Sparsity

Yubo Tao

May 21, 2019

Outline

- Examples of Applications of Sparsity
- L_2, L_1, and L_0 Norms
  - Linear Inverse Problems
  - Minimum L_2, L_1, and L_0 Norm Solution
- Solution Approaches
  - Matching Pursuit
  - Smooth Reformulations
  - Dictionary Learning
- Sparse Solutions to Some Applications

Reference:
Michael Elad, Sparse and Redundant Representations and Their Applications in Signal and Image Processing
Aggelos K. Katsaggelos, Fundamentals of Digital Image and Video Processing

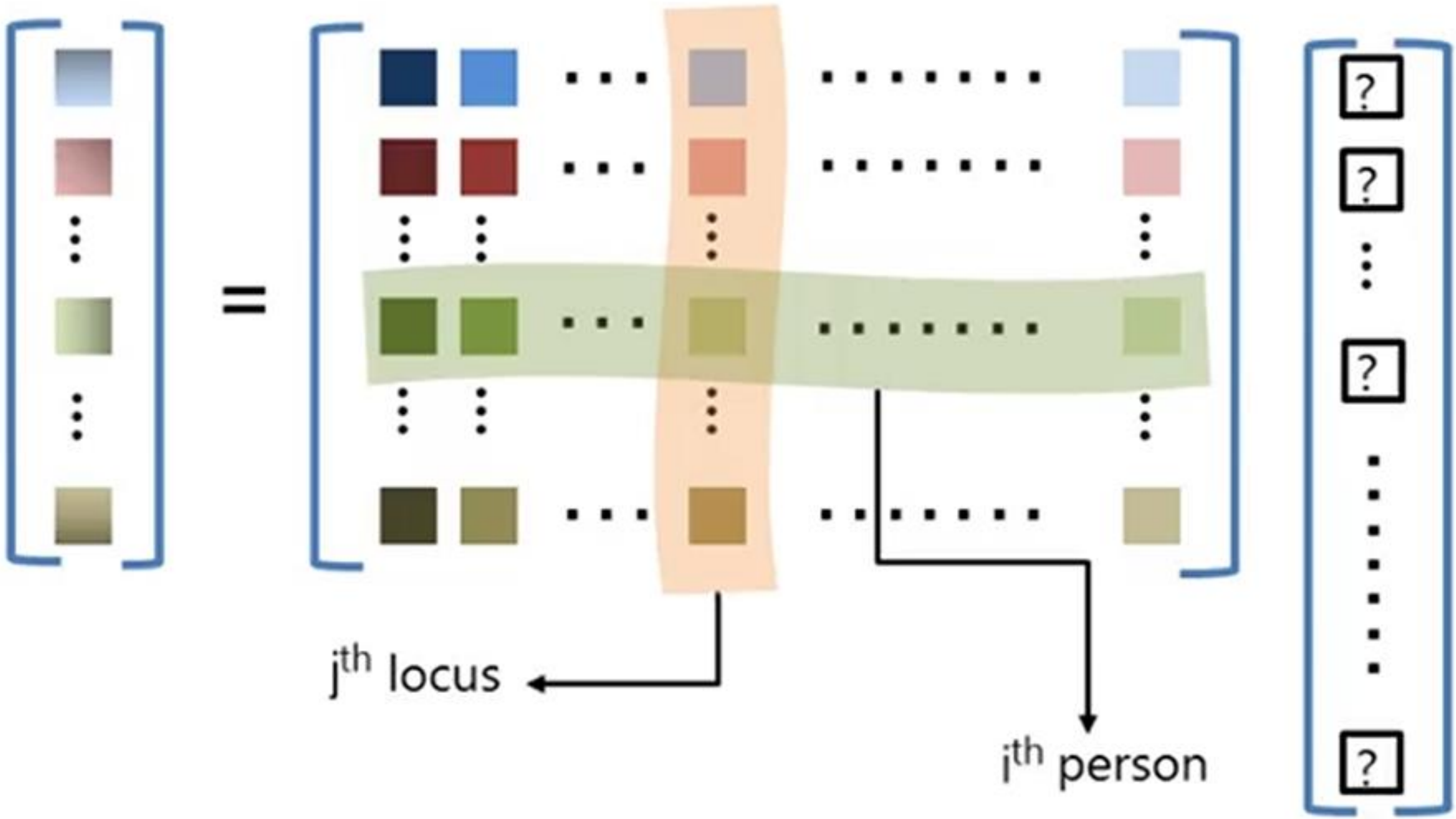
What is Sparsity?

- A vector is said to be sparse if it only has “a few” non-zero components
- The vector can represent a signal (image), which may be sparse in its native domain (e.g., image of sky at night) or can be made sparse in another domain (e.g., natural images in the DFT domain)
- A sparse vector may originate in numerous applications

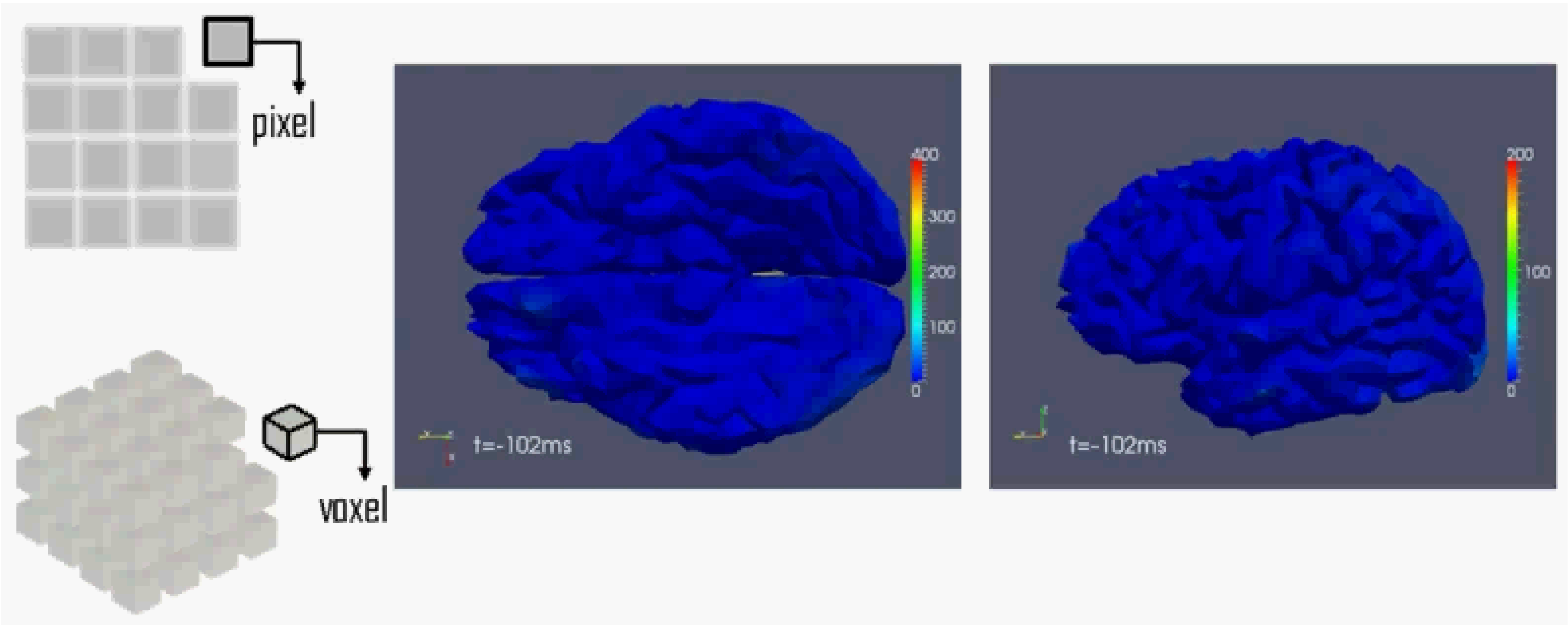
Applications

- Image and Video Processing
- Machine Learning
- Statistics
- Genetics
- Econometrics
- Neuroscience
- …

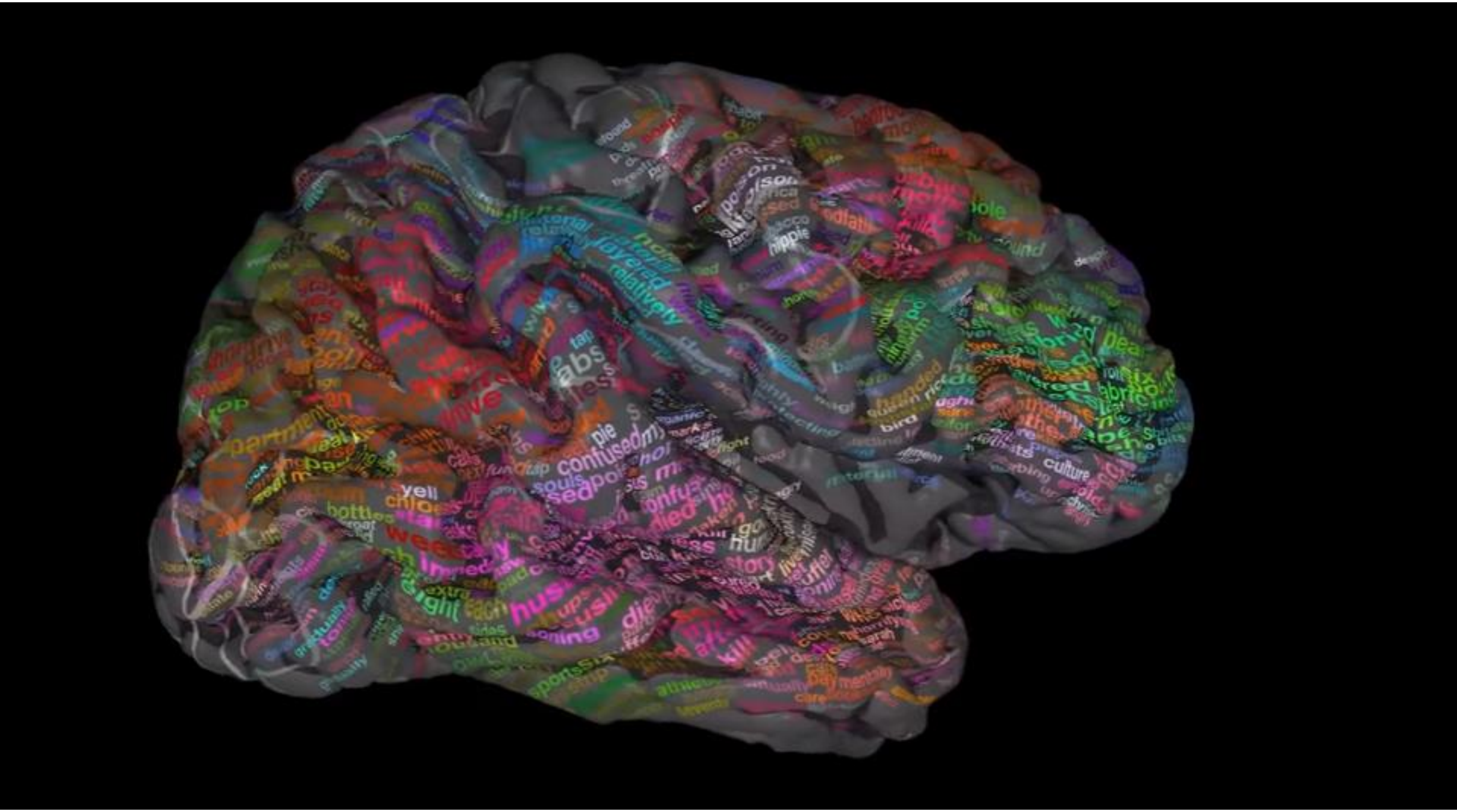
Genetics



Neuroscience

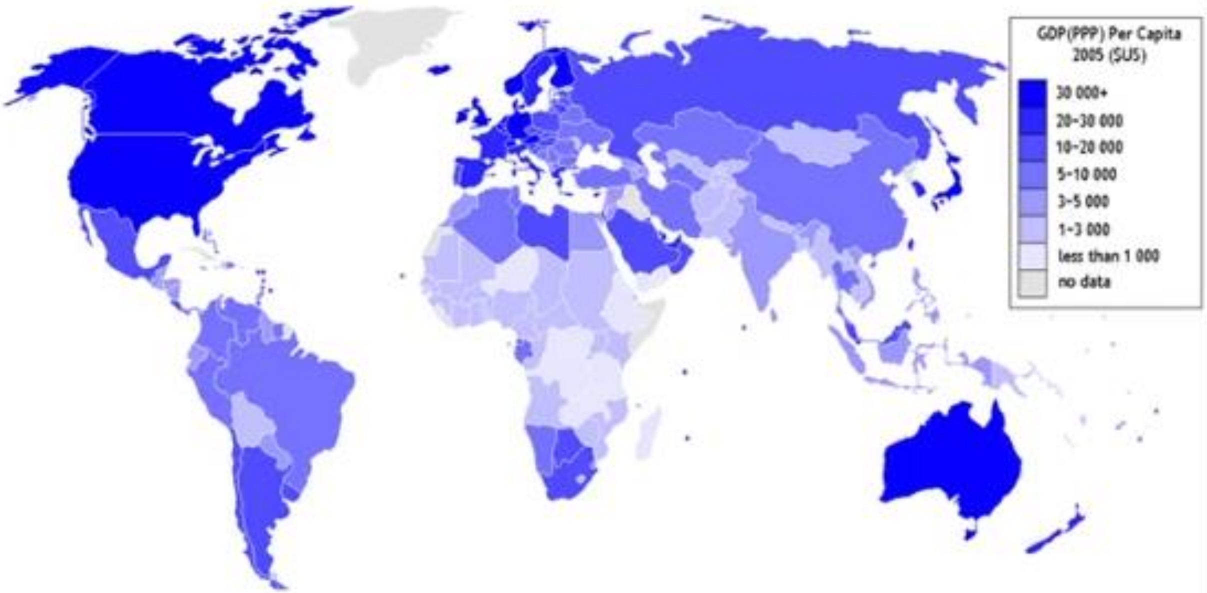


Neuroscience



Econometrics

- $Ax = b$
  - x sparse



Population density
Fraction of tropical area
Size of economy
Defense spending
Life expectancy
Public investment

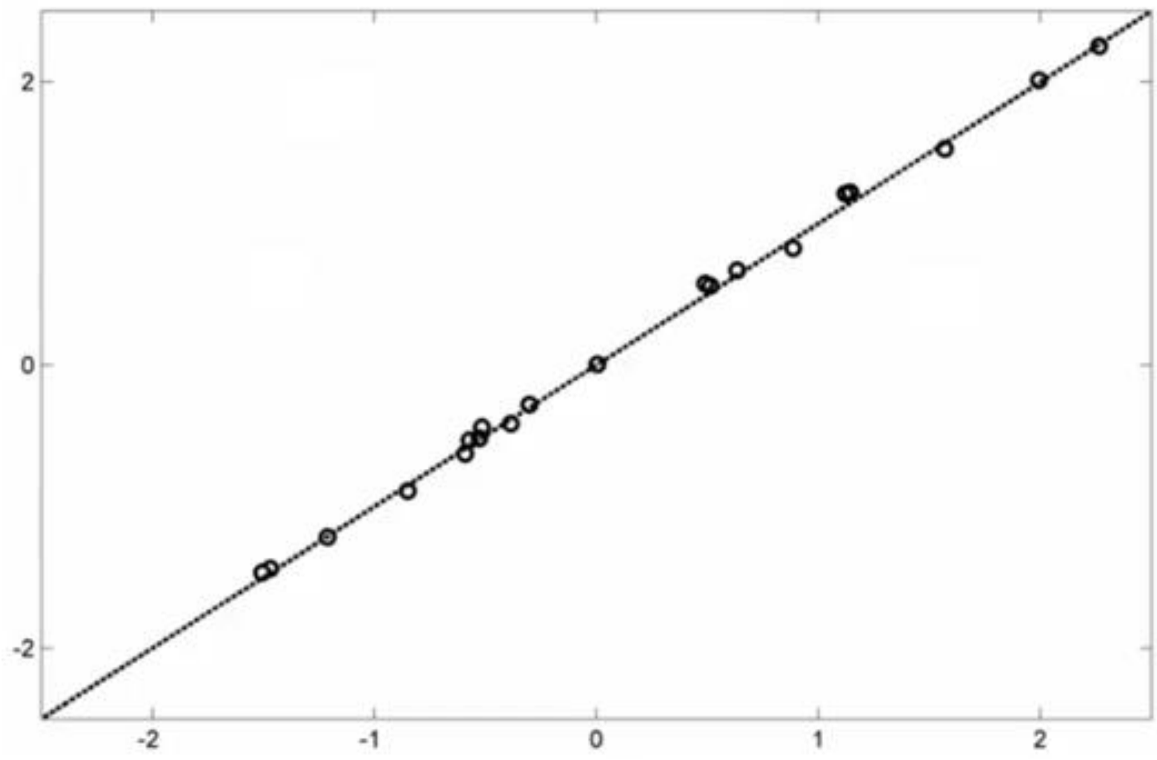
⋮

Land area

SOURCE: mapsof.net

Robust Regression

- $(a_i, b_i), a_i {\color{red}x} \approx b_i$
- $\min_x \sum_i (a_i {\color{red}x} - b_i)^2$
- $A{\color{red}x} \approx b$
- $\min_x \|A{\color{red}x} - b\|_2^2$

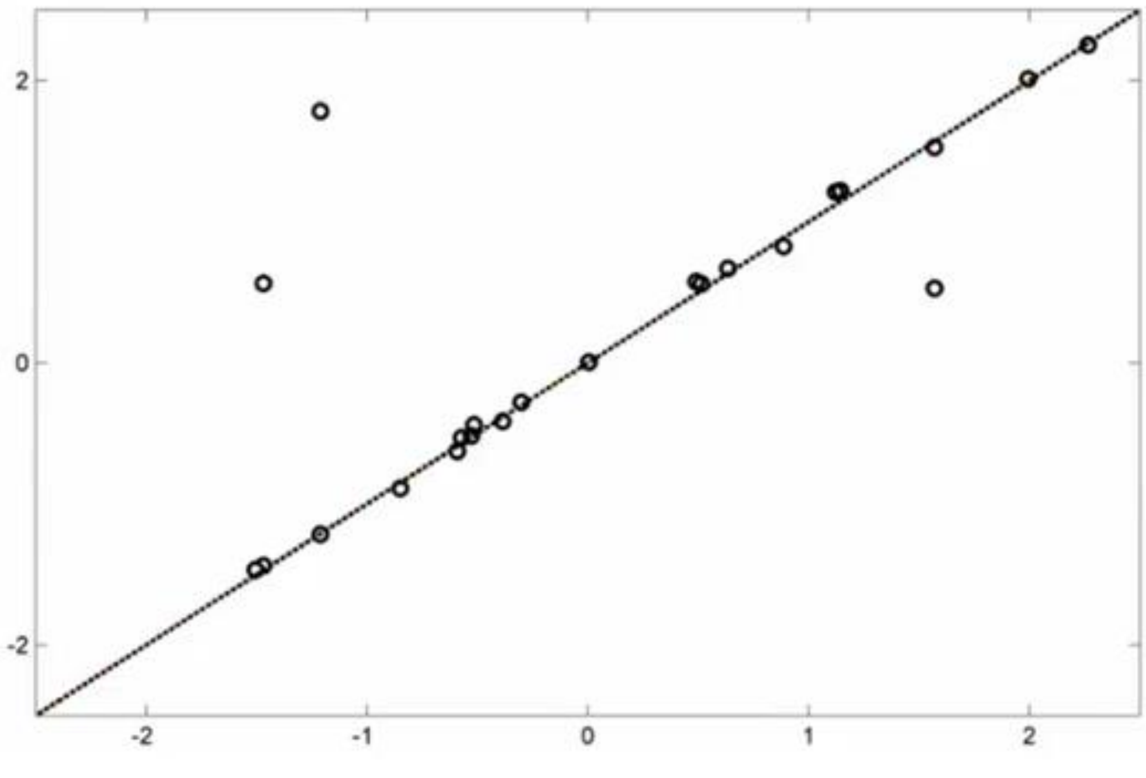


Matrix Calculus

- Example: Least-squares
  - $x = [x_1 \dots x_n]$, $\|x\|_2^2 = x^T x = \sum_1^n {x_i}^2$
  - $\|b - Ax\|_2^2 = (b - Ax)^T (b - Ax)$
  - $\qquad = b^T b - b^T Ax - x^T A^T b + x^T A^T Ax$
  - $\qquad = b^T b - 2x^T A^T b + x^T A^T Ax$
  - $D\|x\|_2^2 = \begin{bmatrix} \partial_{x_1} \|x\|_2^2 \\ \vdots \\ \partial_{x_n} \|x\|_2^2 \end{bmatrix} = \begin{bmatrix} \partial_{x_1} \sum_1^n {x_i}^2 \\ \vdots \\ \partial_{x_n} \sum_1^n {x_i}^2 \end{bmatrix} = \begin{bmatrix} 2x_1 \\ \vdots \\ 2x_n \end{bmatrix} = 2x$
  - $D_x(x^T y) = y, D_y(x^T y) = x$, $D(x^T A^T Ax) = 2A^T Ax$
  - $D(\|b - Ax\|_2^2) = -2A^T b + 2A^T Ax = 0$

