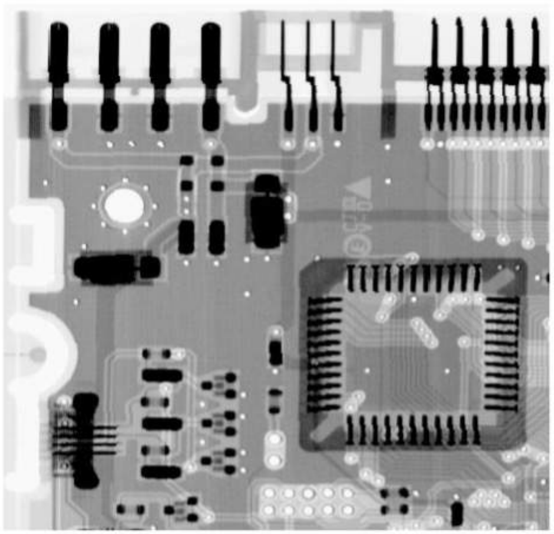
SVD decomposition of an image



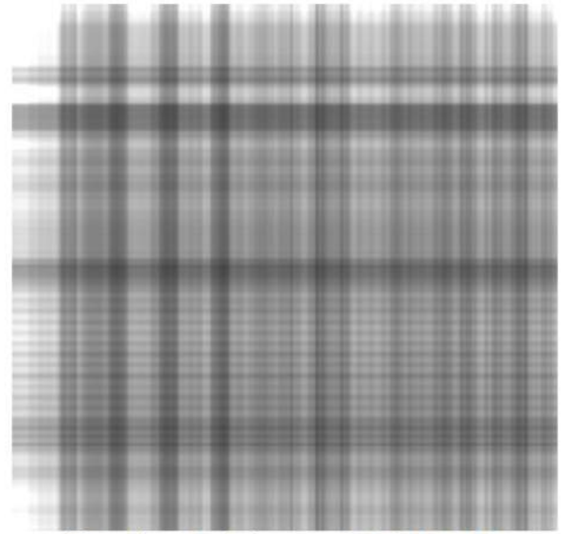


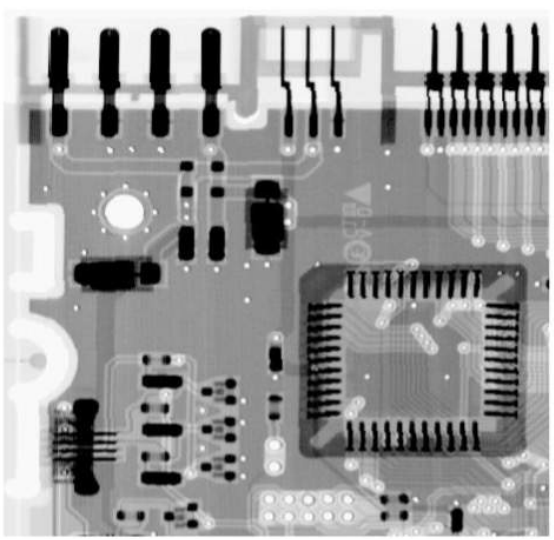
Low rank approximation can capture key (big) object