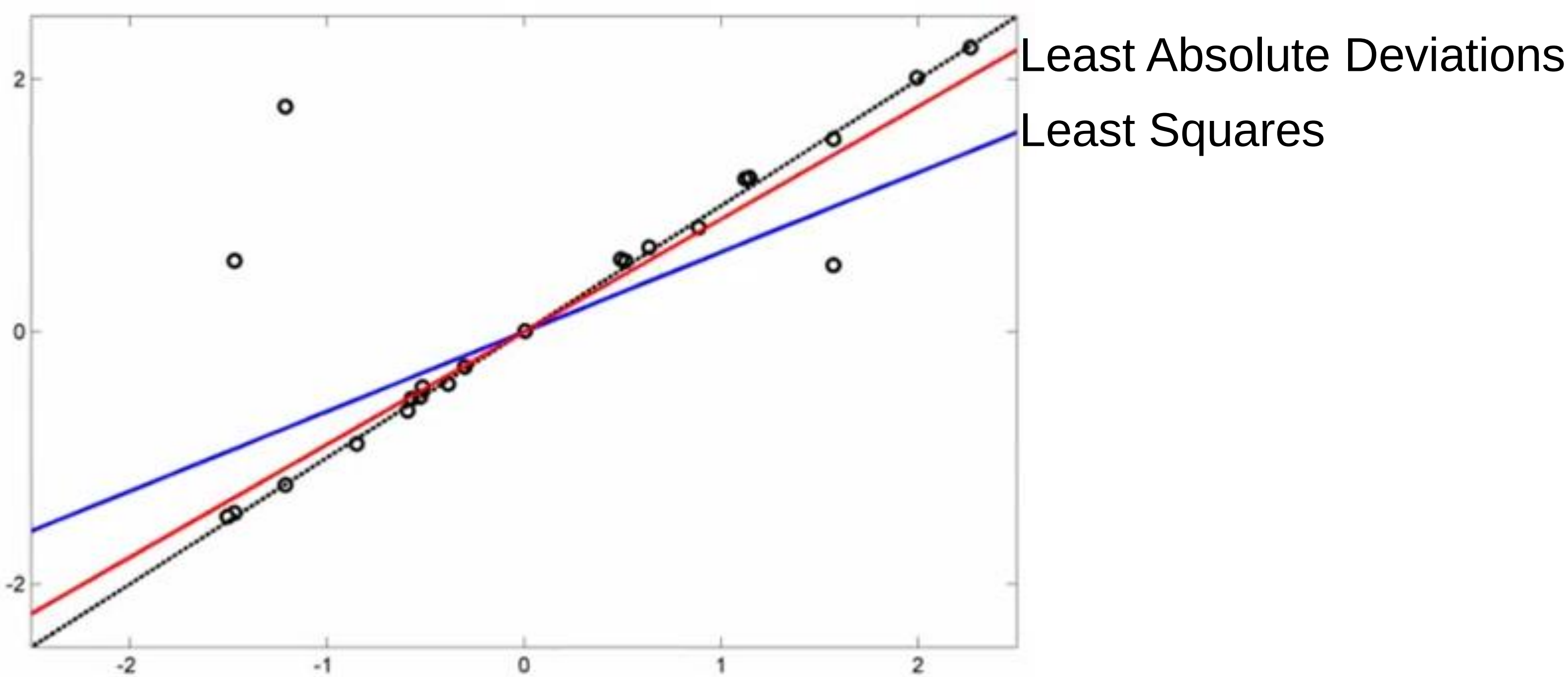
https://en.wikipedia.org/wiki/Matrix_calculus

Robust Regression



Robust Regression

- $Ax + e = b$
  - e a sparse vector
  - Least Absolute Deviations (最小绝对偏差)



Recommender Systems

- Matrix Completion Problem
- Rank Minimization Problem

Image Denoising

- $y_i \cong Ax_i$ (*A* a fixed dictionary, *x* a sparse vector)
- $\min_{x_i} \|y_i - Ax_i\|_2^2 + \lambda \|x_i\|_1$

Image Inpainting

- $y_i \cong RAx_i$ (R mask, A dictionary, x sparse)
- $x^* = \min_{x_i} \|y_i - RAx_i\|_2^{\ 2} + \lambda \|x_i\|_1$

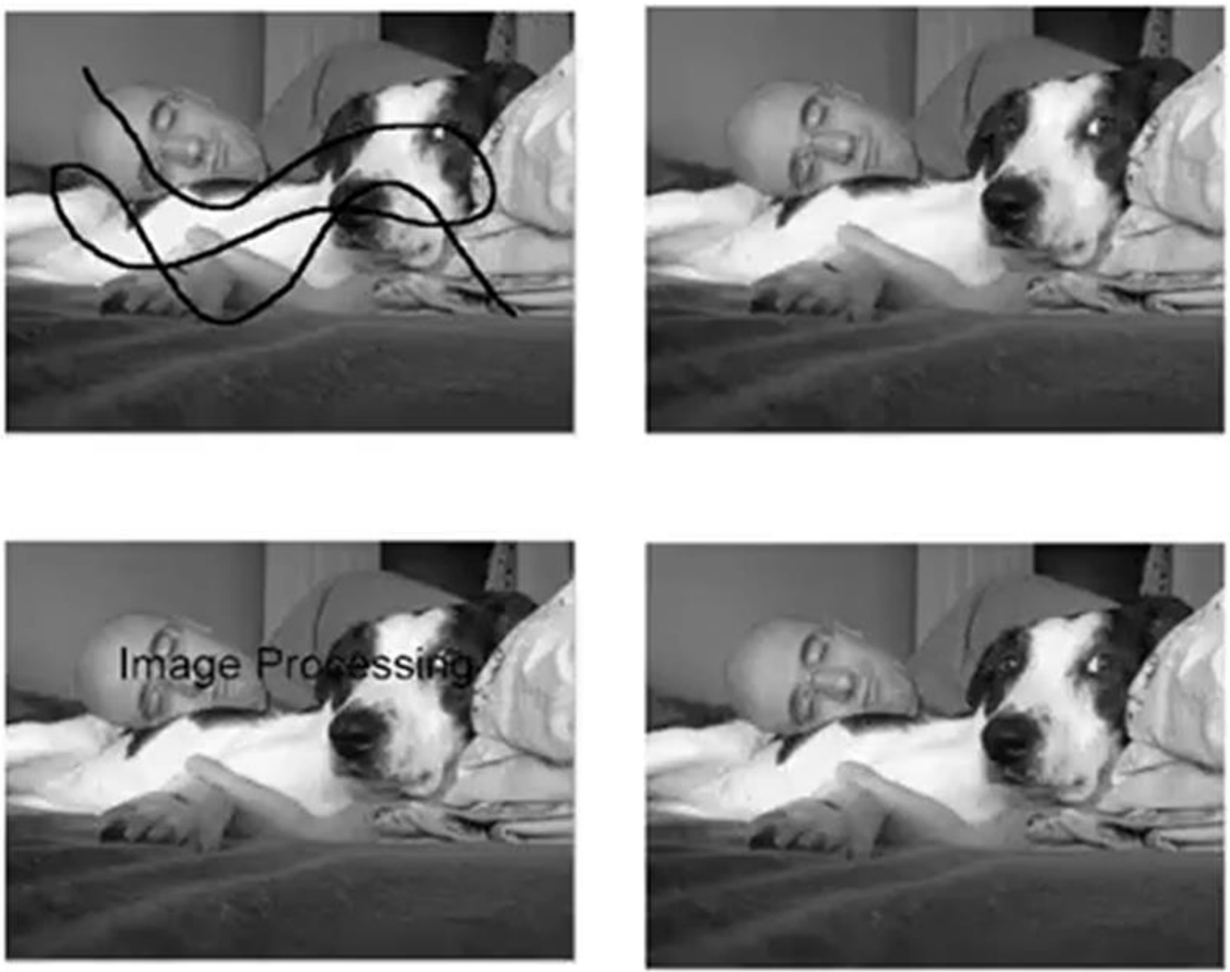
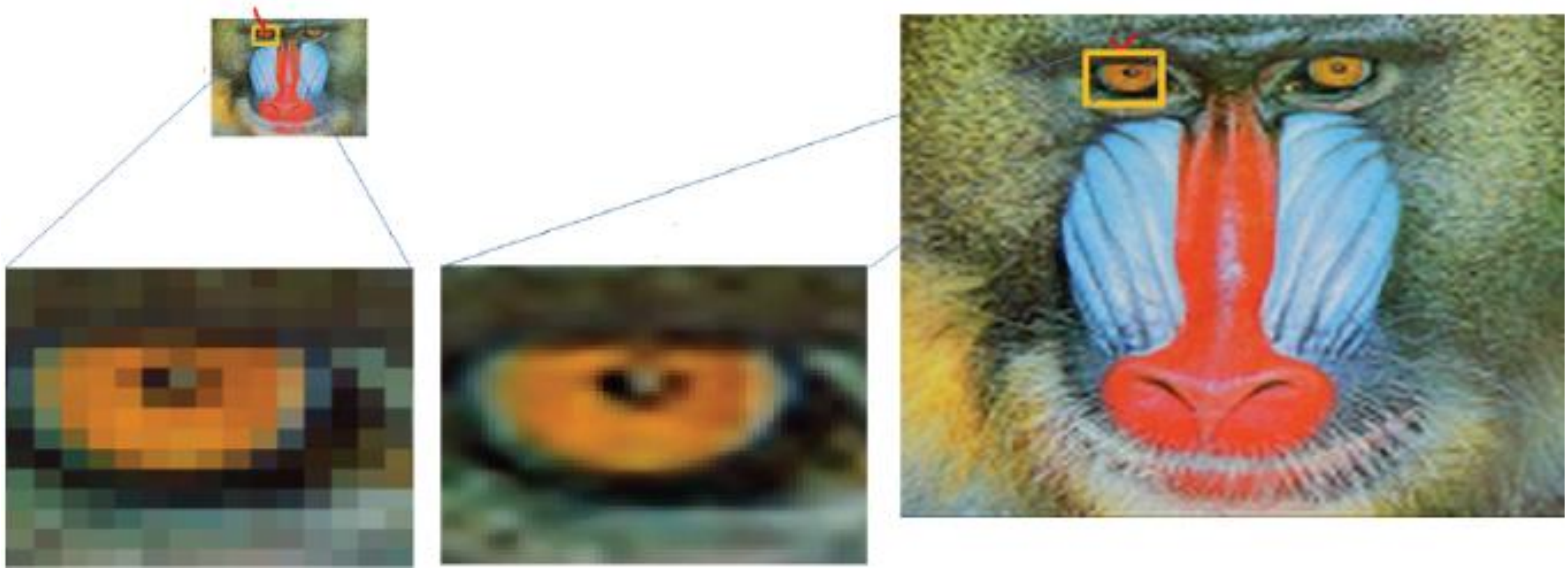


Image Super-Resolution

- $y_{LR} = A_{LR}\, x_{sparse}$
- $y_{HR} = A_{HR}\, x_{sparse}$



Video Surveillance

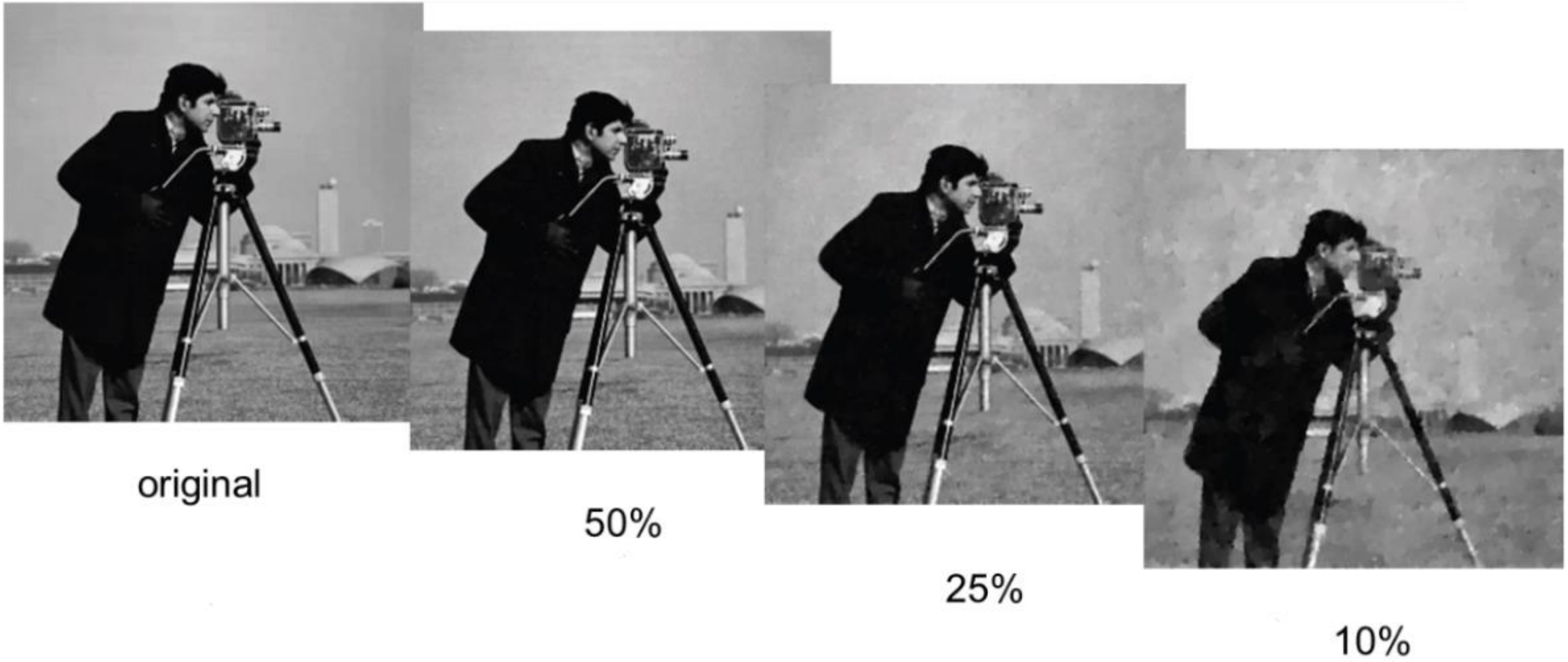
Background
Low-rank matrix

Foreground
Sparse matrix

Robust Face Recognition

- $b = Ax + e$
  - x and e both sparse

Compressive Sensing

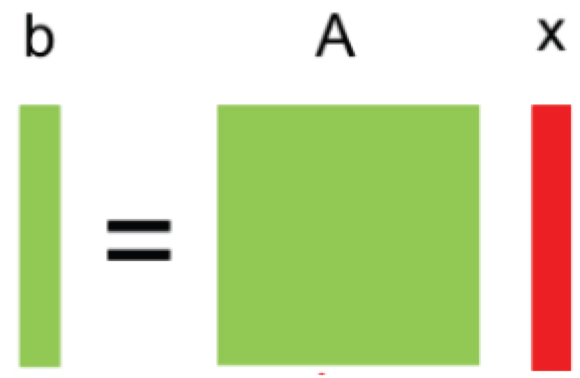


Outline

- Examples of Applications of Sparsity
- L_2, L_1, and L_0 Norms
  - Linear Inverse Problems
  - Minimum L_2, L_1, and L_0 Norm Solution
- Solution Approaches
  - Matching Pursuit
  - Smooth Reformulations
  - Dictionary Learning
- Sparse Solutions to Some Applications
  - Image Denoising, Image Inpainting, Image Super-Resolution, Robust Face Recognition, Video Surveillance, Compressive Sensing

Linear Inverse Problems

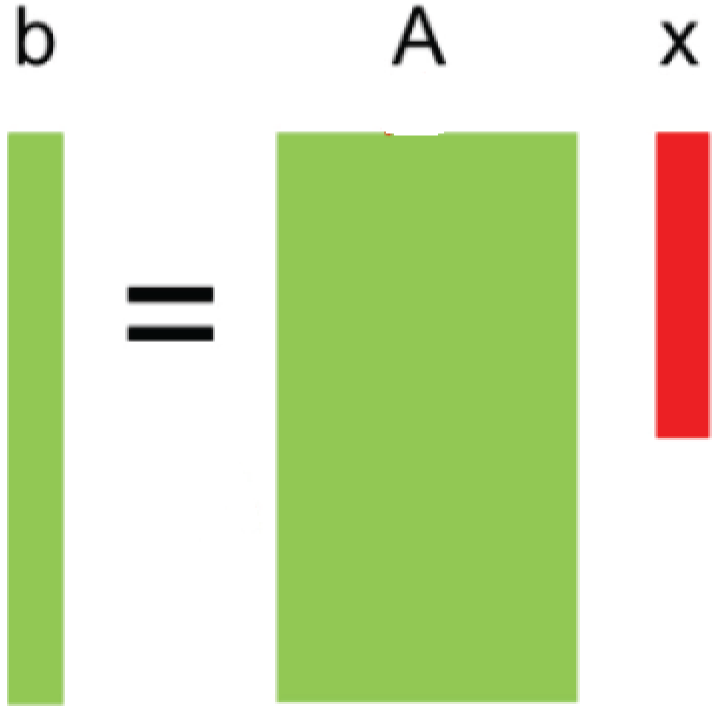
- Full-determined system of equations
  - # equations = # unknowns



- Unique solution (if A is full rank)
  - $x^* = A^{-1}b$
  - $x^* = \frac{b}{A}$?

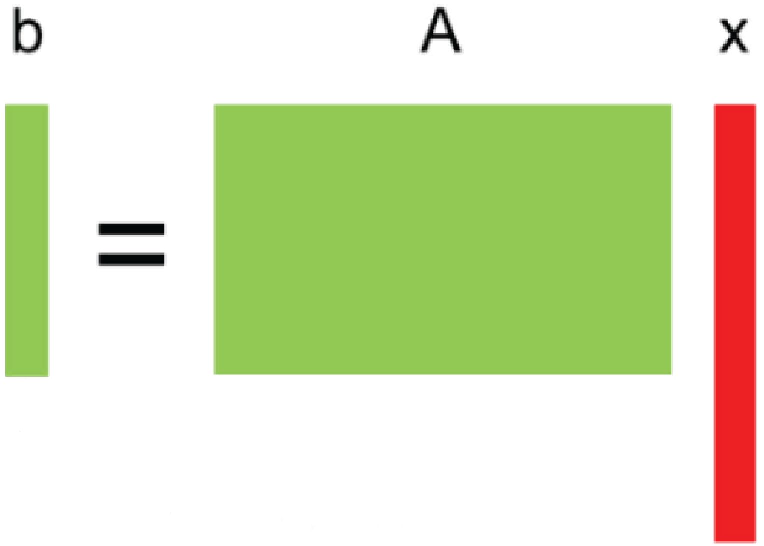
Linear Inverse Problems

- Over-determined system of equations (超定方程)
  - # equations > # unknowns
- The Least Squares solution is given by
  - $x^* = (A^T A)^{-1} A^T b$



Linear Inverse Problems

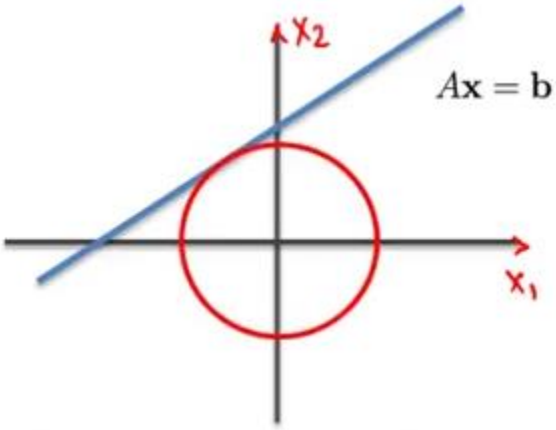
- Under-determined system of equations (欠定方程)
  - # equations < # unknowns
- Infinitely many solutions (usually!)
- How to pick x? it depends on the application.
- Regularization (正则化)
  - $\min_{x} J(x) \ subject\ to\ b = Ax$



Minimum L_2 Norm Solution

- We want x to be 'small' (in the L_2 sense)

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i{}^2}$$



- The problem to solve is

$$\min_x \|x\|_2$$
$$subject\ to\ b = Ax$$

- The closed form solution is given by

$$x^* = A^T(AA^T)^{-1}b$$

KKT Conditions

- The vector $x^* \in R^n$ is a critical point for minimizing f subject to $g(x) = 0$ and $h(x) \geq 0$ when there exists $\lambda \in R^m$ and $\mu \in R^p$ such that
  - $0 = \nabla f(x^*) - \sum_i \lambda_i \nabla g(x^*) - \sum_j \mu_j \nabla h_j(x^*)$ (stationarity)
  - $g(x^*) = 0$ and $h(x^*) \geq 0$ (primal feasibility)
  - $\mu_j h_j(x^*) = 0$ for all j (complementary slackness)
  - $\mu_j \geq 0$ for all j (dual feasibility)
- When h is removed, this reduces to the Lagrange multiplier (拉格朗日乘子法) criterion

KKT: http://www.cad.zju.edu.cn/home/zhx/csmath/doku.php?id=2019

Lagrangian

Consider general minimization problem

$$\begin{aligned}\min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \dots m \\ & \ell_j(x) = 0, \quad j = 1, \dots r\end{aligned}$$

Need not be convex, but of course we will pay special attention to convex case

We define the Lagrangian as

$$L(x, u, v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x)$$

New variables $u \in \mathbb{R}^m$, $v \in \mathbb{R}^r$, with $u \geq 0$ (implicitly, we define $L(x, u, v) = -\infty$ for $u < 0$)