Another example of SVD decomposition

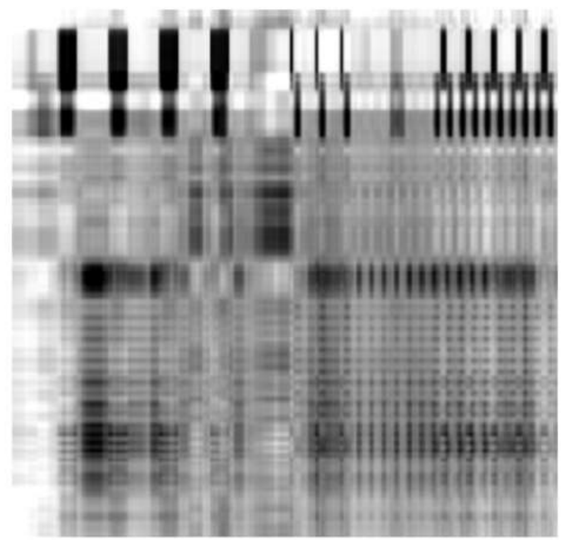


Rank=1, Error=15192.4654

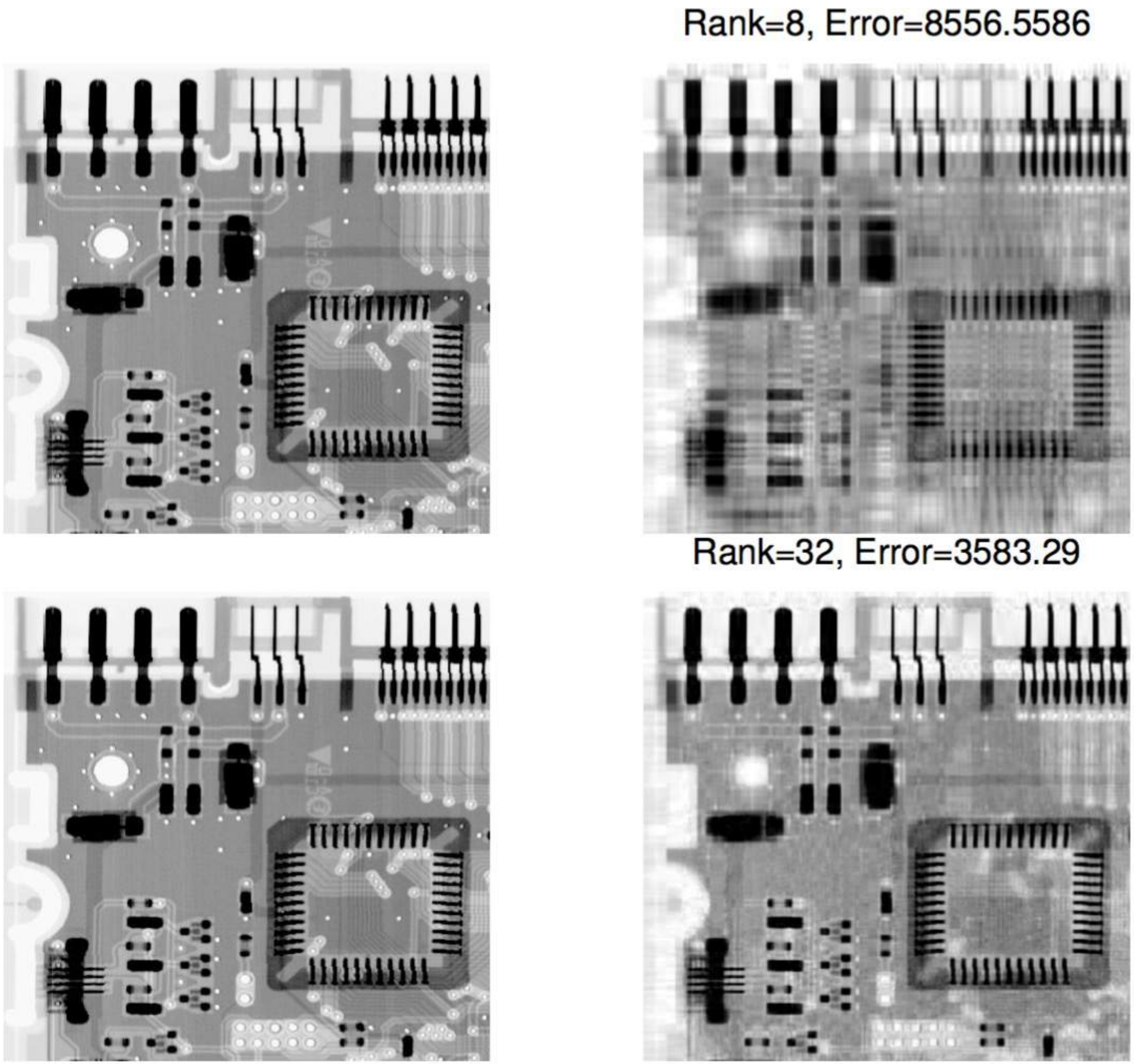




Rank=4, Error=11178.36



Another example of SVD decomposition



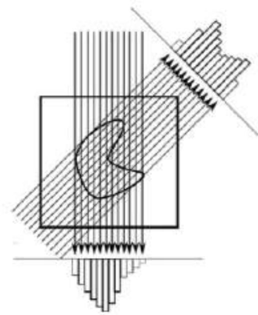
Inverse problems

Example from CT reconstruction

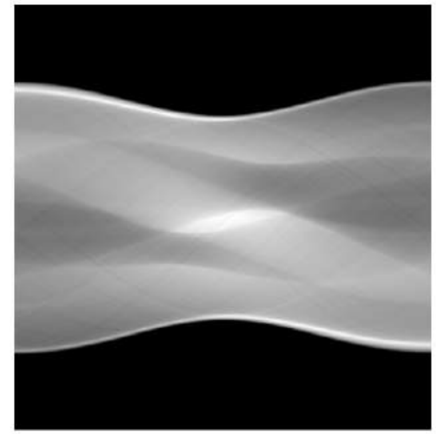
CT reconstruction

Object

$\bar{\boldsymbol{x}}$



$\Longleftarrow$

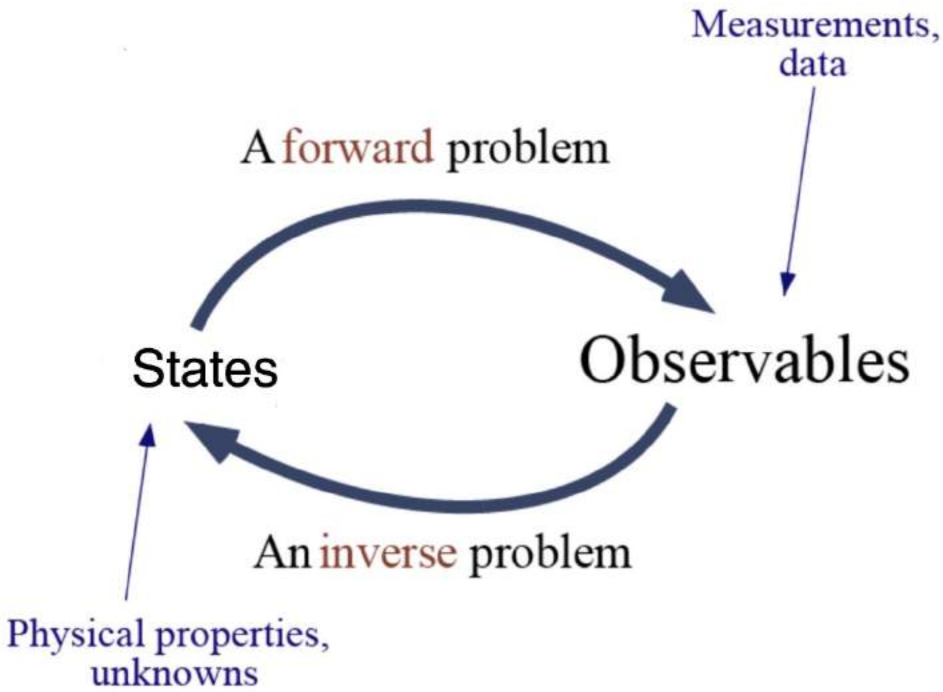


Measurements

$\boldsymbol{b} = \mathcal{N}(\boldsymbol{A}\bar{\boldsymbol{x}})$