Minimum L_2 Norm Solution

- Derivation of closed form solution

$$\min_x \|x\|_2$$
$$subject\ to\ b = Ax$$

- $\min_x(\|x\|_2 + \lambda^T(Ax - b)) = \min_x L(x)$
- KKT
  - $\nabla_x L(x) = 0 \qquad x + A^T\lambda = 0 \qquad x = -A^T\lambda$
  - $\nabla_\lambda L(x) = 0 \qquad Ax - b = 0 \qquad -AA^T\lambda = b$

$$\lambda = -(AA^T)^{-1}b$$

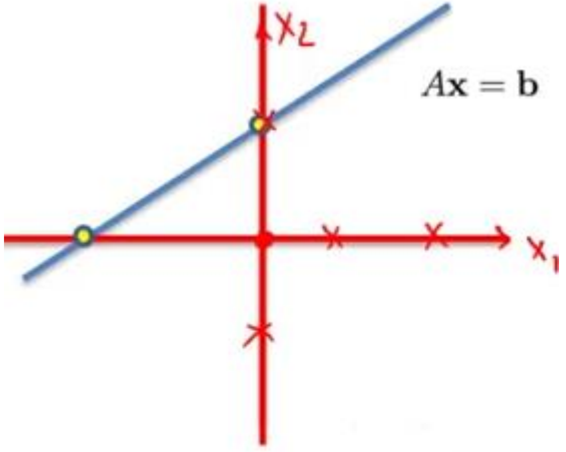
$$x^* = A^T(AA^T)^{-1}b$$

Minimum L_0 Norm Solution

- We want x to be ‘sparse’
  - It should have few non-zero entries

- Sparsity can be modeled via the L_0 norm

$$||x||_0 = \#non - zero\ entries\ in\ \boldsymbol{x}$$



- The problem to solve it now

$$\min_x ||x||_0$$
$$subject\ to\ b = Ax$$

- Find the sparsest solution x to $Ax = b$
  - NP-hard

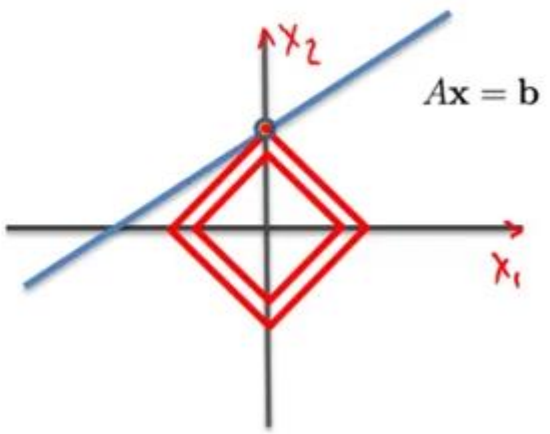
Minimum L_1 Norm Solution

- Another special solution is the one with minimum L_1 norm

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

- The problem to solve is now

$$\min_x \|x\|_1$$
$$subject\ to\ b = Ax$$



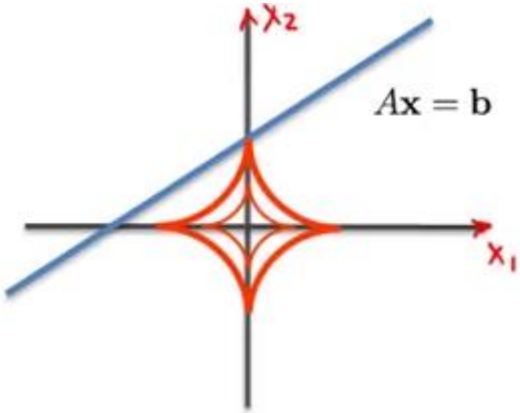
Minimum L_p Norm Solution

- Another class of special solutions minimizes the L_p norm ($0 < p < 1$)

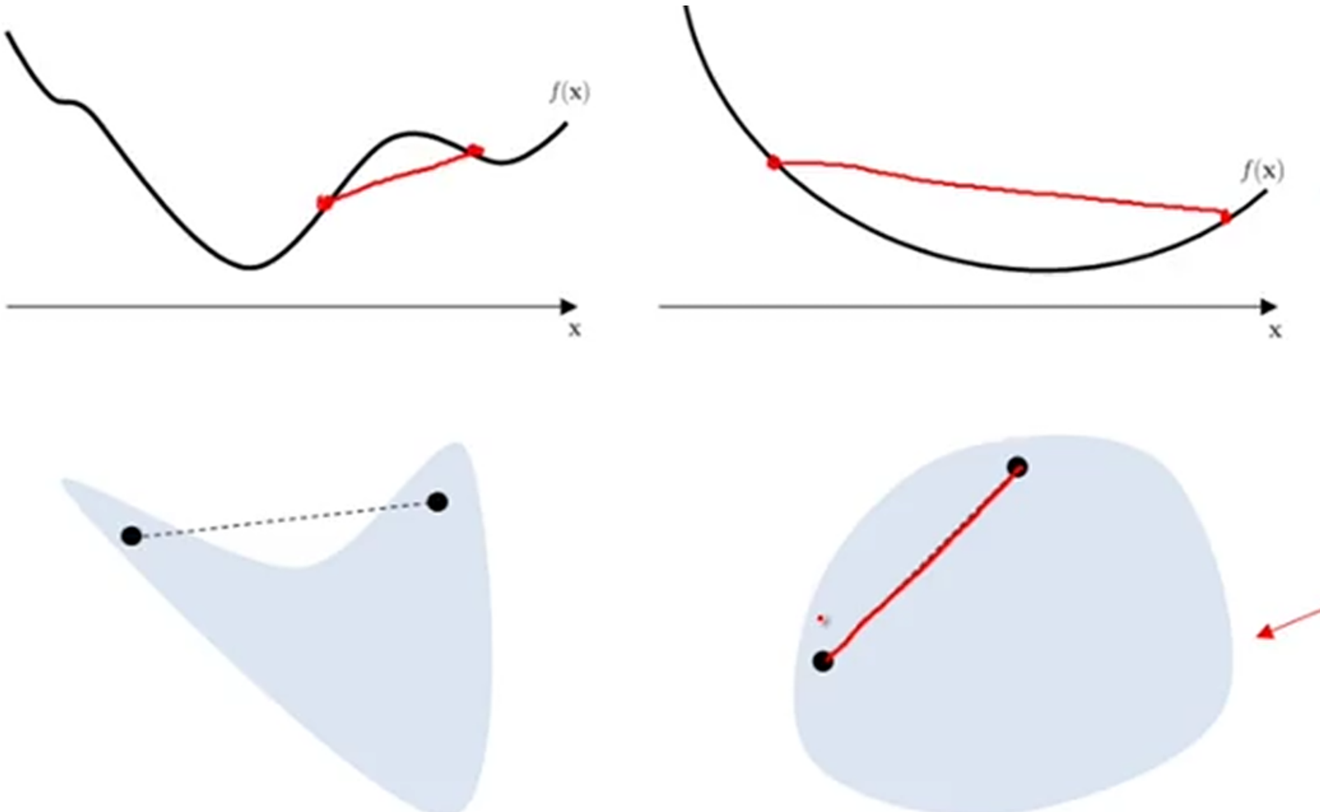
$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

- The problem to solve is now

$$\min_x \|x\|_p$$
$$subject\ to\ b = Ax$$

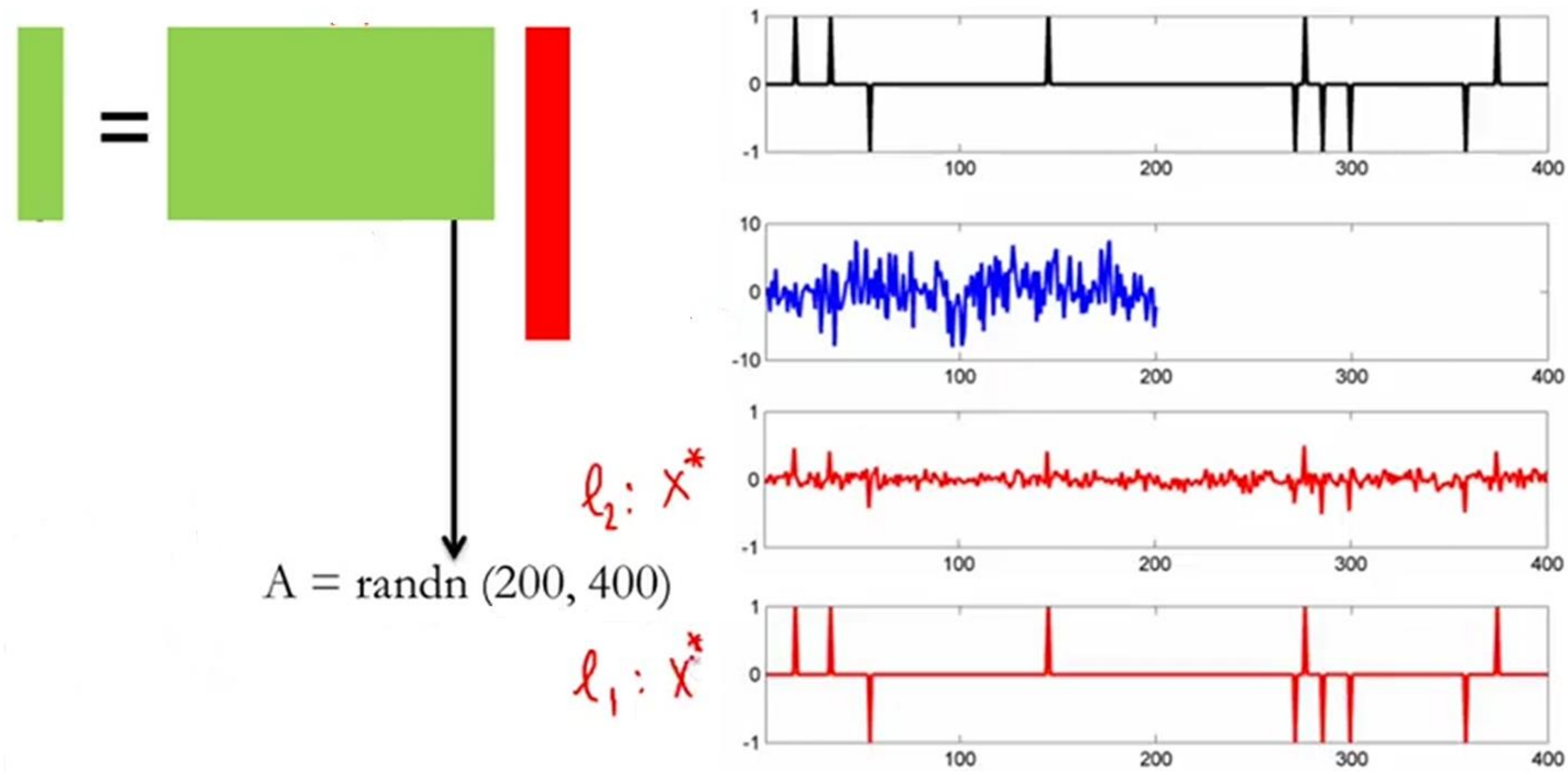


On Convexity

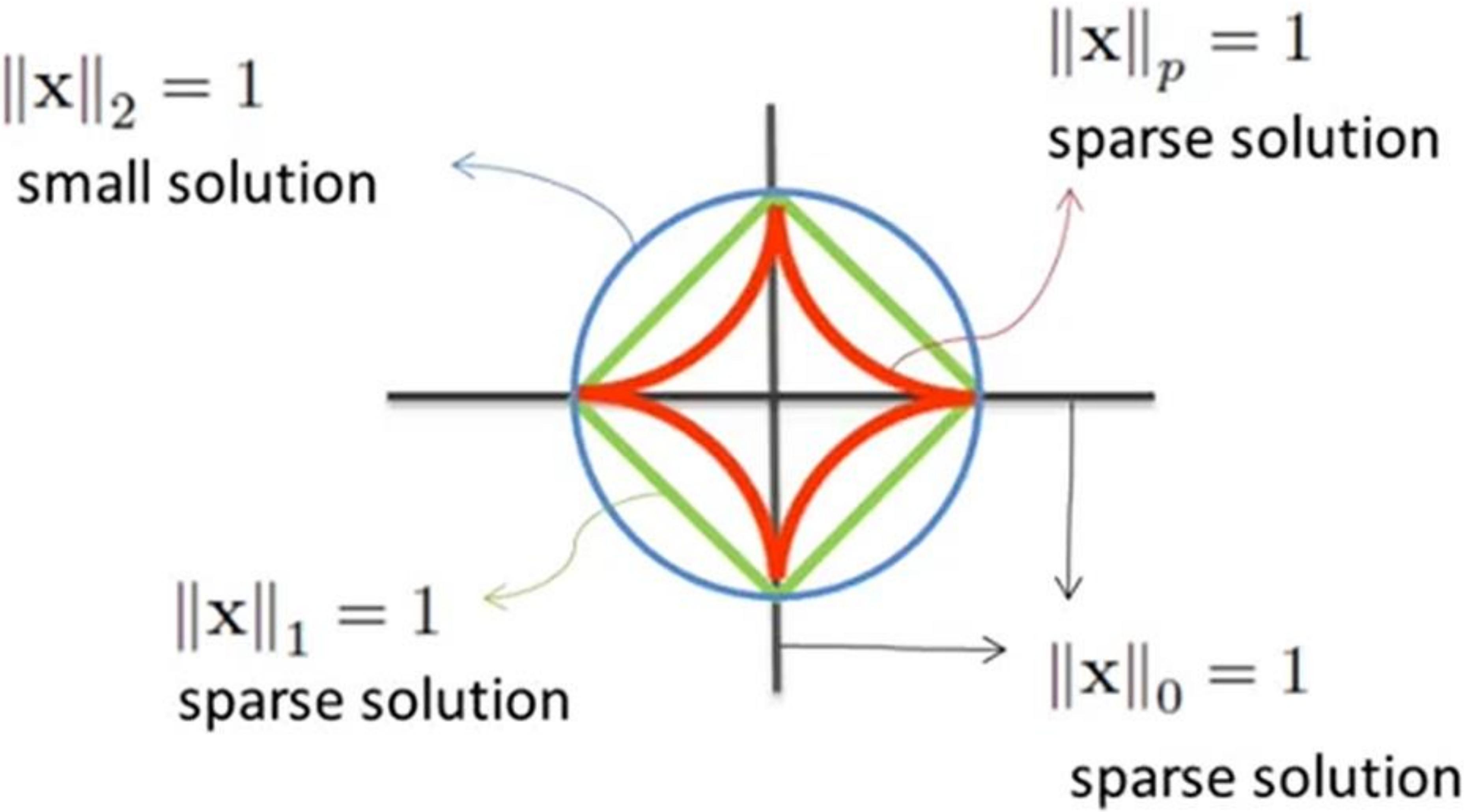


$$\min_{\mathbf{x}} f(\mathbf{x})$$
$$\text{subject to } \mathbf{x} \in \mathcal{S}$$

L_2 norm vs. L_1 norm



All Norm Balls in One Picture



L_0 norm vs. L_1 norm

$$\min_x \|x\|_0$$
$$subject\ to\ b = Ax$$

- Models sparsity directly
- Non-convex
- NP-hard
- Greedy approaches (Matching Pursuit) approximate the solution

$$\min_x \|x\|_1$$
$$subject\ to\ b = Ax$$

- Models sparsity indirectly
- Convex
- Non-smooth
- Can be solved via convex optimization algorithms

Outline

- Examples of Applications of Sparsity
- L_2, L_1, and L_0 Norms
  - Linear Inverse Problems
  - Minimum L_2, L_1, and L_0 Norm Solution
- Solution Approaches
  - Matching Pursuit
  - Smooth Reformulations
  - Dictionary Learning
- Sparse Solutions to Some Applications
  - Image Denoising, Image Inpainting, Image Super-Resolution, Robust Face Recognition, Video Surveillance, Compressive Sensing

Reformulation

$$\min_{\mathbf{x}} ||\mathbf{x}||_0$$
$$subject\, to\; A\mathbf{x} = \mathbf{b}$$

Noise in observation

$$\min_{\mathbf{x}} ||\mathbf{x}||_0$$
$$subject\, to\; ||A\mathbf{x} - \mathbf{b}||_2 \leq \epsilon$$

x(e)

Swapping constraint & objective

$$\min_{\mathbf{x}} ||A\mathbf{x} - \mathbf{b}||_2$$
$$subject\, to\; ||\mathbf{x}||_0 \leq S$$

x(S)

Reformulation

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to } A\mathbf{x} = \mathbf{b}$$

Noise in observation

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to } \|A\mathbf{x} - \mathbf{b}\|_2 \le \epsilon$$

Swapping constraint & objective

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2 \quad \text{subject to } \|\mathbf{x}\|_1 \le S$$

Bringing constraint to the objective

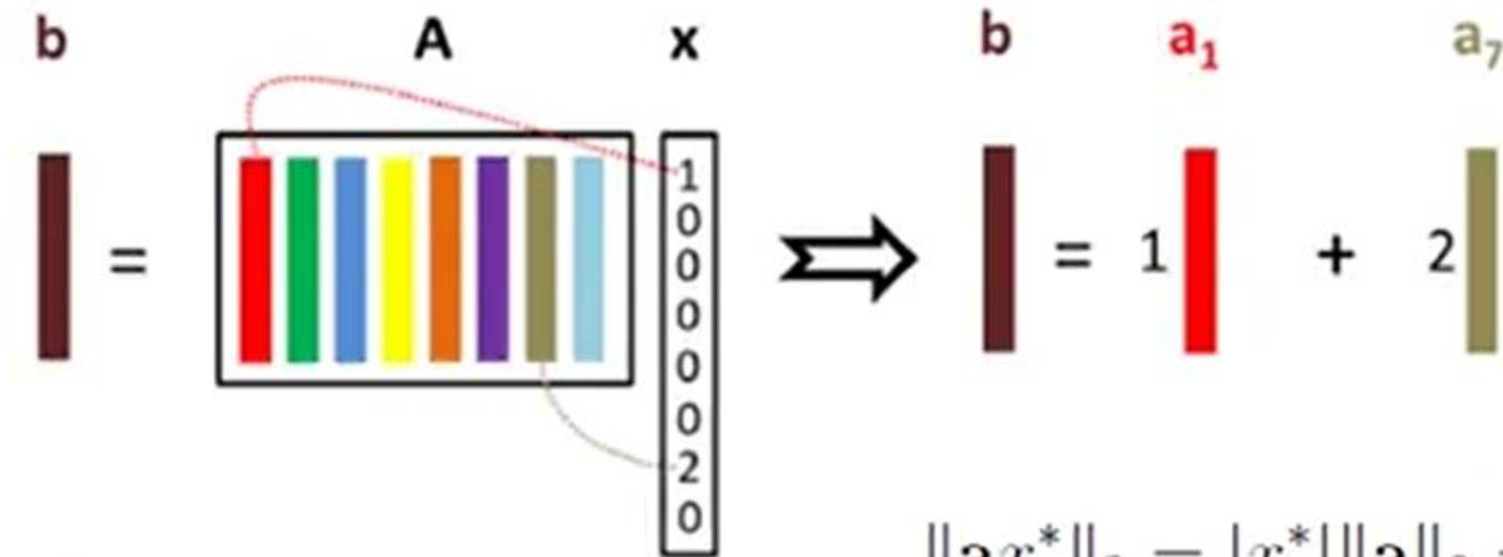
$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2 + \lambda \|\mathbf{x}\|_1$$

$x(\lambda) \le S$

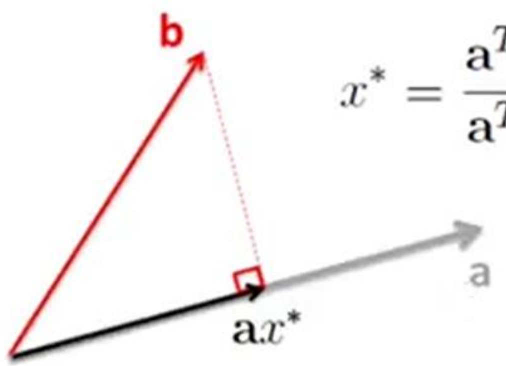
LASSO problem

Matching Pursuit

$$\min_{x} \|Ax - b\|_2$$
$$subject\ to\ \|x\|_0 \leq S$$



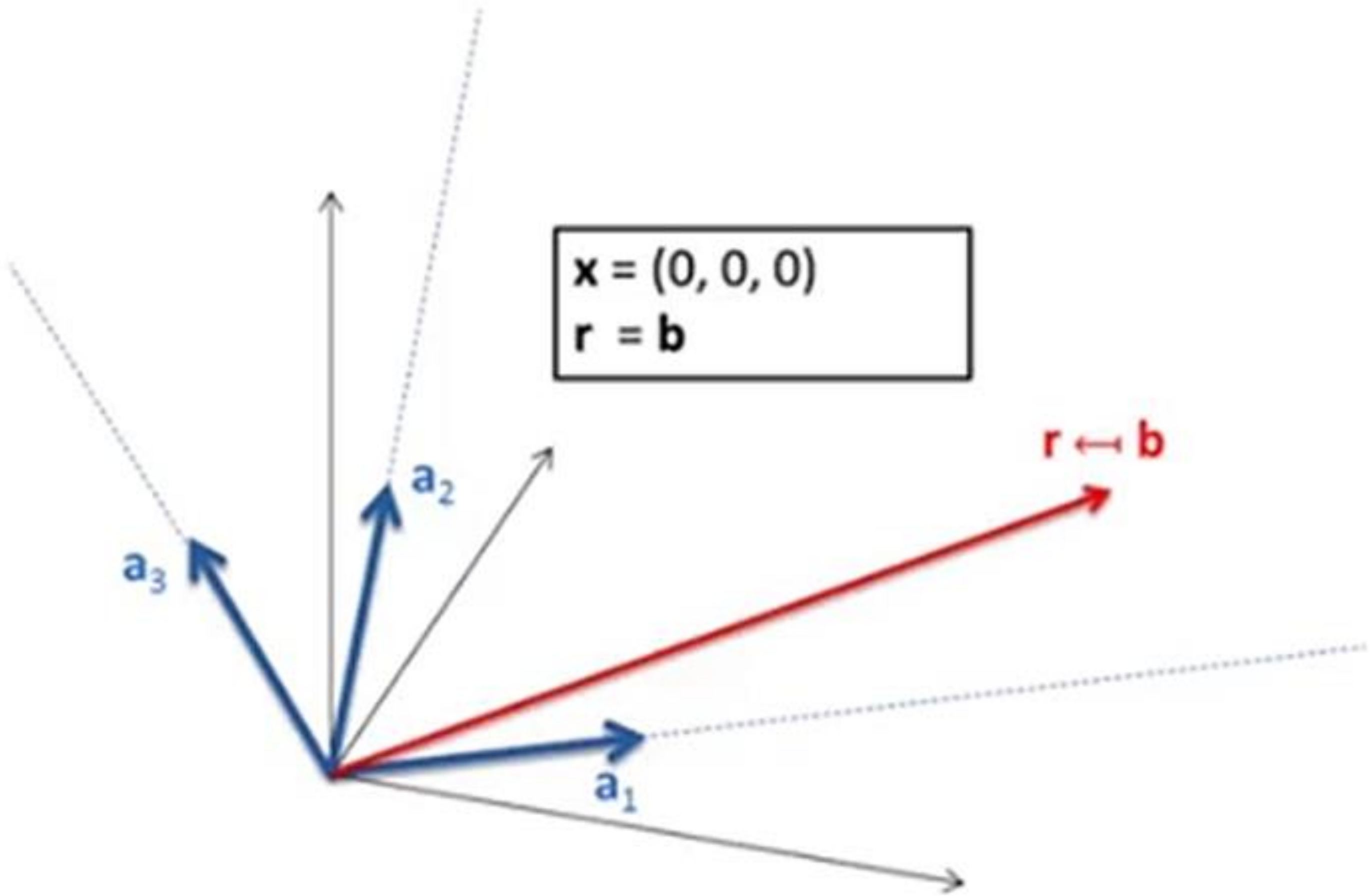
$$\|\mathbf{a}x^*\|_2 = |x^*|\|\mathbf{a}\|_2 = |x^*|$$



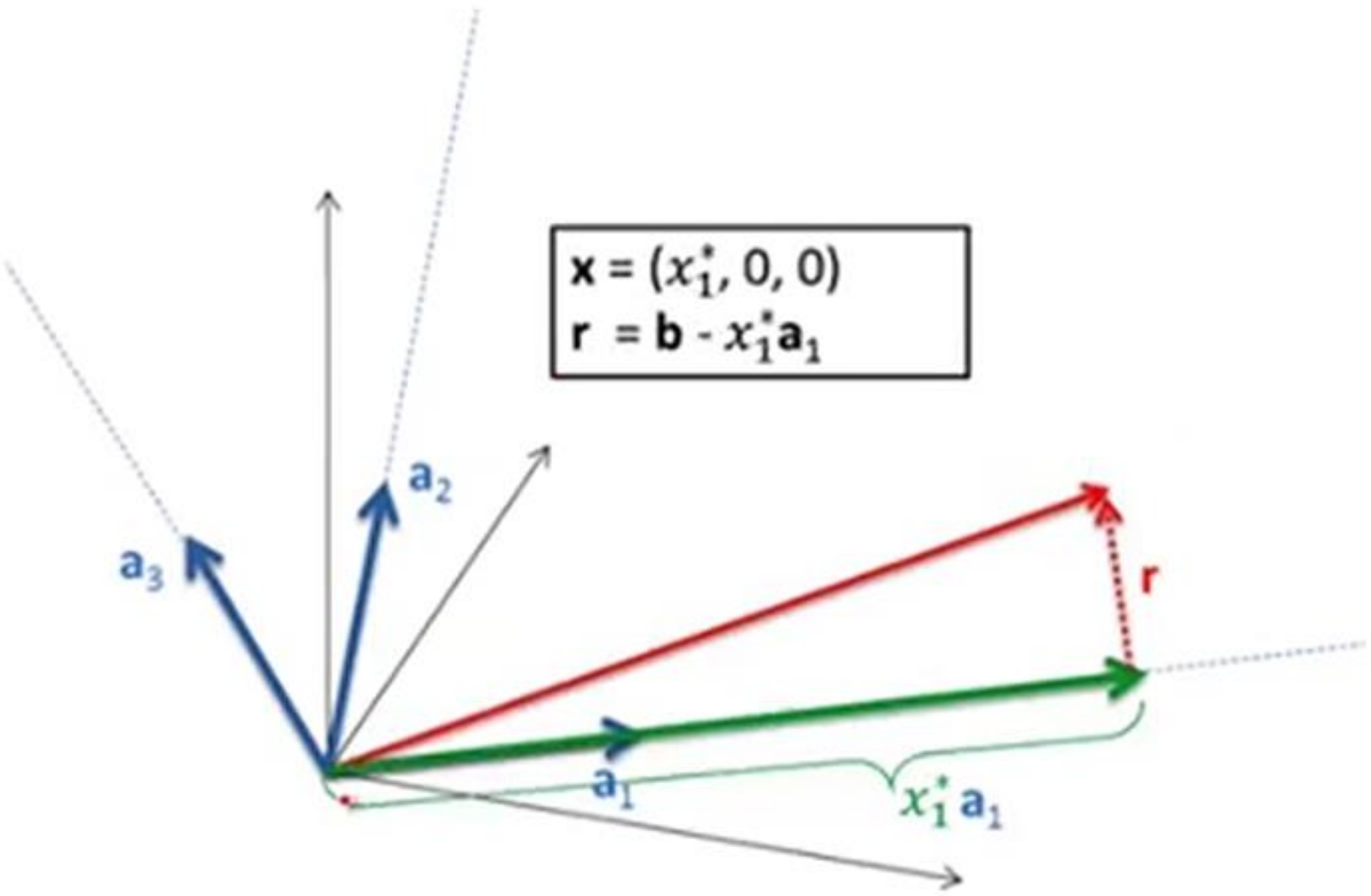
$$x^* = \frac{\mathbf{a}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}} = \mathbf{a}^T\mathbf{b}$$

Best column $i = \underset{k}{argmax}\, |x_k^*|$

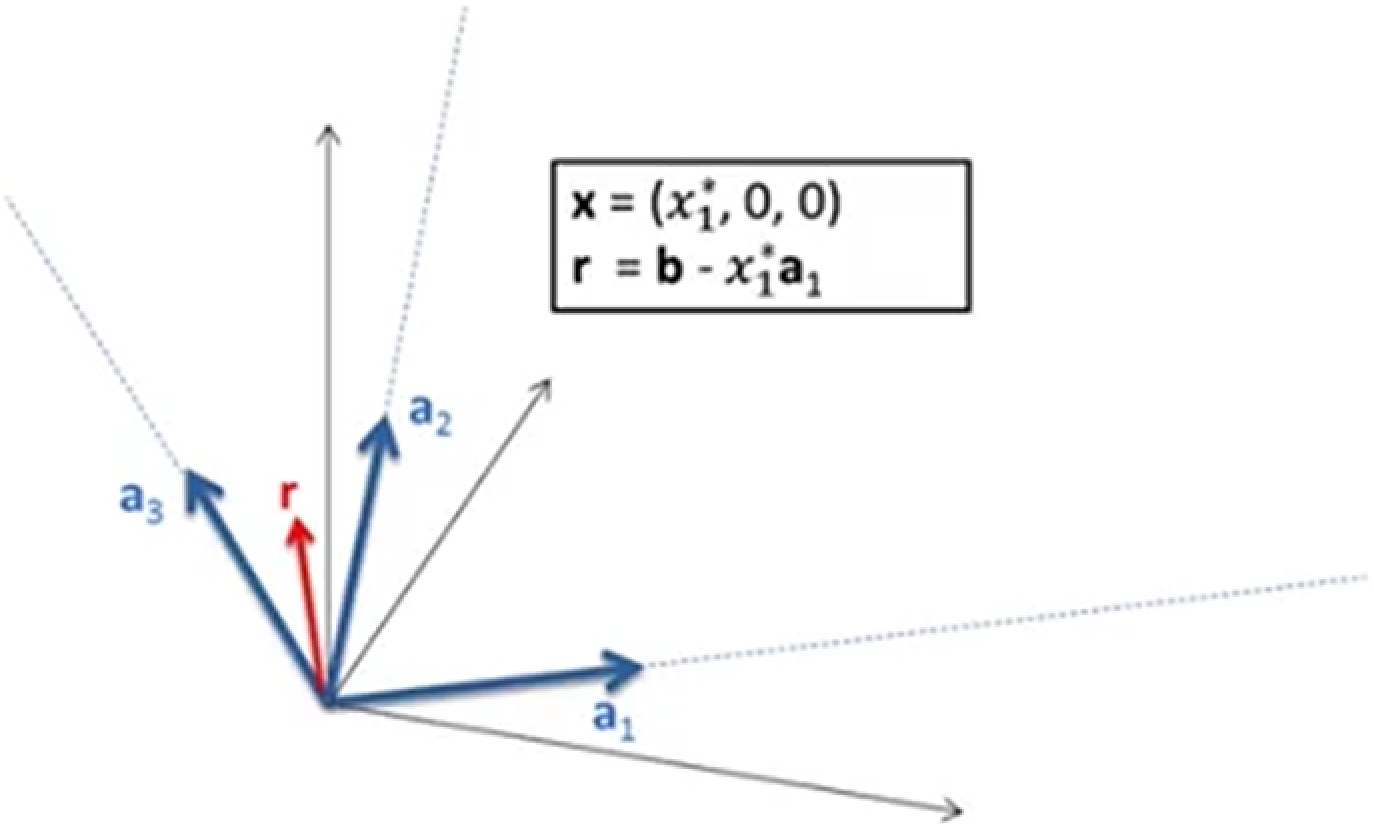
Matching Pursuit



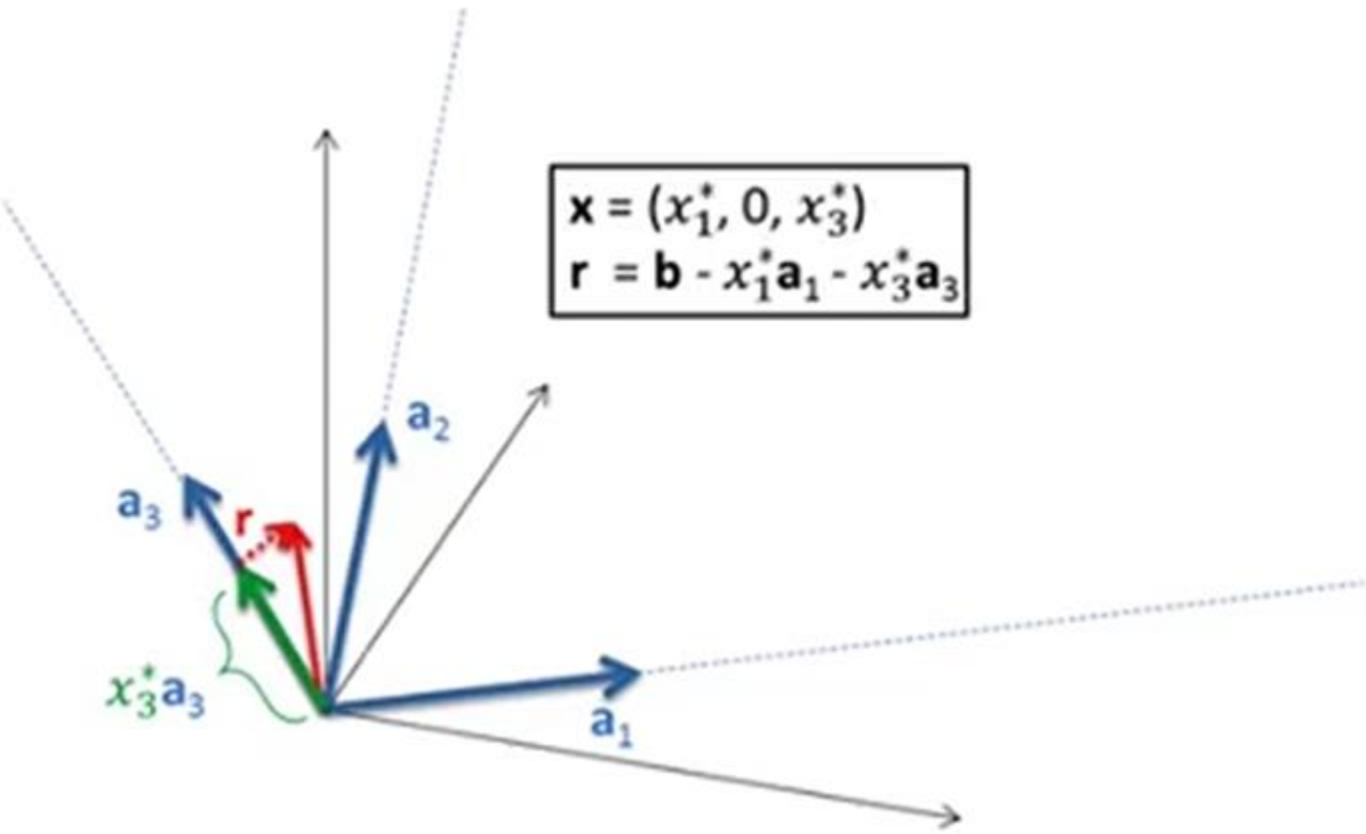
Matching Pursuit



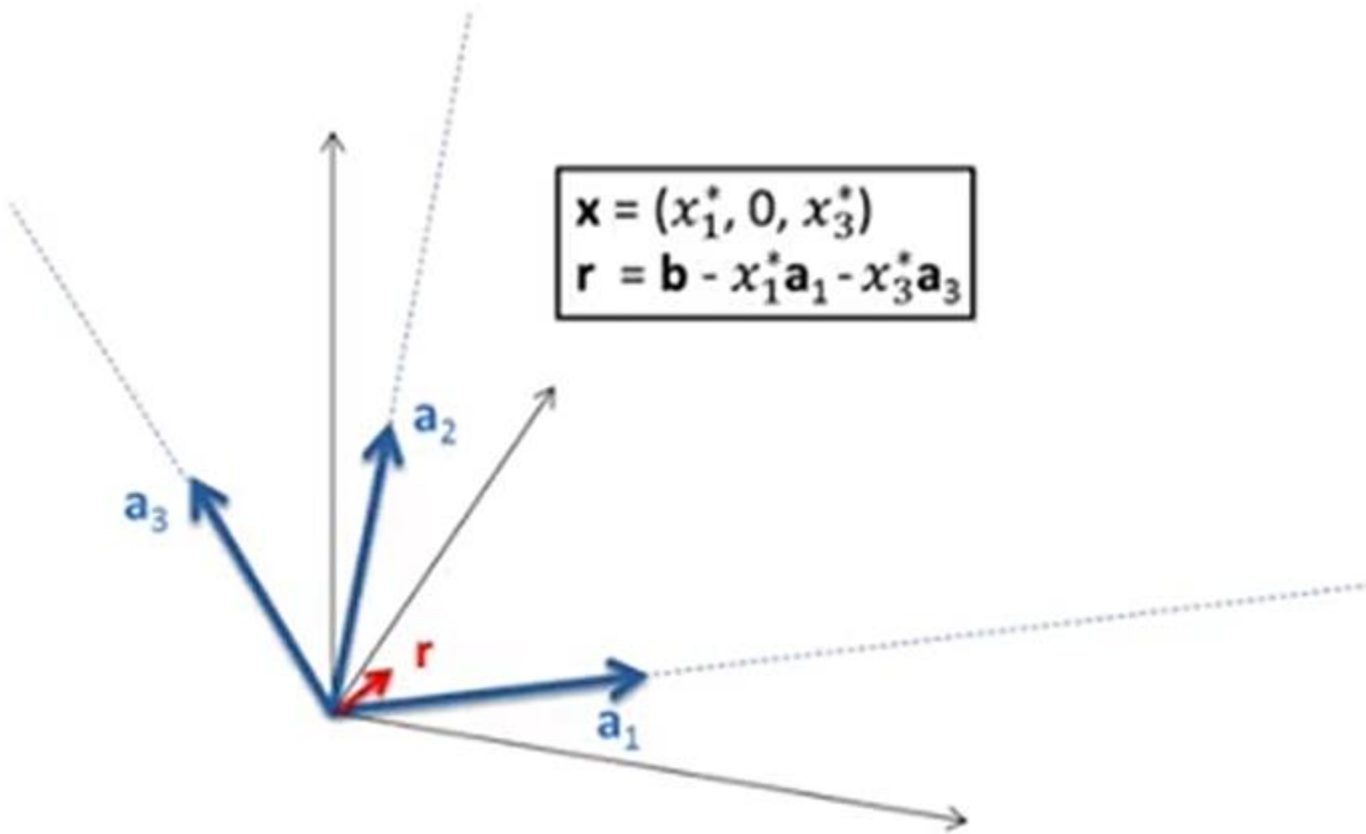
Matching Pursuit



Matching Pursuit



Matching Pursuit



Orthogonal Matching Pursuit

$$\boxed{\begin{array}{c} \min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2 \\ subject\, to\ \|\mathbf{x}\|_0 \leq S \end{array}}$$

Input: A (with unit norm columns), $\mathbf{b}$, and S.
Initialize $\mathbf{r} = \mathbf{b}$ and $\Omega = \emptyset$.
While $\|\mathbf{x}\|_0 < S$
compute $x_j = \mathbf{a}_j^T \mathbf{r}$ for all $j \notin \Omega$

$$i = \underset{j \notin \Omega}{argmax} |x_j|$$

$$\Omega \longleftarrow \Omega \cup \{i\}$$

$$\mathbf{x}_\Omega^* = \underset{\mathbf{x}}{argmin} \|A_\Omega \mathbf{x} - \mathbf{b}\|_2^2$$

$$\mathbf{r} \longleftarrow \mathbf{b} - A_\Omega \mathbf{x}_\Omega^*$$

Orthogonal Matching Pursuit

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0$$
$$\text{subject to } \|A\mathbf{x} - \mathbf{b}\|_2 \leq \epsilon$$

Input: A (with unit norm columns), b, and ϵ.

Initialize $\mathbf{r} = \mathbf{b}$ and $\Omega = \emptyset$.