- **Our Problem:** Reconstruct $\bar{\boldsymbol{x}}$ from $\boldsymbol{b}$ with given $\boldsymbol{A}$.
- It is a highly **ill-posed** inverse problem.

Inverse problems



Why are inverse problems difficult?

- Forward models are not explicitly invertible
- Errors in the measurements (and also in the forward model) can lead to errors in the solution

Hadamard condition

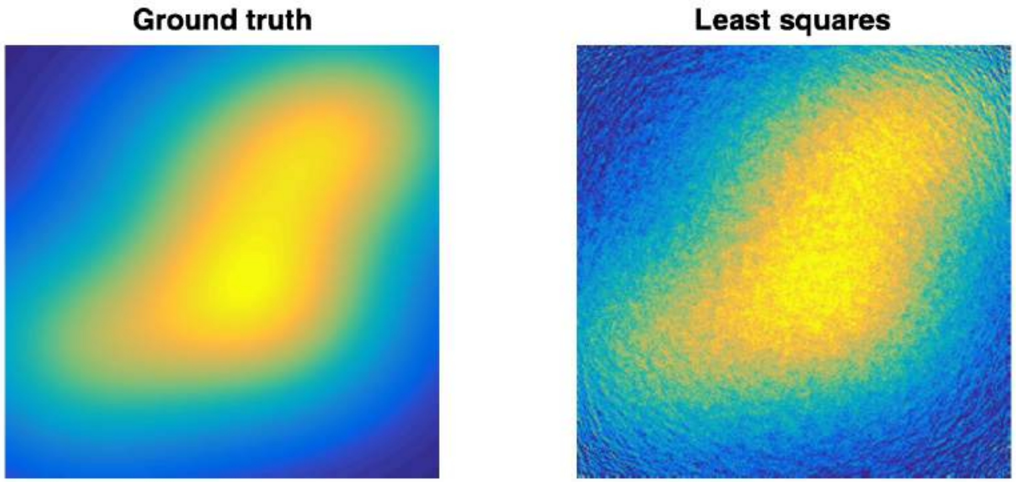
A problem is called well-posed if

- there exists a solution to the problem (existence),
- there is at most one solution to the problem (uniqueness),
- the solution depends continuously on the measurement (stability).