While $\|\mathbf{r}\|_2^2 > \epsilon$

compute $x_j = \mathbf{a}_j^T \mathbf{r}$ for all $j \notin \Omega$

$i = \underset{j \notin \Omega}{argmax} |x_j|$

$\Omega \longleftarrow \Omega \cup \{i\}$

$\boxed{\mathbf{x}_\Omega^* = \underset{\mathbf{x}}{argmin} \|A_\Omega \mathbf{x} - \mathbf{b}\|_2^2}$

$\mathbf{r} \longleftarrow \mathbf{b} - A_\Omega \mathbf{x}_\Omega^*$

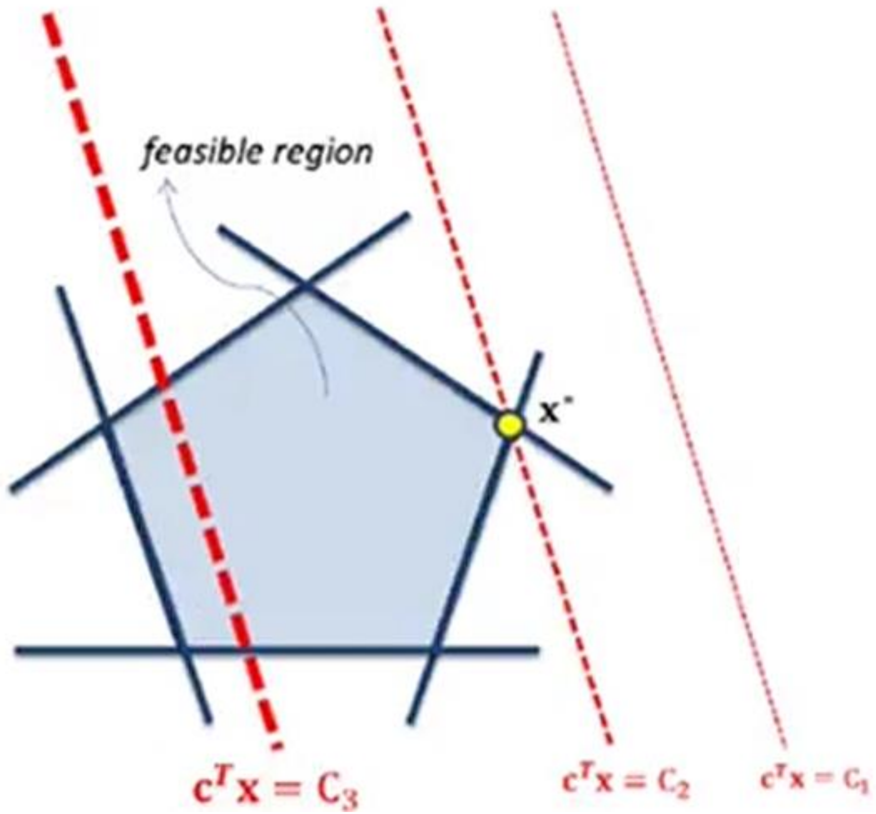
Prony's Method

- The Kruskal rank of a set of vector $\{A_i\}$ is the maximum r such that all subsets of r vectors are linearly independent
- If $\|x\|_0 \leq r/2$ then r is the unique sparsest solution to $\mathrm{Ax} = \mathrm{b}$
- Prony's Method
  - Any k-sparse signal can be recovered from just the first $2k$ values of its discrete Fourier transform
  - Compressed sensing
    - Find a w where $\|x - w\|_1 \leq C\delta_k(x)$ from a few $(\tilde{O}(k))$ measurements

Linear Programs (线性规划)

$$\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x}$$

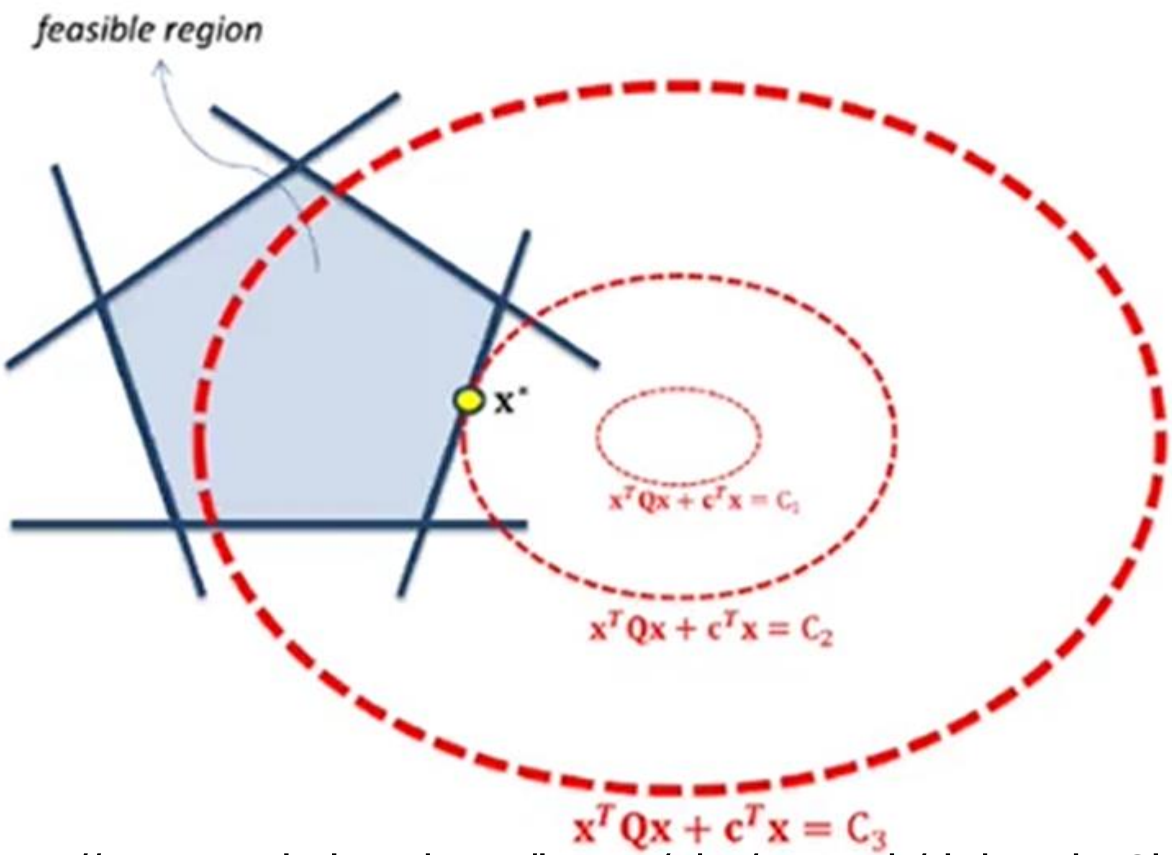
$$subject\ to\ F_i\mathbf{x} + \mathbf{g}_i \leq \mathbf{0}\ \forall i$$



LP: http://www.cad.zju.edu.cn/home/zhx/csmath/doku.php?id=2019

Quadratic Programs (二次规划)

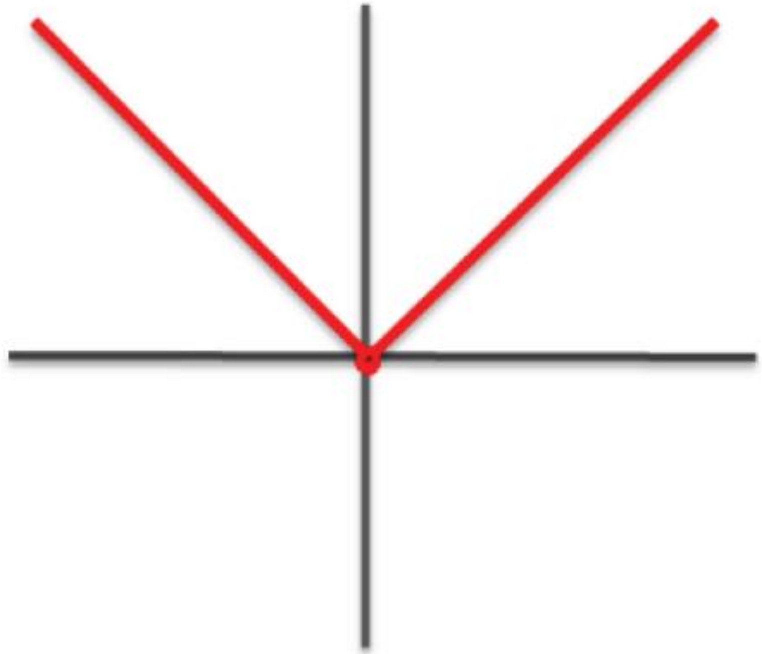
$$\min_{\mathbf{x}} \mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$$
$$subject\ to\ F_i\mathbf{x} + \mathbf{g}_i \leq 0\ \forall i$$



QP: http://www.cad.zju.edu.cn/home/zhx/csmath/doku.php?id=2019

Smooth Reformulation Tricks

- The L_1 norm is non-differentiable at the origin



- We introduce two reformulation tricks that transform sparse optimization problems into well-studied Linear programs and Quadratic programs

Positive-Negative Split Trick

$$p_i = \begin{cases} x_i & if\ x_i > 0 \\ 0 & else \end{cases} \qquad n_i = \begin{cases} -x_i & if\ x_i < 0 \\ 0 & else \end{cases}$$

$$x_i = p_i - n_i$$

$$\|\mathbf{x}\|_1 = \mathbf{1}^T(\mathbf{p} + \mathbf{n})$$

Positive-Negative Split Trick

$$\min_{x} \|x\|_1$$
$$subject\ to\ Ax = b$$

→ Linear Programs

$$x = p - n$$

$$\min_{p,n} 1^T(p+n)$$
$$s.t.\ A(p-n) = b$$
$$p, n \geq 0$$

$$Z = \begin{bmatrix} p \\ n \end{bmatrix}$$

$$\min_{p,n} 1^T Z$$
$$s.t.\ CZ = b$$
$$Z \geq 0$$

$$C = AF$$
$$F = \begin{bmatrix} I_{N\times N} & -I_{N\times N} \end{bmatrix}$$

Positive-Negative Split Trick

$$\min_{x} \|Ax - b\|_2 + \lambda\|x\|_1$$

LASSO → Quadratic Programs

$$x = p - n$$

$$\min_{p,n} \|A(p - n) - b\|_2 + \lambda 1^T p + \lambda 1^T n$$
$$s.t.\ p, n \geq 0$$

$$Z = \begin{bmatrix} p \\ n \end{bmatrix}$$

$$B = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}$$

$$\min_{Z} Z^T B Z + C^T Z$$
$$s.t.\ Z \geq 0$$

$$C = \lambda 1 + 2\begin{bmatrix} -A^T b \\ A^T b \end{bmatrix}$$

Suppression Trick

$$\min_x \|x\|_1$$
$$subject\ to\ Ax = b$$

→ Linear Programs

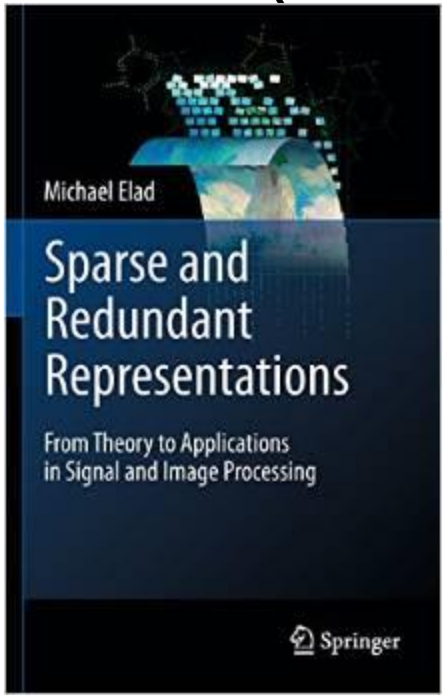
$$s, |x_k| \le s_k$$

$$\min_{x,s} 1^T s$$
$$s.t. Ax = b$$
$$|x_k| \le s_k, \forall k$$
$$s \ge 0$$

$$\min_{x,s} 1^T s$$
$$s.t. Ax = b$$
$$x_k \le s_k, \forall k$$
$$x_k \ge -s_k, \forall k$$
$$s \ge 0$$

Advanced Methods

- Stagewise OMP (StOMP), compressive sampling matching pursuit (CoSaMP)
- FISTA (Fast Iterative Shrinkage Algorithm)
- ADMM (Alternating Direction Method of Multipliers)



$$\min_{x,y} f(x) + g(y)$$
$$subject\ to\ Ax + By = c$$

Sparse and Redundant Representations and Their Applications in Signal and Image Processing
http://www.cs.technion.ac.il/~elad/teaching/courses/Sparse_Representations_Winter_2014/index.htm

Dictionary Learning

$$\min_{\mathbf{x}} \|A\mathbf{x}_1 - \mathbf{b}_1\|_2^2 \quad \text{subject to } \|\mathbf{x}_1\|_0 \le s$$

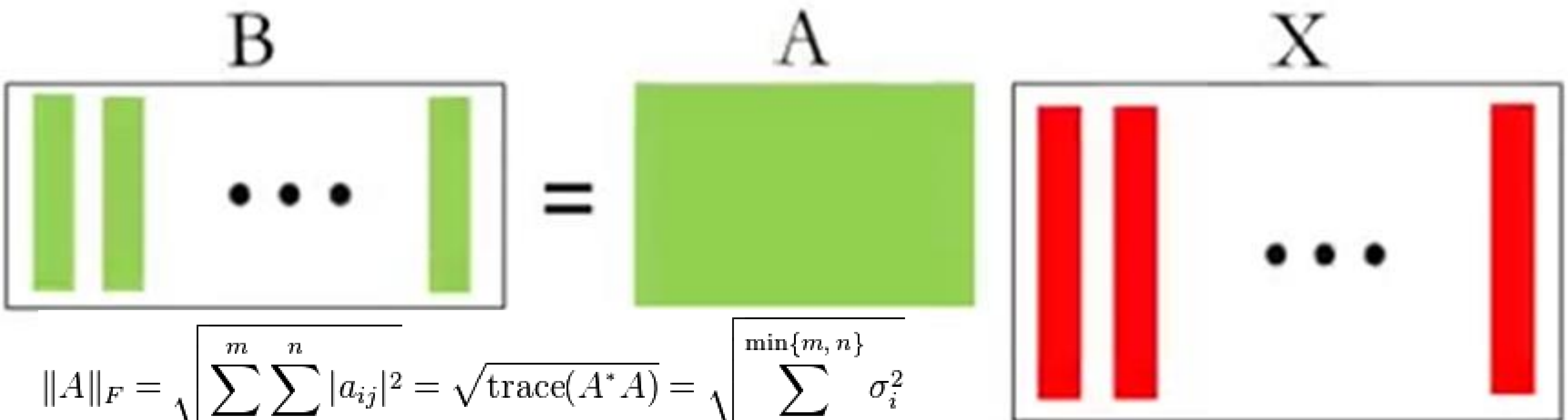
$$\vdots$$

$$\min_{\mathbf{x}} \|A\mathbf{x}_n - \mathbf{b}_n\|_2^2 \quad \text{subject to } \|\mathbf{x}_n\|_0 \le s$$

$$\min_{X} \|AX - B\|_F^2 \quad \text{subject to } \|\mathbf{x}_i\|_0 \le s \;\; 1 \le i \le n$$

What if we can choose A too?

$$\min_{A,X} \|AX - B\|_F^2 \quad \text{subject to } \|\mathbf{x}_i\|_0 \le s \;\; 1 \le i \le n$$



Frobenius norm $\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\operatorname{trace}(A^*A)} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2}$

Dictionary Learning

- Three most important cases for sparse recovery
  - A has full column rank
    - Each b_i is a linear combination of at most $\tilde{O}(\sqrt{n})$ columns in A
  - A is incoherent
    - The columns of $A \in R^{m \times n}$ are μ-incoherent if for all $i \neq j$

$$|\langle A_i, A_j \rangle| \leq \mu ||A_i|| \cdot ||A_j||$$

  - A is RIP
    - A matrix A is RIP with constant δ_k if for all k-sparse vectors x we have

$$(1 - \delta_k)||x||_2^2 \leq ||Ax||_2^2 \leq (1 + \delta_k)||x||_2^2$$

Method of Optimal Directions

$$\min_{A,X} \|AX - B\|_F^2$$
$$subject\ to\ \|x_i\|_0 \le S\ \forall i$$

Alternating minimization Similar to EM Algorithm

Keep *A* fixed; solve for *X*

$$\min_{x_i} \|Ax_i - b_i\|_2^2$$
$$subject\ to\ \|x_i\|_0 \le S$$

Keep *X* fixed; solve for *A*

$$\min_{A} \|AX - B\|_F^2$$

Least Squares:

$$A = BX^T(XX^T)^{-1}$$

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\operatorname{trace}(A^*A)} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2}$$

Frobenius norm

Bi-convex Dictionary Learning

$$\min_{A,X} \|AX - B\|_F^2 + \lambda \|X\|_1$$

Alternating minimization

Keep A fixed; solve for X

$$\min_{X} \|AX - B\|_F^2 + \lambda \|X\|_1$$

A series of LASSO problems

Keep X fixed; solve for A

$$\min_{A} \|AX - B\|_F^2$$

Least Squares:

$$A = BX^T (XX^T)^{-1}$$

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\operatorname{trace}(A^* A)} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i^2}$$

Frobenius norm

K-SVD

K-SVD [5] Start with an initial guess for A. Then repeat the following procedure:

- Given A, compute a sparse X so that $AX \approx B$ (again, using a pursuit method)
- Group all data points $B^{(1)}$ where the corresponding X vector has a non-zero at index i. Subtract off components in the other directions

$$B^{(1)} - \sum_{j \neq i} A_j X_j^{(1)}$$

- Compute the first singular vector v_1 of the residual matrix, and update the column A_i to v_1

Matrix Calculus

- Trace (迹)
  - $\text{tr}(A) = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{22} + \cdots + a_{nn}$
  - $\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B)$
  - $\text{tr}(cA) = c\,\text{tr}(A)$
  - $\text{tr}(A) = \text{tr}(A^T)$
  - $\text{tr}(X^TY) = \text{tr}(XY^T) = \text{tr}(Y^TX) = \text{tr}(YX^T)$
- $\min_{A} \|AX - B\|_F^2$
  - $\|M\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} {m_{ij}}^2$
  - $\|M\|_F^2 = tr(MM^T) = tr(M^TM)$
  - $\|AX - B\|_F^2 = tr((AX - B)^T(AX - B))$
  - $\|AX - B\|_F^2 = tr(X^TA^TAX - X^TA^TB - B^TAX + B^TB)$

Matrix Calculus

- Trace
  - tr(ABCD) = tr(BCDA) = tr(CDAB) = tr(DABC)
  - d tr(X) = tr(dX)
- $\min_{A} \|AX - B\|_F^2$
  - $\|M\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} {m_{ij}}^2 = tr(MM^T) = tr(M^T M)$
  - $\|AX - B\|_F^2 = tr((AX - B)^T (AX - B))$
  - $\|AX - B\|_F^2 = tr(\mathrm{X}^T \mathrm{A}^T AX - \mathrm{X}^T \mathrm{A}^T B - B^T AX + \mathrm{B}^T B)$
  - $\frac{\partial}{\partial A} \|AX - B\|_F^2 = \frac{\partial}{\partial A} tr(AX\mathrm{X}^T \mathrm{A}^T - B\mathrm{X}^T \mathrm{A}^T - XB^T A + \mathrm{B}^T B)$

Matrix Calculus

- $D_X tr(A^T X) = D_X tr(AX^T) = A$
  - $\frac{\partial tr(\mathrm{A}^{\mathrm{T}} X)}{\partial x_{ij}} = \frac{\partial \sum_{ij} a_{ij} x_{ij}}{\partial x_{ij}} = a_{ij}$
- $D_X tr(AX) = D_X tr(XA) = A^T$
- $D_X tr(XAX^T B) = B^T X A^T + BXA$
- $\min_A \|AX - B\|_F^2$
  - $\frac{\partial}{\partial A} \|AX - B\|_F^2 = \frac{\partial}{\partial A} tr(AX\mathrm{X}^T \mathrm{A}^T - B\mathrm{X}^T \mathrm{A}^T - XB^T A + \mathrm{B}^T B)$
  - $\frac{\partial}{\partial A} \|AX - B\|_F^2 = AXX^T + AXX^T - B\mathrm{X}^T - B\mathrm{X}^T$
  - $\frac{\partial}{\partial A} \|AX - B\|_F^2 = 2AXX^T - 2B\mathrm{X}^T = 0$
  - $A = BX^T (XX^T)^{-1}$

Matrix Calculus

Chain Rule

$$\frac{dy}{dx} = \frac{dy}{dz} \bullet \frac{dz}{dx}$$

- $\nabla \|Ax - b\|_2^2 = A^T \nabla_{Ax-b} \|Ax - b\|_2^2$
- $= A^T 2(Ax - b)$
- $= 2A^T (Ax - b)$

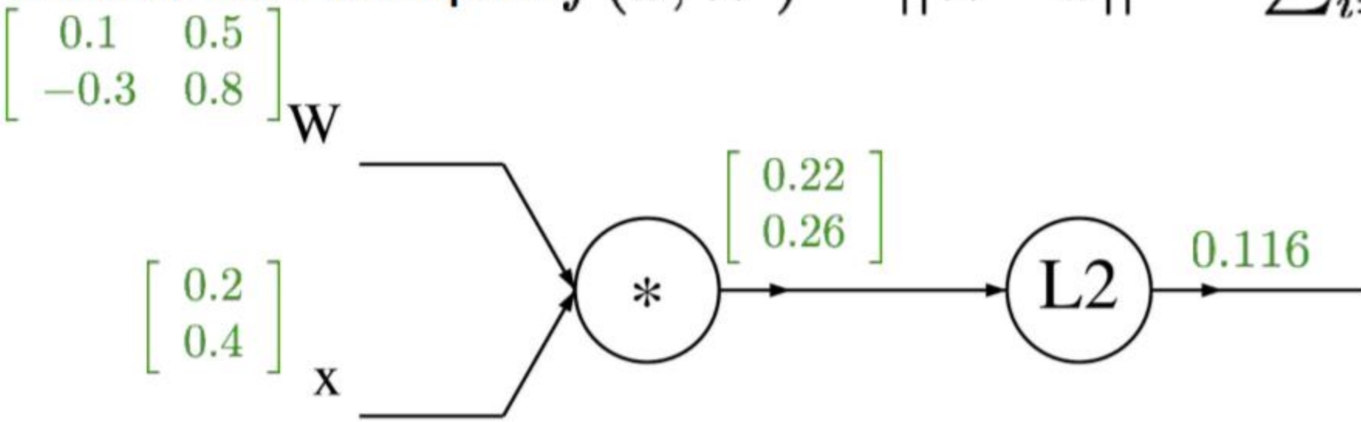
- $\nabla \|AX - B\|_F^2 = \nabla \|X^T A^T - B^T\|_F^2$
- $= (\nabla_{A^T} \|X^T A^T - B^T\|_F^2)^T$
- $= X(\nabla_{X^T A^T - B^T} \|X^T A^T - B^T\|_F^2)^T$
- $= (X2(X^T A^T - B^T))^T$
- $= 2(AX - B)X^T$
- $A = BX^T (XX^T)^{-1}$

$$\|A\|_F^2 = \|A^T\|_F^2$$

$$\nabla_X \|AX - B\|_F^2 = (\nabla_{X^T} \|AX - B\|_F^2)^T$$

Computational Graph

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^{n} (W \cdot x)_i^2$

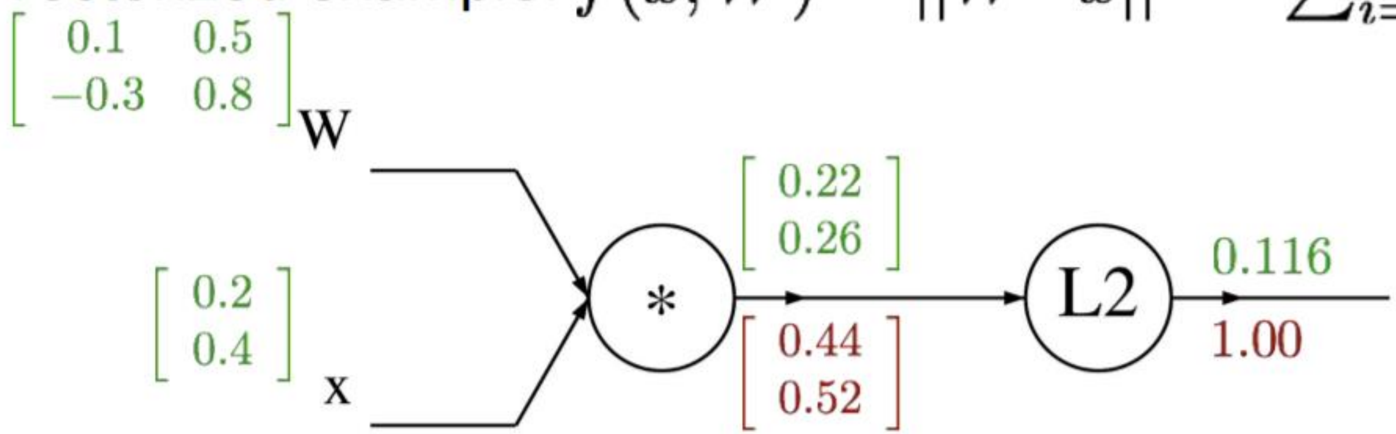


$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

Computational Graph

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^{n} (W \cdot x)_i^2$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