Otherwise the problem is called ill-posed.

Example

- If too many measurements and no consistence, the solution of $\boldsymbol{Ax} = \boldsymbol{b}$ does not exist.
- If no enough measurements, the solution of $\boldsymbol{Ax} = \boldsymbol{b}$ is not unique.
- Even we have a unique least-squares solution, it can be not good enough due to lack of the stability.



More questions need be considered

- Why are inverse problems difficult?

  $\Longleftarrow$ It's often ILL-POSED!

- How can we solve an ill-posed inverse problem?
  - Does the measurements actually contain the information we want?
  - Which solution do we want?
  - The measurement may not be enough by itself to completely determine the unknown. What other prior information of the "unknown" do we have?

  $\Longleftarrow$ We can use REGULARIZATION techniques!

More questions need be considered

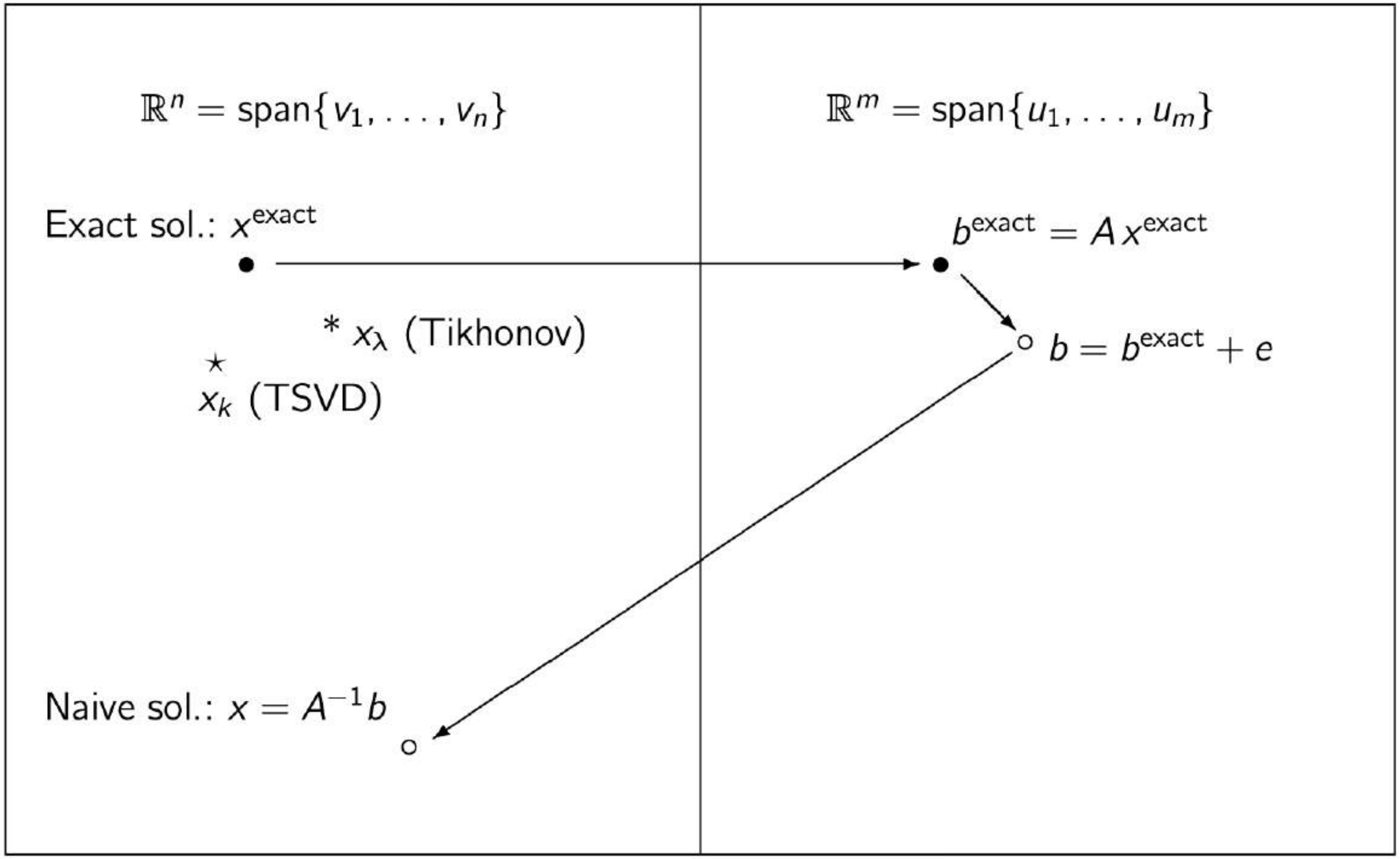
- Why are inverse problems difficult?

  $\Longleftarrow$ It's often ILL-POSED!

- How can we solve an ill-posed inverse problem?
  - Does the measurements actually contain the information we want?
  - Which solution do we want?
  - The measurement may not be enough by itself to completely determine the unknown. What other prior information of the "unknown" do we have?

  $\Longleftarrow$ We can use REGULARIZATION techniques!

Illustration of the need for regularization



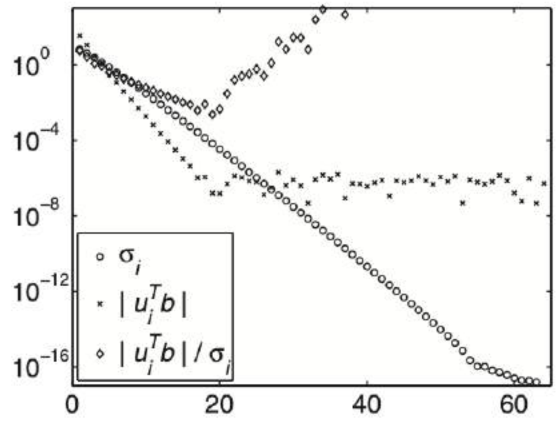
Truncated SVD

Considering the linear inverse problem

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \qquad \text{with } \boldsymbol{b} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}.$$

Based on the SVD of $\boldsymbol{A}$, the "naive" solution is given by

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} = \sum_{i=1}^{I} \frac{\boldsymbol{u}_i^\top \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i = \bar{\boldsymbol{x}} + \sum_{i=1}^{I} \frac{\boldsymbol{u}_i^\top \boldsymbol{e}}{\sigma_i} \boldsymbol{v}_i$$



Truncated SVD

The solution of Truncated SVD is

$$\boldsymbol{x}_{\mathrm{TSVD}} = V\Sigma_k^{\dagger}U^{\top}\boldsymbol{b} = \sum_{i=1}^{k}\frac{\boldsymbol{u}_i^{\top}\boldsymbol{b}}{\sigma_i}\boldsymbol{v}_i$$

with $\Sigma_k^{\dagger} = \mathrm{diag}(\sigma_1^{-1}, \cdots, \sigma_k^{-1}, 0, \cdots, 0)$.

- **Regularization parameter:**
  k, i.e, the number of SVD components.
- **Advantages:**
  - Intuitive
  - Easy to compute, *if we have the SVD*
- **Drawback:**
  - For large-scale problem, it is infeasible to compute the SVD