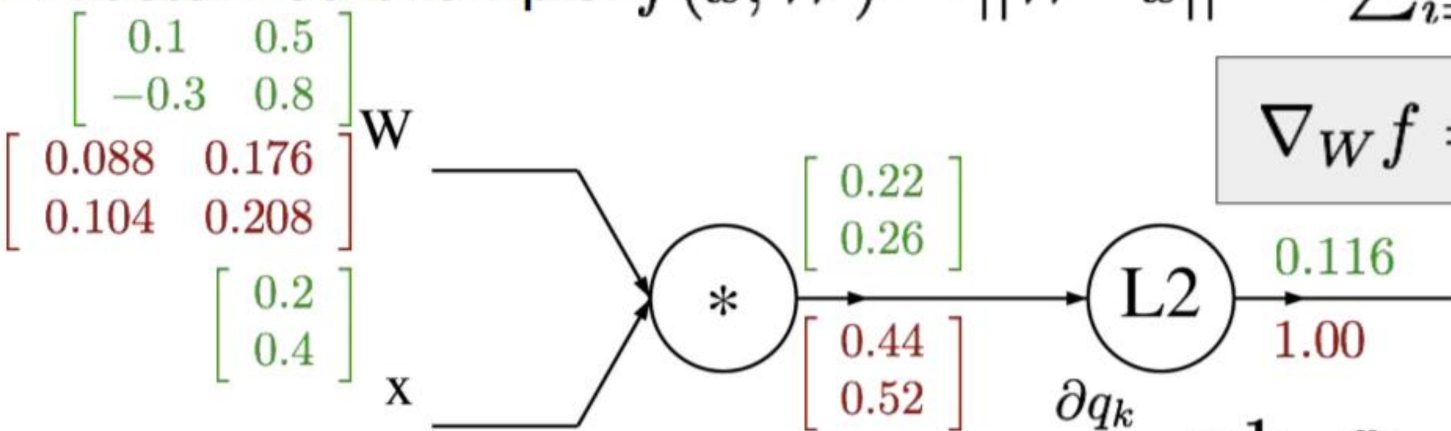
$$\frac{\partial f}{\partial q_i} = 2q_i$$

$$\boxed{\nabla_q f = 2q}$$

Computational Graph

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^{n} (W \cdot x)_i^2$

$$\boxed{\nabla_W f = 2q \cdot x^T}$$



$$\frac{\partial q_k}{\partial W_{i,j}} = \mathbf{1}_{k=i} x_j$$

$$\begin{aligned}
\frac{\partial f}{\partial W_{i,j}} &= \sum_k \frac{\partial f}{\partial q_k} \frac{\partial q_k}{\partial W_{i,j}} \\
&= \sum_k (2q_k)(\mathbf{1}_{k=i} x_j) \\
&= 2q_i x_j
\end{aligned}$$

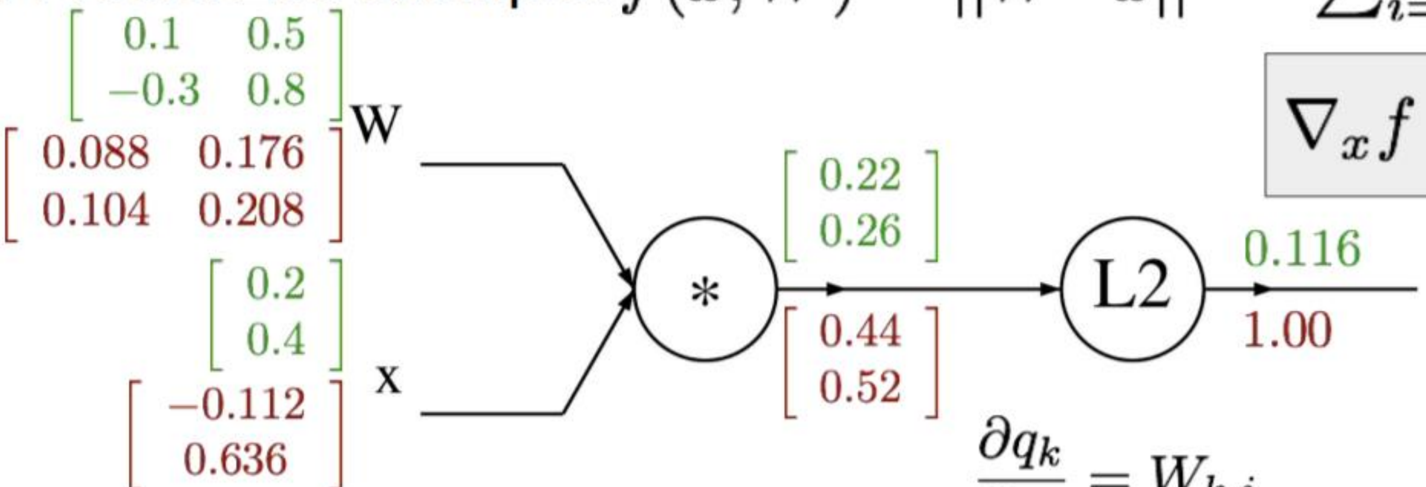
$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

Computational Graph

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^{n} (W \cdot x)_i^2$

$$\boxed{\nabla_x f = 2W^T \cdot q}$$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

$$\frac{\partial q_k}{\partial x_i} = W_{k,i}$$

$$\frac{\partial f}{\partial x_i} = \sum_k \frac{\partial f}{\partial q_k} \frac{\partial q_k}{\partial x_i} = \sum_k 2q_k W_{k,i}$$

Computational Graph

In discussion section: A matrix example...

$$z_1 = XW_1$$

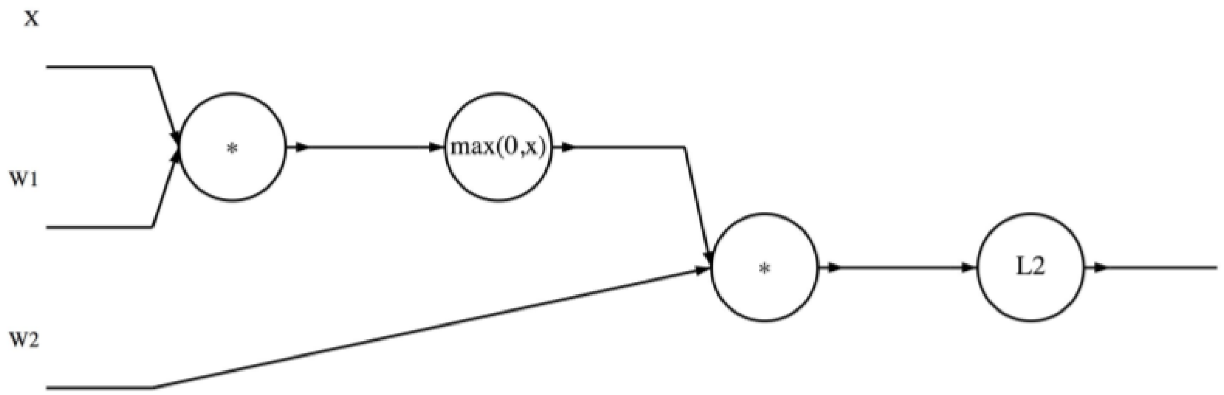
$$h_1 = \mathrm{ReLU}(z_1)$$

$$\hat{y} = h_1 W_2$$

$$L = ||\hat{y}||_2^2$$

$$\frac{\partial L}{\partial W_2} = \textbf{?}$$

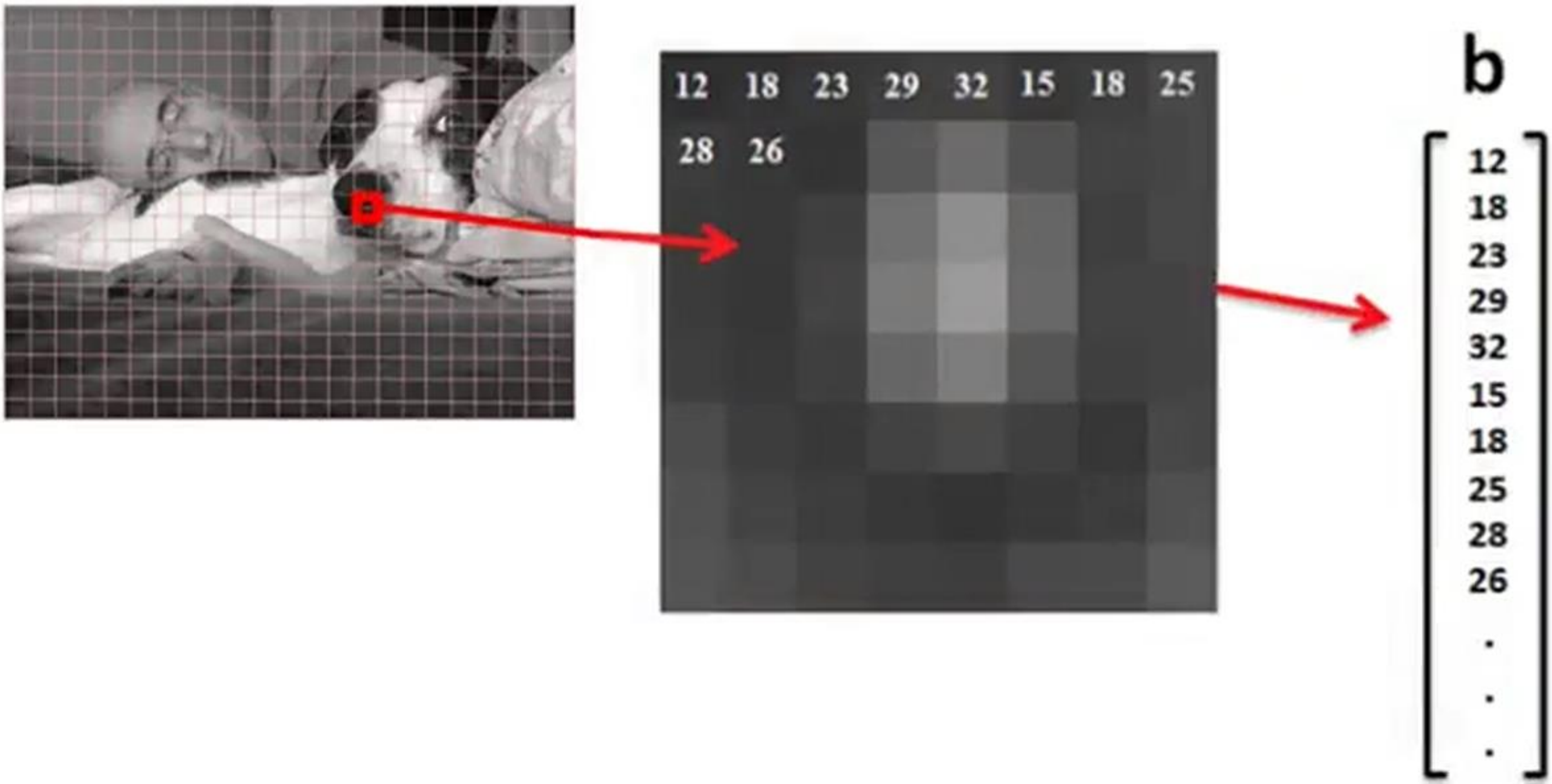
$$\frac{\partial L}{\partial W_1} = \textbf{?}$$



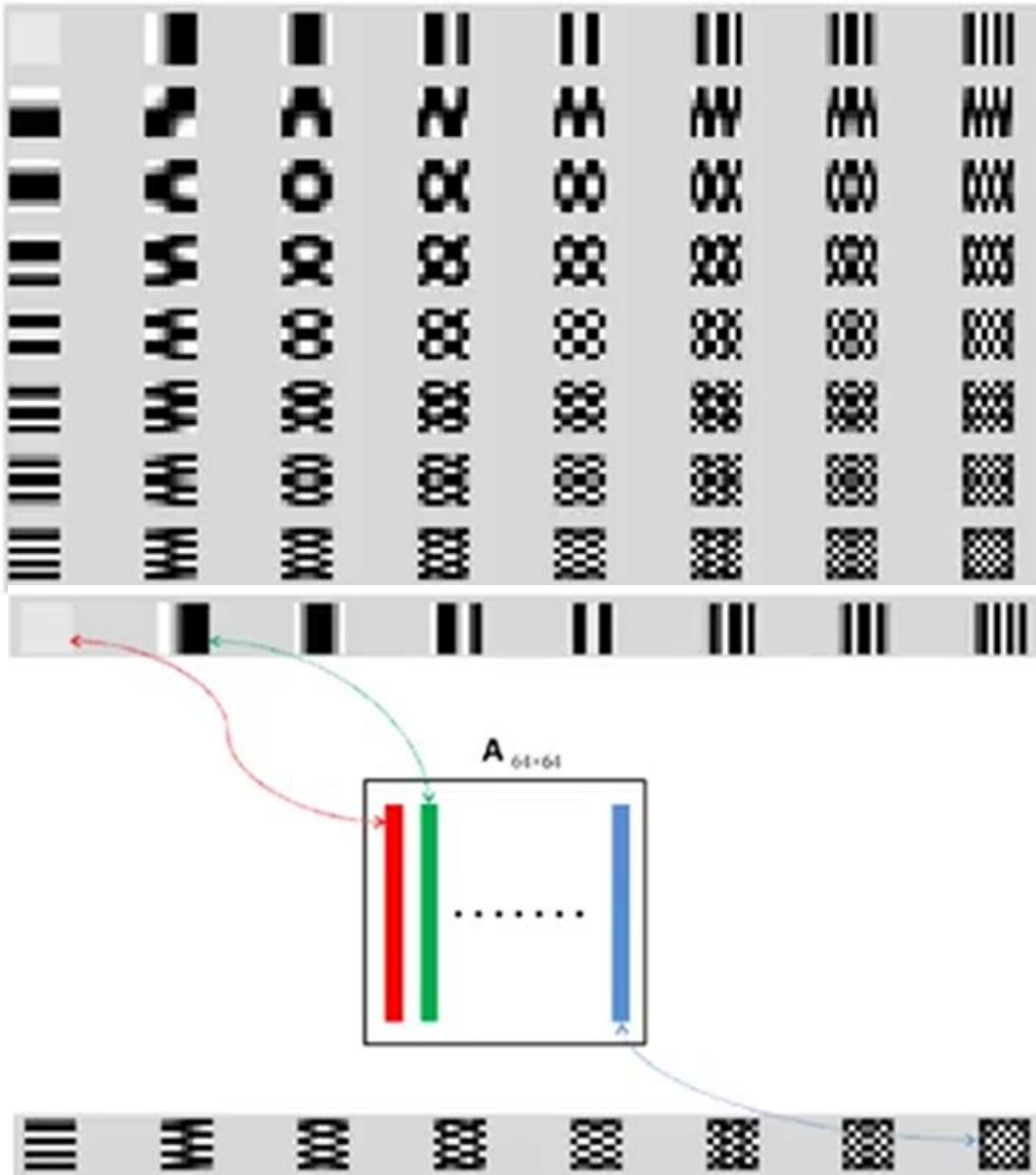
Outline

- Examples of Applications of Sparsity
- L_2, L_1, and L_0 Norms
  - Linear Inverse Problems
  - Minimum L_2, L_1, and L_0 Norm Solution
- Solution Approaches
  - Matching Pursuit
  - Smooth Reformulations
  - Dictionary Learning
- Sparse Solutions to Some Applications
  - Image Denoising, Image Inpainting, Image Super-Resolution, Robust Face Recognition, Video Surveillance, Compressive Sensing

Forming b



Forming A: DCT Dictionary



Experiment

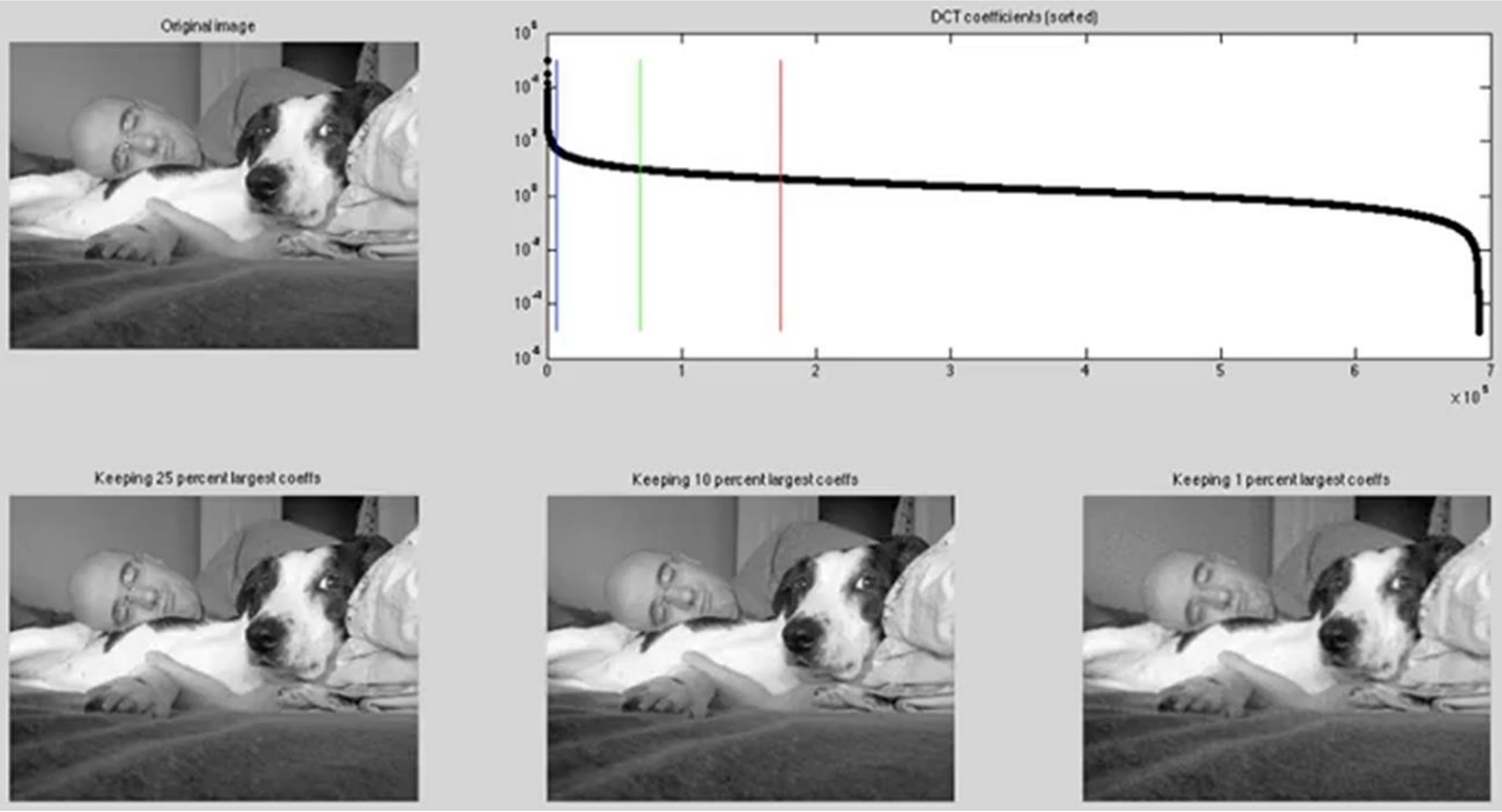

Original image
DCT coefficients (sorted)
Keeping 25 percent largest coeffs
Keeping 10 percent largest coeffs
Keeping 1 percent largest coeffs