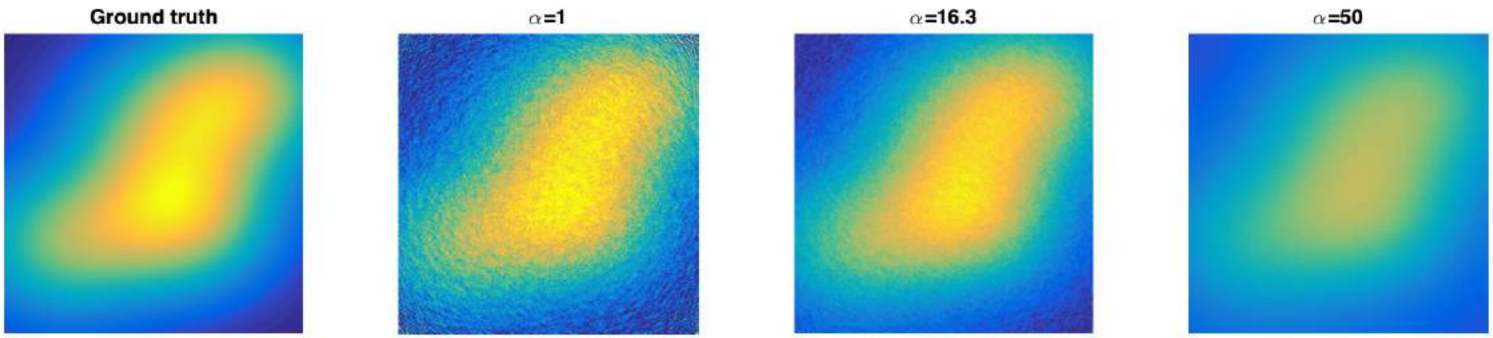
Tikhonov regularization

Idea: If we control the norm of the solution, then we should be able to suppress most of the large noise components.

The Tikhonov solution $\boldsymbol{x}_{\text{Tik}}$ is defined as the solution to

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \alpha\frac{1}{2}\|\boldsymbol{x}\|_2^2$$

- **Regularization parameter:** α
- α large: strong regularity, over smoothing.
- α small: good fitting



The solution of Tikhonov regularization

Reformulate as a linear least squares problem

$$\min_{\boldsymbol{x}} \frac{1}{2} \left\| \begin{pmatrix} \boldsymbol{A} \\ \sqrt{\alpha}\boldsymbol{I} \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} \boldsymbol{b} \\ 0 \end{pmatrix} \right\|_2^2$$

The normal equation is

$$(\boldsymbol{A}^T\boldsymbol{A} + \alpha\boldsymbol{I})\,\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b},$$

The solution is

$$\begin{aligned} \boldsymbol{x}_{\text{Tik}} &= (\boldsymbol{A}^T\boldsymbol{A} + \alpha\boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{b} \\ &= V(\Sigma^2 + \alpha\boldsymbol{I})^{-1}\Sigma^\top U^\top \boldsymbol{b} \\ &= \sum_{i=1}^{n} \frac{\sigma_i(\boldsymbol{u}_i^\top\boldsymbol{b})}{\sigma_i^2 + \alpha}\boldsymbol{v}_i \end{aligned}$$

Compare with TSVD

- The solution of TSVD is

$$\boldsymbol{x}_{\text{TSVD}} = \sum_{i=1}^{k} \frac{\boldsymbol{u}_i^\top \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i = \sum_{i=1}^{n} \varphi_i^{\text{TSVD}} \frac{\boldsymbol{u}_i^\top \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i$$

with $\varphi_i^{\text{TSVD}} = \begin{cases} 1, & 1 \le i \le k, \\ 0, & k < i \le n. \end{cases}$

- The solution of Tikhonov regularization is

$$\boldsymbol{x}_{\text{Tik}} = \sum_{i=1}^{n} \frac{\sigma_i(\boldsymbol{u}_i^\top \boldsymbol{b})}{\sigma_i^2 + \alpha} \boldsymbol{v}_i = \sum_{i=1}^{n} \varphi_i^{\text{Tik}} \frac{\boldsymbol{u}_i^\top \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i$$

with $\varphi_i^{\text{Tik}} = \frac{\sigma_i^2}{\sigma_i^2+\alpha} \approx \begin{cases} 1, & \sigma_i \gg \sqrt{\alpha}\ , \\ \frac{\sigma_i^2}{\alpha}, & \sigma_i \ll \sqrt{\alpha}\ . \end{cases}$

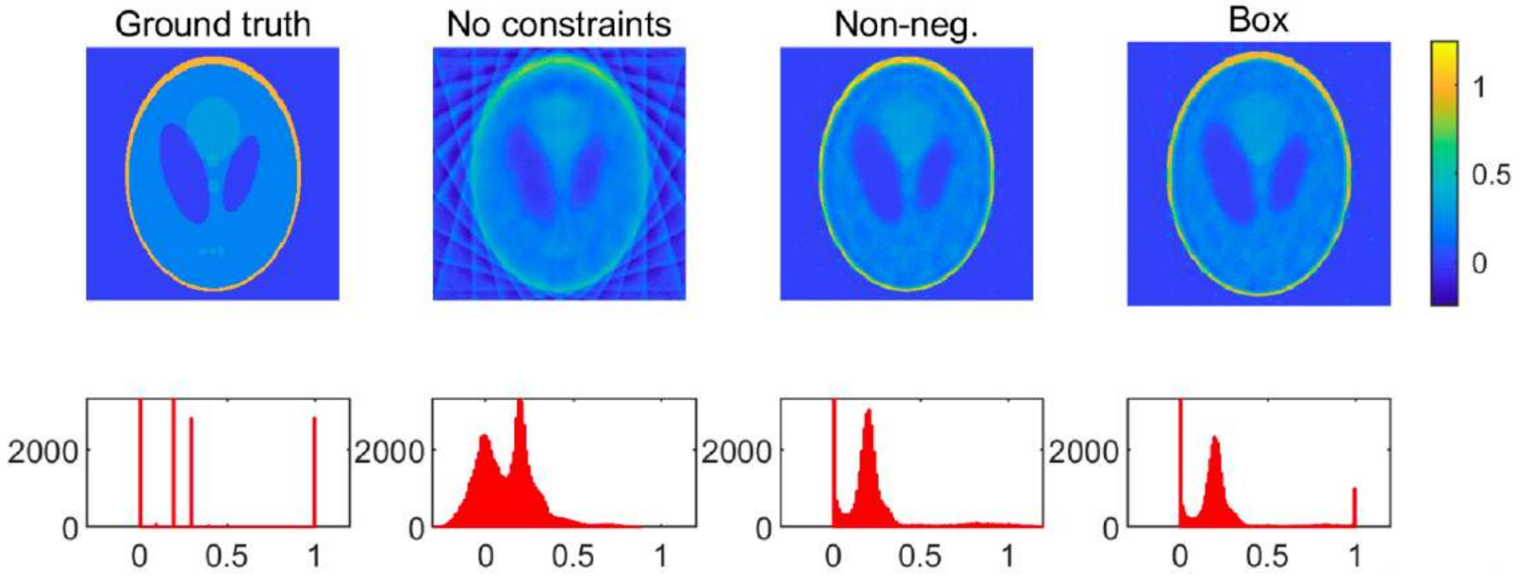
Non-negativity and box constraints

- **Non-negativity constrained Tikhonov problem:**

$$\min_{\boldsymbol{x} \geq 0} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \alpha \frac{1}{2}\|\boldsymbol{x}\|_2^2$$

- **Box constrained Tikhonov problem:**

$$\min_{\boldsymbol{x} \in [a,b]^n} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \alpha \frac{1}{2}\|\boldsymbol{x}\|_2^2$$



Gaussian noise

$$\boldsymbol{b} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}$$

where $\boldsymbol{e}$ denotes additive white Gaussian noise with zero mean and the covariance $\eta^2 \boldsymbol{I}_m$.

- All elements in $\boldsymbol{e}$ are independent.
- $\boldsymbol{e}$ is independent on $\bar{\boldsymbol{x}}$.
- Each element $\boldsymbol{e}_i$ can be seen as a Gaussian random variable with mean 0 and variance η^2.

Maximum likelihood estimate

$$\boldsymbol{b} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}$$

where $\boldsymbol{e}$ denotes additive white Gaussian noise with zero mean and the covariance $\eta^2 \boldsymbol{I}_m$.

- The probability density for observing $\boldsymbol{b}$ given $\boldsymbol{x}$ is

$$\pi(\boldsymbol{b} \,|\, \boldsymbol{x}) = \pi(\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi}\eta)^m} \exp\left(-\frac{\|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}\|_2^2}{2\eta^2}\right), \quad (1)$$

  which is called the *likelihood* of $\boldsymbol{x}$.
- **Maximum likelihood (ML) estimate** can be obtained by solving:

$$\max_{\boldsymbol{x}} \pi(\boldsymbol{b} \,|\, \boldsymbol{x}) \quad \Longleftrightarrow \quad \min_{\boldsymbol{x}} -\log\left(\pi(\boldsymbol{b} \,|\, \boldsymbol{x})\right).$$

- With the likelihood of $\boldsymbol{x}$ given in (1), we obtain the ML estimation problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}\|_2^2 \ .$$

MAP esitmate

To obtain a stable solution, we can incorporate prior information on $\bar{\boldsymbol{x}}$ by applying Bayes formula:

$$\pi(\boldsymbol{x} \mid \boldsymbol{b}) = \frac{\pi(\boldsymbol{b} \mid \boldsymbol{x})\, \pi_{\text{prior}}(\boldsymbol{x})}{\pi(\boldsymbol{b})} \; .$$

- $\pi(\boldsymbol{x} \mid \boldsymbol{b})$ is the posterior.
- $\pi(\boldsymbol{b} \mid \boldsymbol{x})$ is the likelihood.
- $\pi_{\text{prior}}(\boldsymbol{x})$ is the prior probability density of $\boldsymbol{x}$.
- $\pi(\boldsymbol{b})$ is the prior probability density of $\boldsymbol{b}$.