Image Denoising

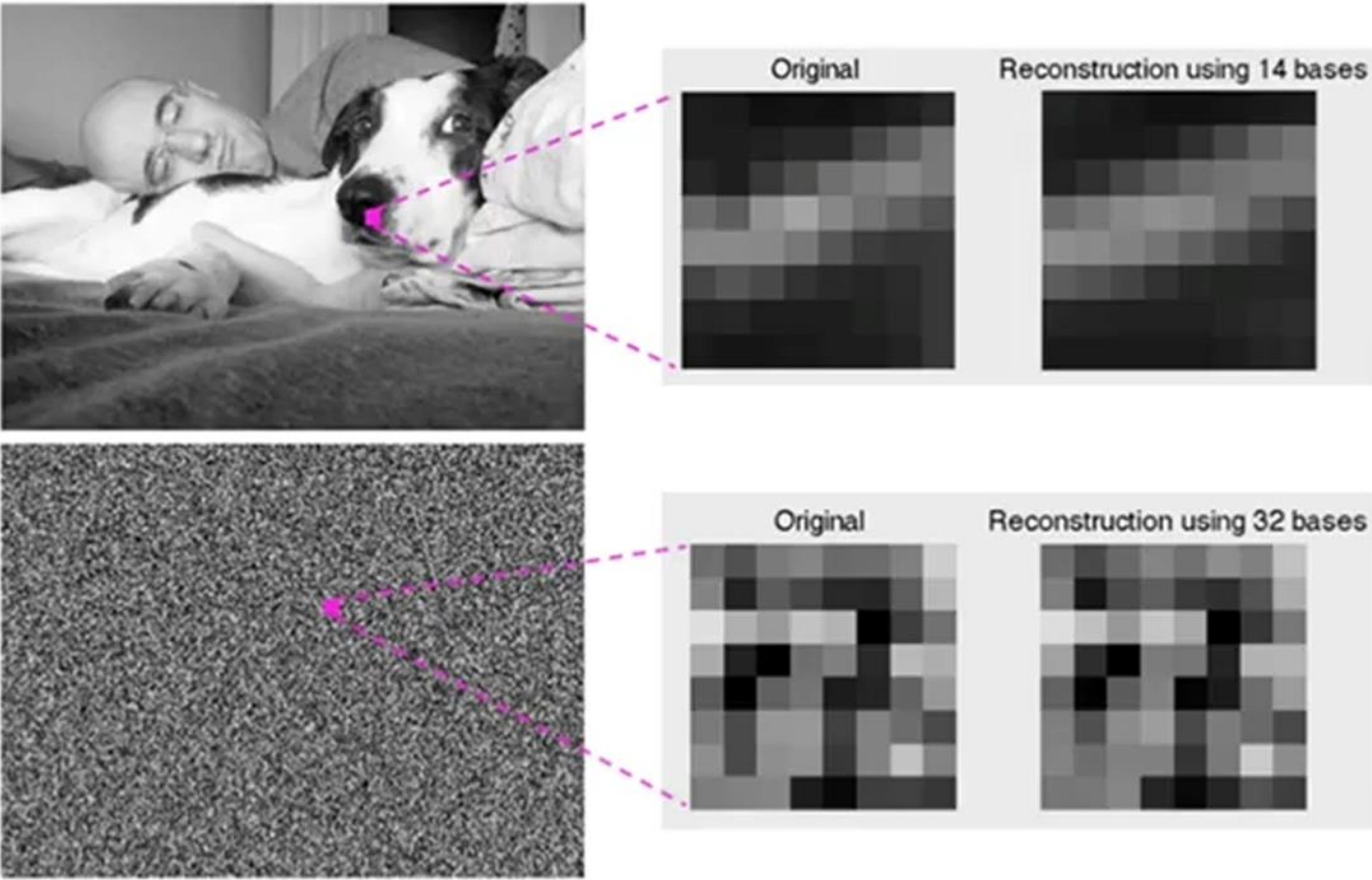


Image Denoising

$$\min_{A,X} \|AX - B\|_F^2 + \lambda \|X\|_1$$

- A Dictionary
- B Input noisy image
- AX^* Recovered image

Image Denoising

PSNR = 22.1 dB

PSNR = 33.4 dB

Image Inpainting

$$\min_{X} \|RAX - B\|_F^2 + \lambda \|X\|_1$$

- R Degradation matrix
- B Input image with missing pixels
- AX^* Recovered image

Image Inpainting

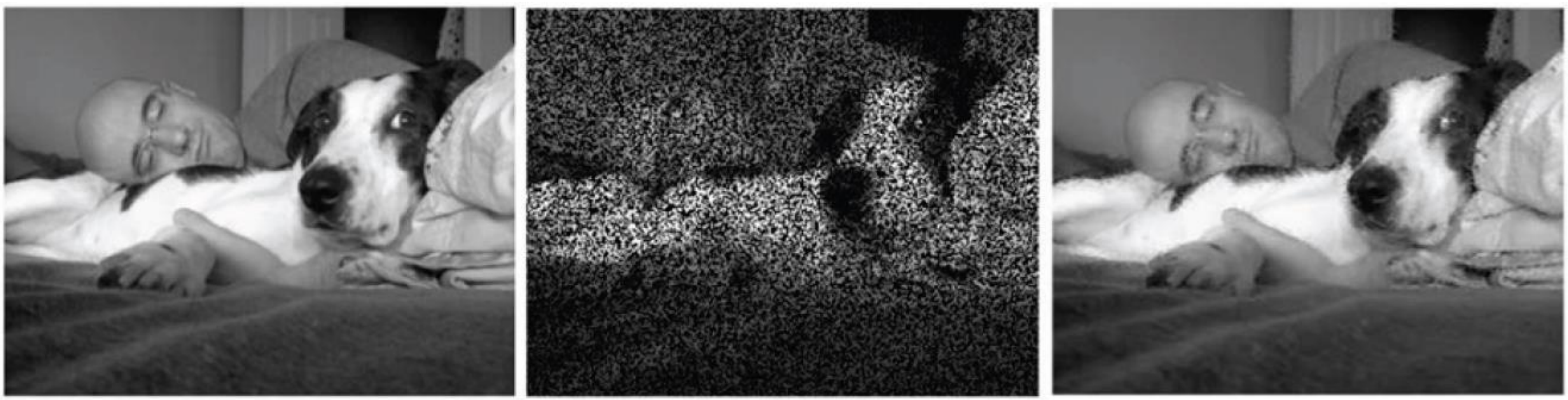
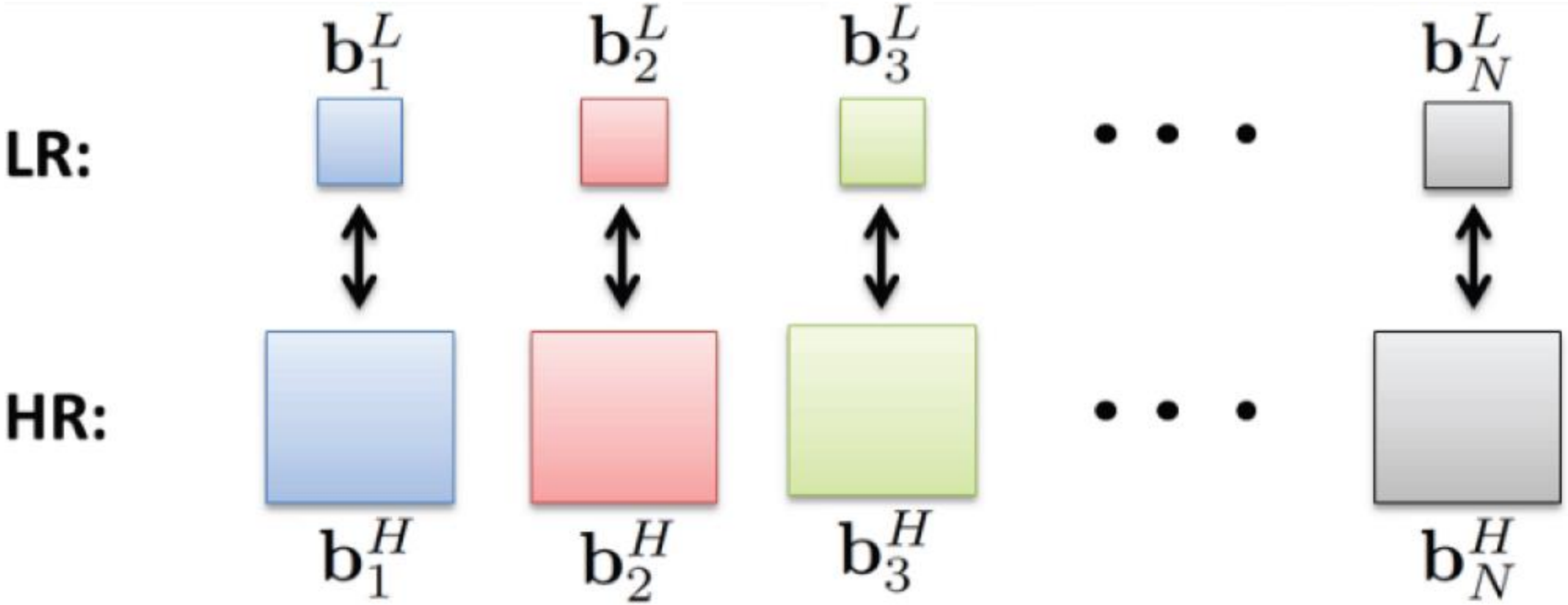


Image Super-Resolution



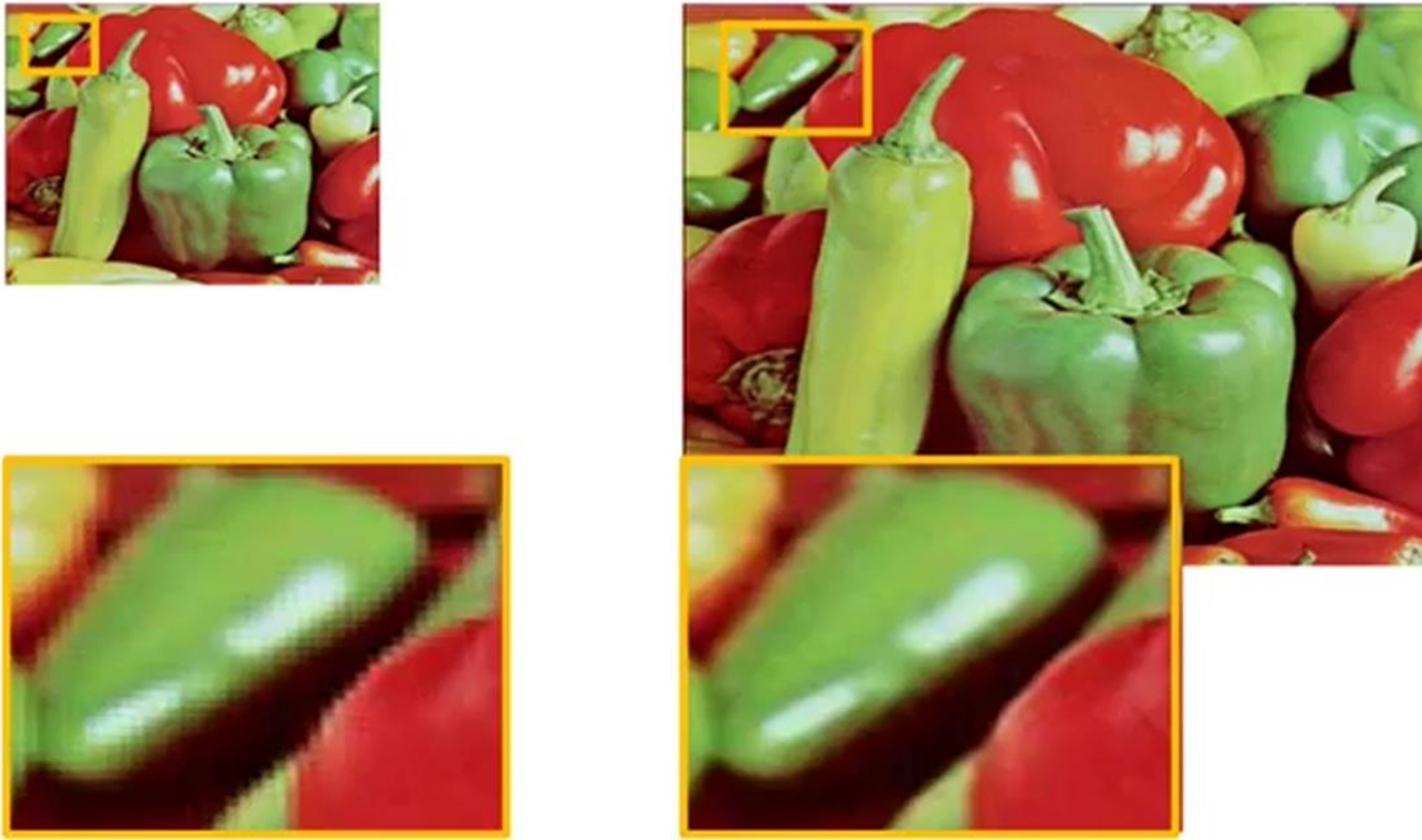
- Training Phase:

$$\min_{A^L, A^H, X} \|A^L X - B^L\|_F^2 + \mu \|A^H X - B^H\|_F^2 + \lambda \|X\|_1$$

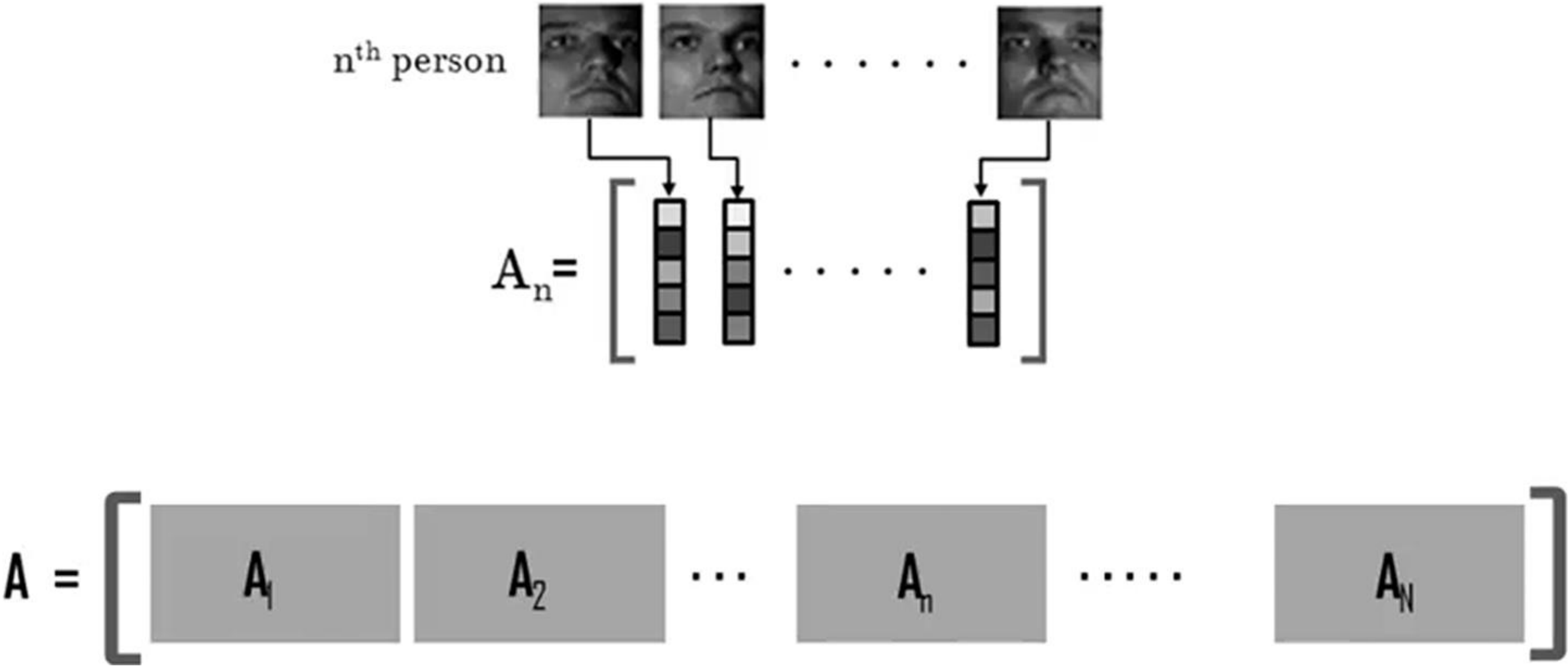
- Reconstruction Phase: $A^H X^*$ super-resolved image

$$X^* = \underset{X}{arg\min} \|A^L X - B^{new}\|_F^2 + \lambda \|X\|_1$$

Image Super-Resolution



Robust Face Recognition



Robust Face Recognition

$$\min_{\mathbf{x},\mathbf{e}} \ \|\mathbf{x}\|_1 + \lambda\|\mathbf{e}\|_1$$
$$\text{subject to } A\mathbf{x} + \mathbf{e} = \mathbf{b}$$

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \lambda\mathbf{e} \end{bmatrix}$$

$$F = [A \ \tfrac{1}{\lambda}I]$$

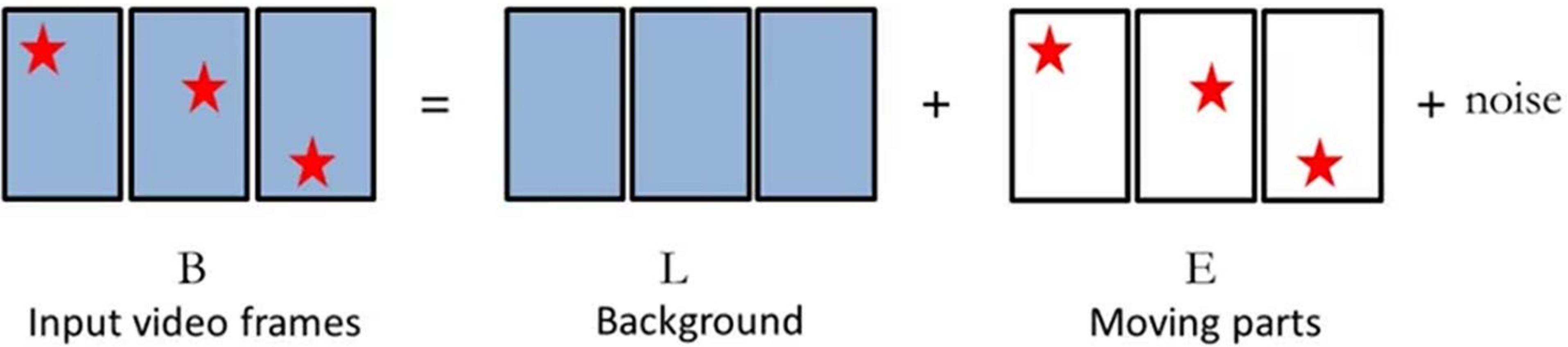
$$\min_{\mathbf{z}} \ \|\mathbf{z}\|_1$$
$$\text{subject to } F\mathbf{z} = \mathbf{b}$$

Basis pursuit

Robust Face Recognition

Video Surveillance

- How to model background L: Low rank
- Moving parts: sparse $\|E\|_0$ or $\|E\|_1$