Maximum a posteriori (MAP) estimate can be obtained by solving:

$$\begin{aligned} \max_{\boldsymbol{x}} \pi(\boldsymbol{x} \mid \boldsymbol{b}) &\iff \max_{\boldsymbol{x}} \frac{\pi(\boldsymbol{b} \mid \boldsymbol{x})\, \pi_{\text{prior}}(\boldsymbol{x})}{\pi(\boldsymbol{b})}, \\ &\iff \min_{\boldsymbol{x}} -\log(\pi(\boldsymbol{b} \mid \boldsymbol{x})) - \log(\pi_{\text{prior}}(\boldsymbol{x})), \end{aligned}$$

Example

If we have

- the likelihood: $\pi(\boldsymbol{b} \mid \boldsymbol{x}) = \frac{1}{(\sqrt{2\pi}\eta)^m} \exp\left(-\frac{\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2}{2\eta^2}\right)$ and
- the prior: $\pi_{\text{prior}}(\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi}\beta)^n} \exp(-\frac{1}{2\beta^2}\|\boldsymbol{x}\|_2^2)$ (Gaussian distribution),

then the **MAP estimate** can be obtained by solving

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \alpha \frac{1}{2}\|\boldsymbol{x}\|_2^2$$

with $\alpha = \eta^2/\beta^2$.

Example

If we have

- the likelihood: $\pi(\boldsymbol{b}\,|\,\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi}\eta)^m} \exp\left(-\frac{\|\boldsymbol{A}\,\boldsymbol{x}-\boldsymbol{b}\|_2^2}{2\eta^2}\right)$ and
- the prior: $\pi_{\text{prior}}(\boldsymbol{x}) = \exp(-\frac{1}{\beta}J(\boldsymbol{x}))$ (Gibbs prior) with $\beta > 0$,

then the **MAP estimate** can be obtained by solving

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}\|_2^2 + \alpha\, J(\boldsymbol{x})$$

with $\alpha = \eta^2/\beta$.

- The term $\frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}\|_2^2$ is called the *data-fidelity* term.
- The term $J(\boldsymbol{x})$ is called the *regularization* term.
- $\alpha > 0$ is the regularization parameter.

Poisson Measurements in X-ray

The measured transmission I_i in a single detector element follows a Poisson distribution $\mathcal{P}(I_0 \exp(-\boldsymbol{r}_i^T \boldsymbol{x}))$:

$$\pi(I_i \,|\, \boldsymbol{x}) = \frac{\left(I_0 \exp(-\boldsymbol{r}_i^T \boldsymbol{x})\right)^{I_i}}{I_i!} \exp\left(-I_0 \exp(-\boldsymbol{r}_i^T \boldsymbol{x})\right),$$

where $\boldsymbol{r}_i^T$ with $i = 1, \cdots, m$ denotes the row of the system matrix $\boldsymbol{A}$.

- The likelihood: $\pi(\boldsymbol{I} \,|\, \boldsymbol{x}) = \prod_{i=1}^{m} \pi(I_i \,|\, \boldsymbol{x})$.
- The ML estimate ($\boldsymbol{b} = -\log(\boldsymbol{I}/I_0)$):

$$\arg\min_{\boldsymbol{x}} -\log\left(\pi(\boldsymbol{b} \,|\, \boldsymbol{x})\right) \iff \arg\min_{\boldsymbol{x}} \exp(-\boldsymbol{b})^T \boldsymbol{A}\boldsymbol{x} + 1^T \exp(-\boldsymbol{A}\boldsymbol{x}).$$

- The MAP estimate: $\arg\min_{\boldsymbol{x}} \exp(-\boldsymbol{b})^T \boldsymbol{A}\boldsymbol{x} + 1^T \exp(-\boldsymbol{A}\boldsymbol{x}) + \alpha J(\boldsymbol{x})$.

Quadratic Approximation for Poisson Noise

Use the second-order Taylor expansion of

$$D_i(\tau) = \exp(-b_i)\,\tau + \exp(-\tau), \qquad i = 1, \ldots, m\ ,$$

to verify that the ML estimation problem can be approximated by the weighted quadratic problem

$$\min_{\boldsymbol{x}} \ \frac{1}{2}(\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b})^T W\,(\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b})$$

with $W = \mathrm{diag}(\exp(-\boldsymbol{b}))$.