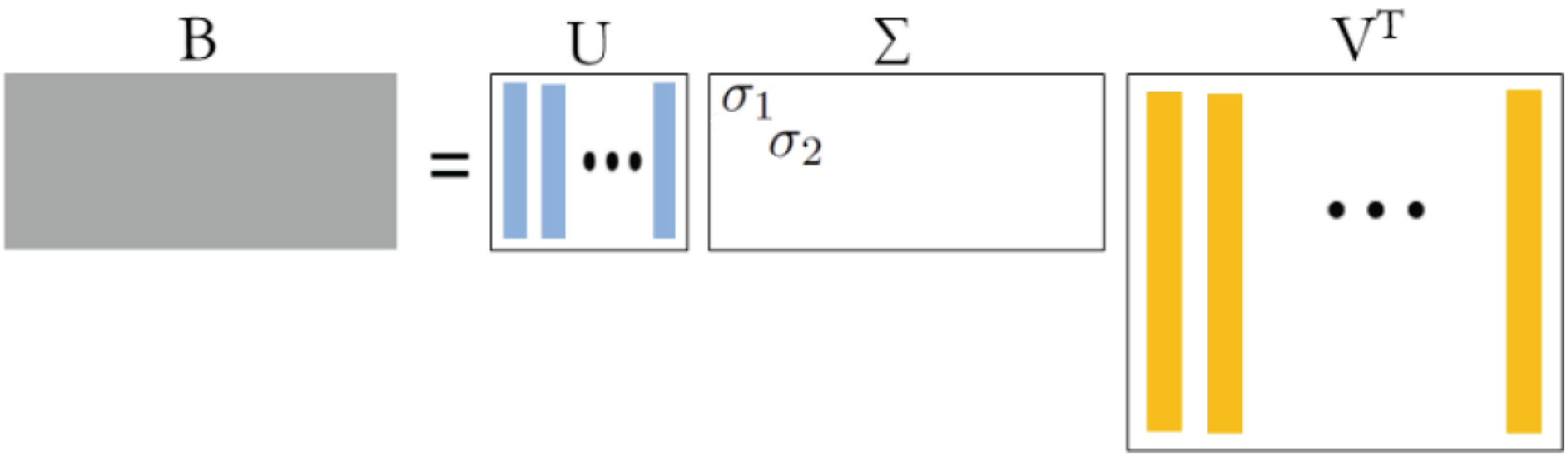
Singular Value Decomposition

$$\min_{L} \|B - L\|_F$$

$$subject\ to\ rank(L) \leq k$$

$$B = \sum_{i=0}^{r} u_i \sigma_i v_i^T$$

$$L = \sum_{i=0}^{k} u_i \sigma_i v_i^T$$



Video Surveillance

$$\min_{L,E} \|B - L - E\|_F^2 + \lambda \|E\|_1$$
$$subject\, to\, rank(L) \leq k$$

$$L = U\Sigma V^T$$

$$L = AX$$

$$\min_{L,E} \|B - L - E\|_F^2 + \lambda \|E\|_1$$
$$subject\, to\, \|\Sigma\|_0 \leq k$$

$$\min_{A,X,E} \|B - AX - E\|_F^2 + \lambda \|E\|_1$$

$$\min_{L,E} \|B - L - E\|_F^2 + \lambda \|E\|_1 + \mu \|L\|_*$$

nuclear norm

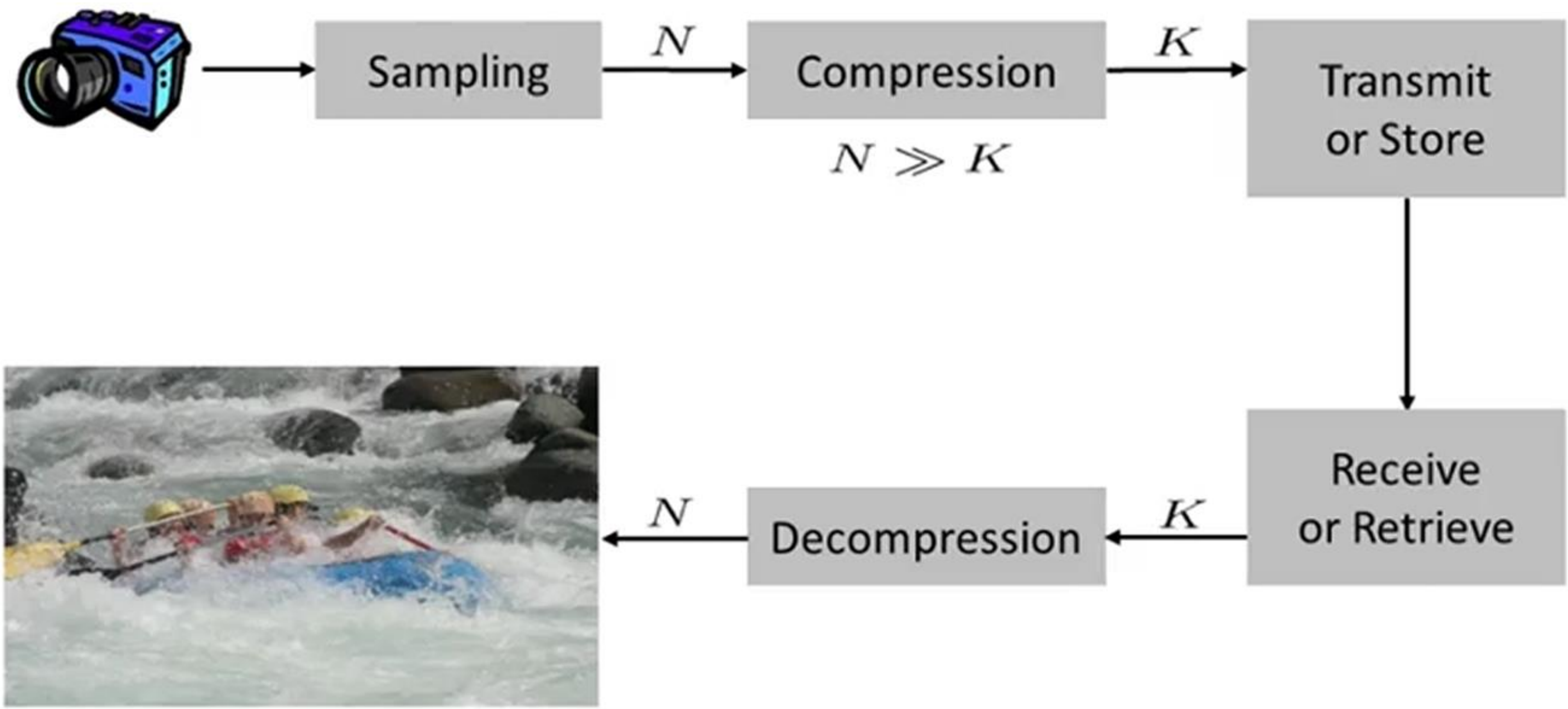
Video Surveillance

B

L

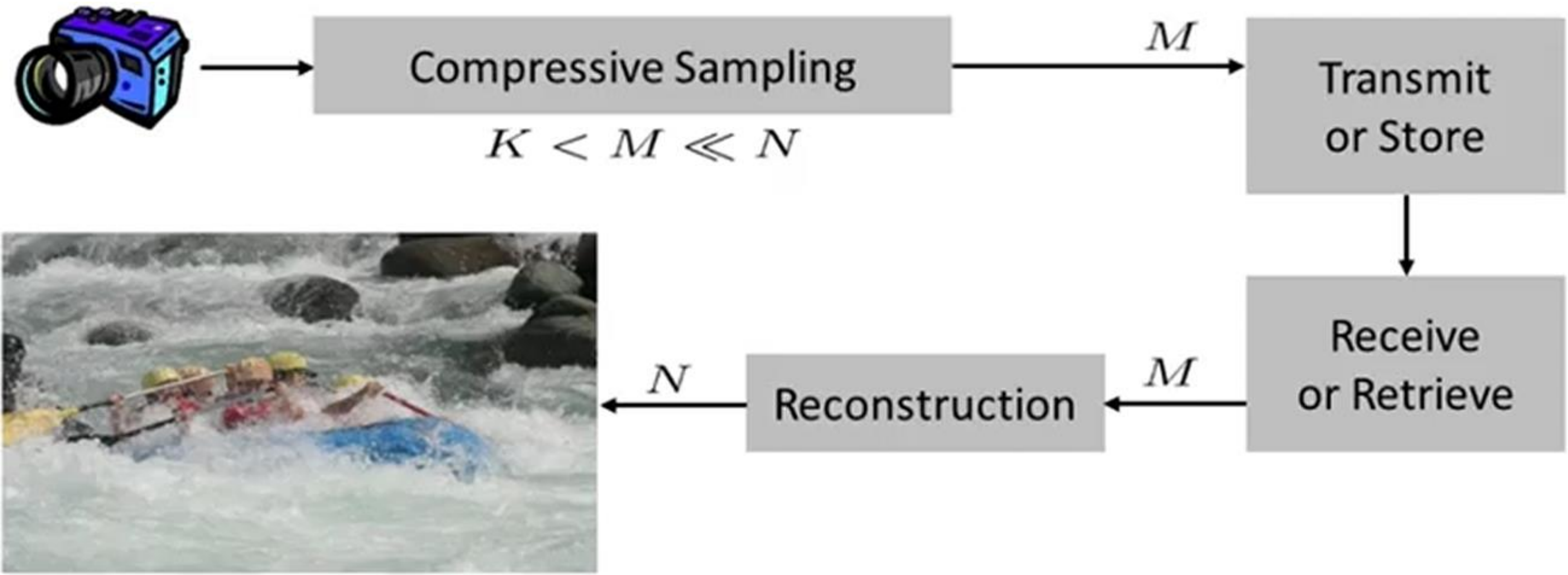
E

Sensing by Sampling



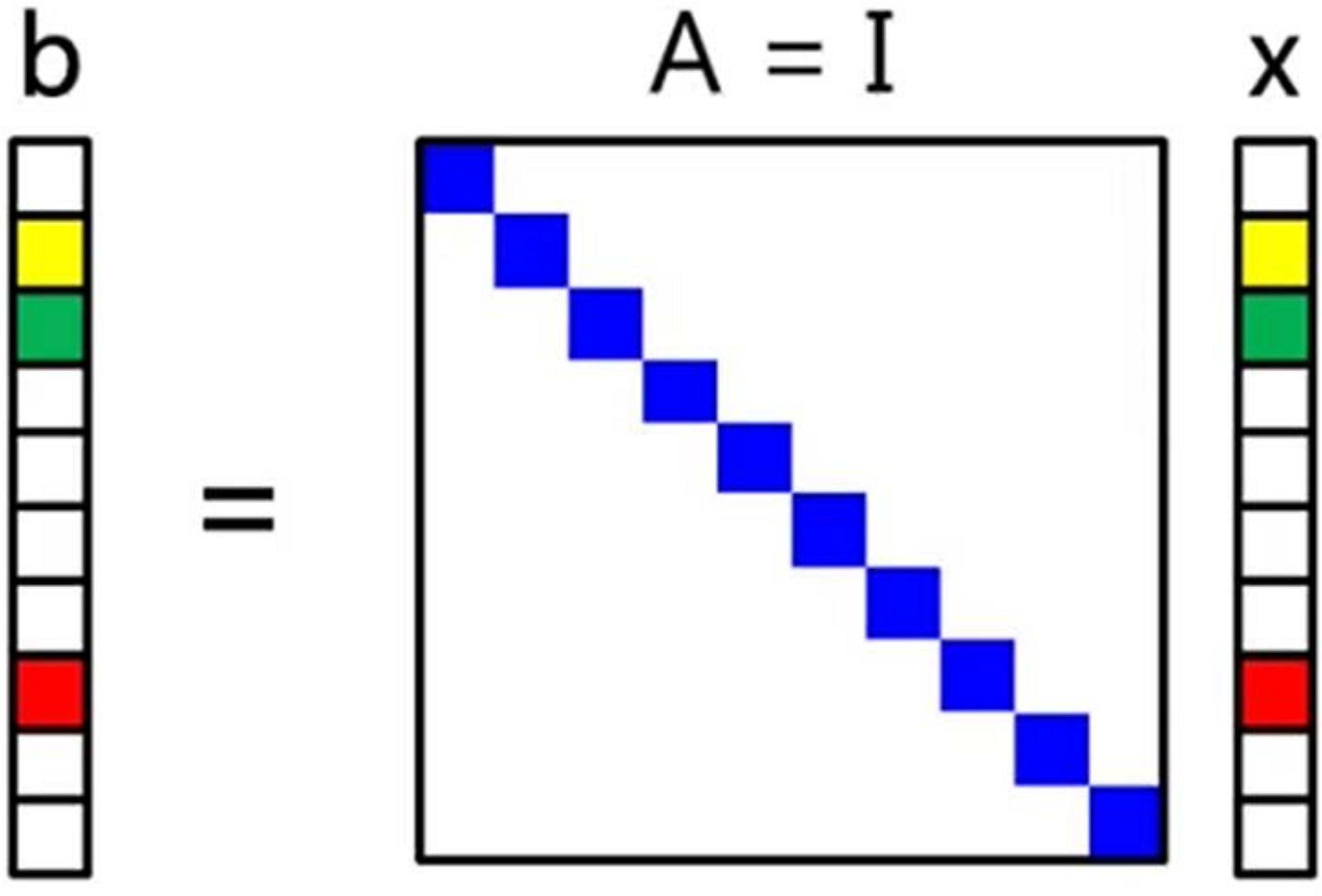
Compressive Sensing

- Directly acquire “compressed” data
- Replace samples by more general “measurements”



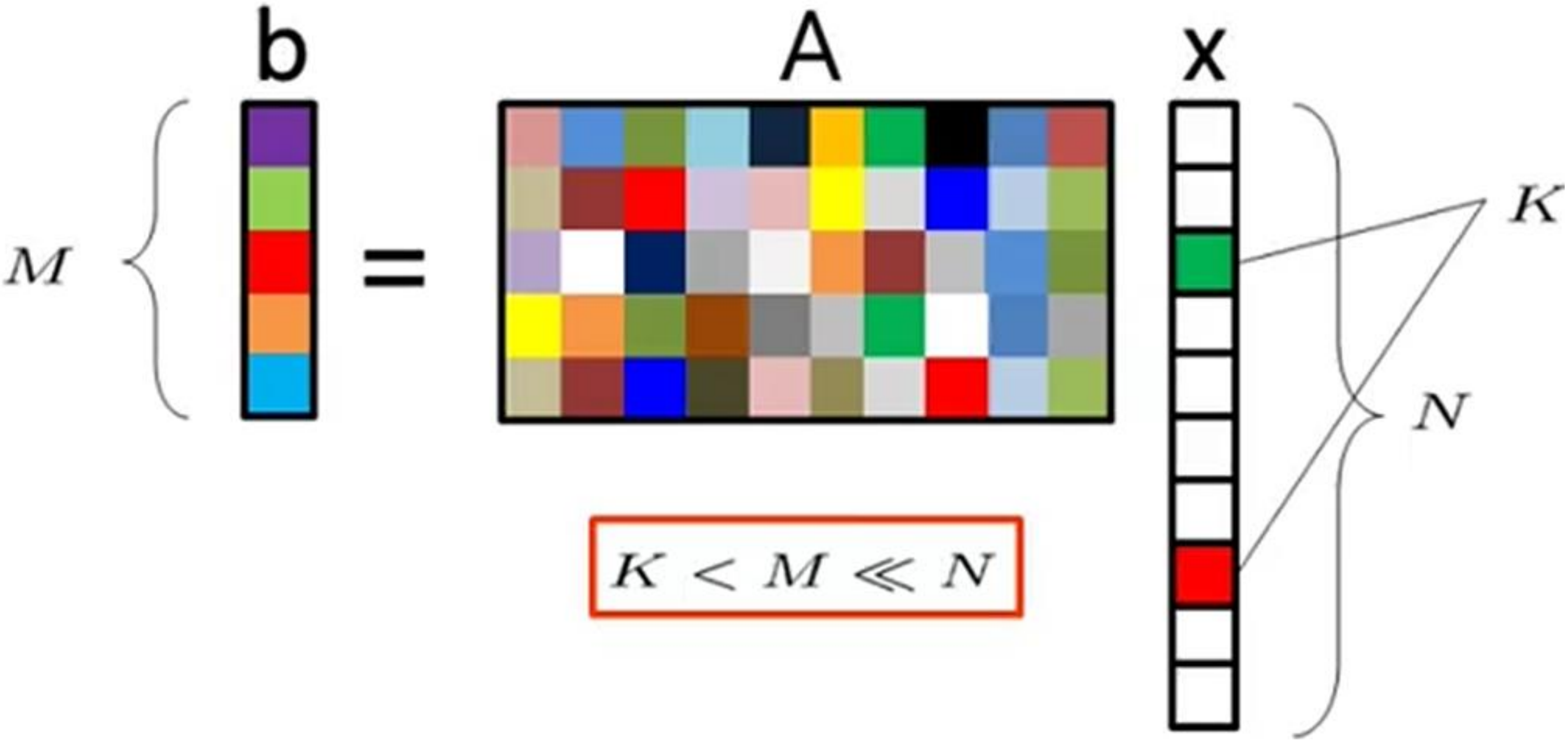
Sampling

- Signal X is K-sparse in basis/dictionary A
  - WLOG assume sparse in space domain

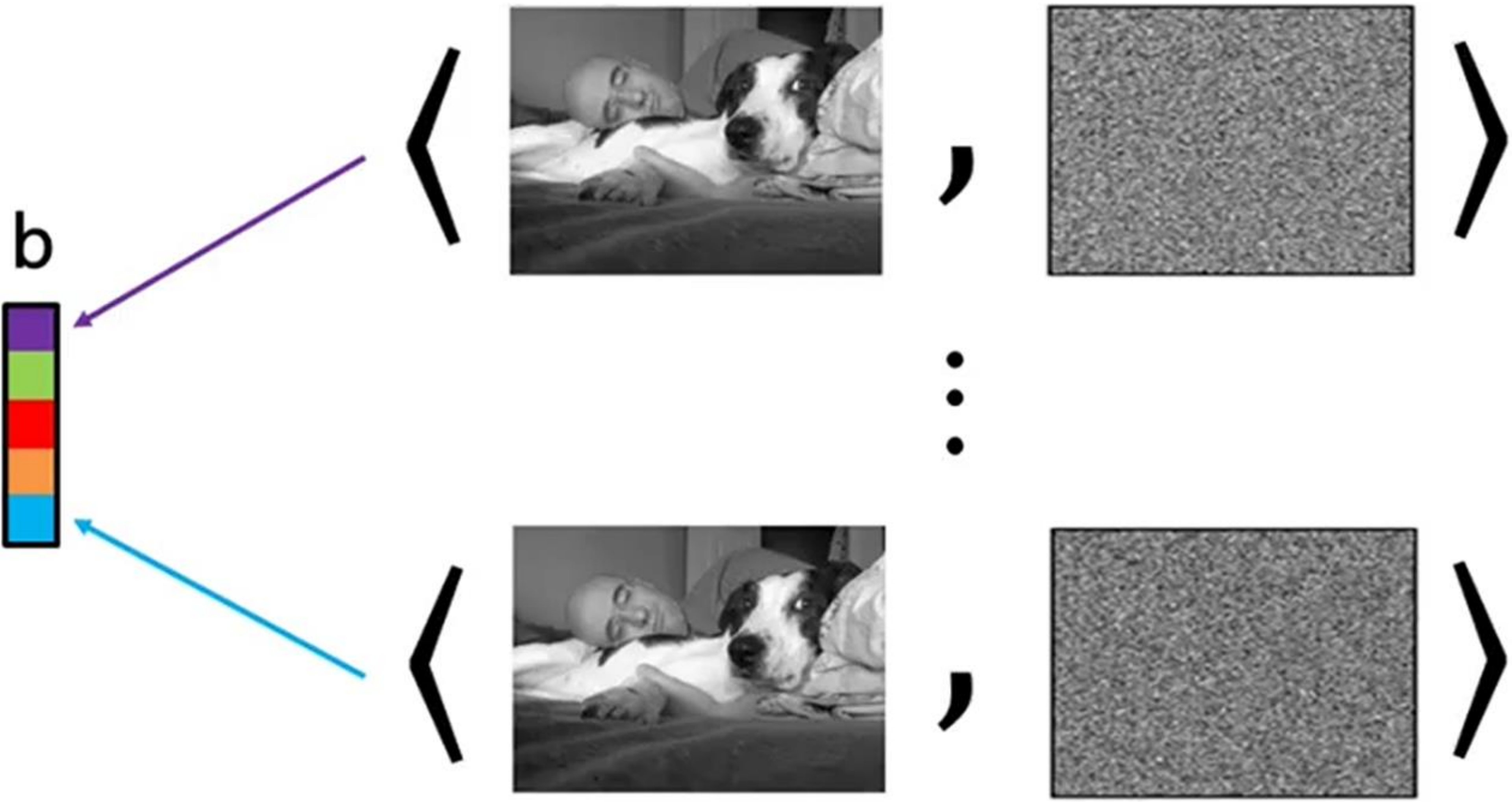


Compressive Data Acquisition

- When data is sparse/compressible, can directly acquire a condensed representation with no/little information loss through dimensionality reduction

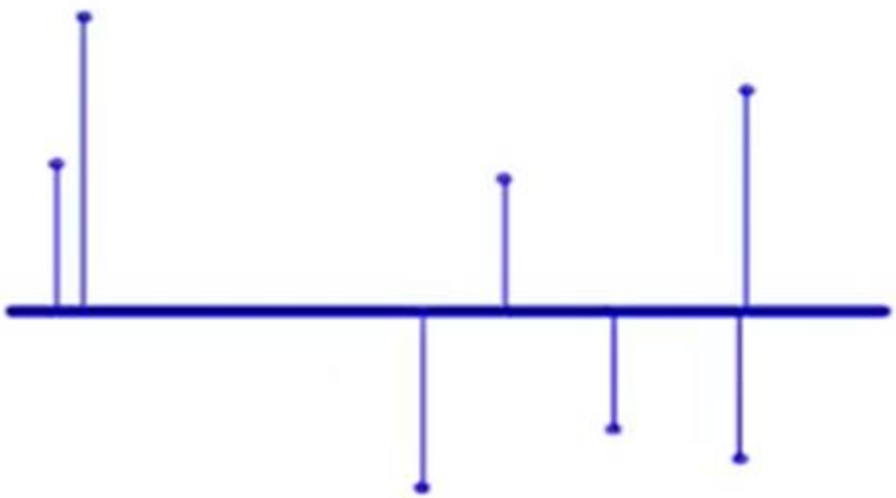


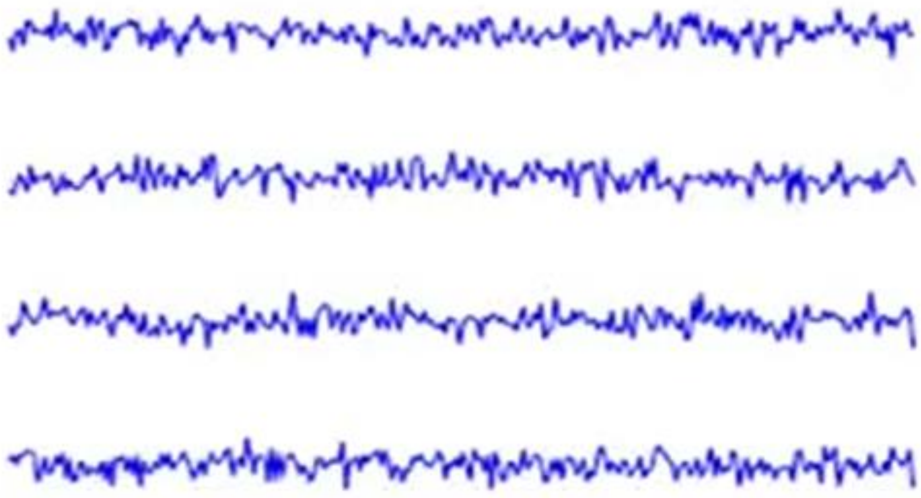
Sampling Matrices



Intuition

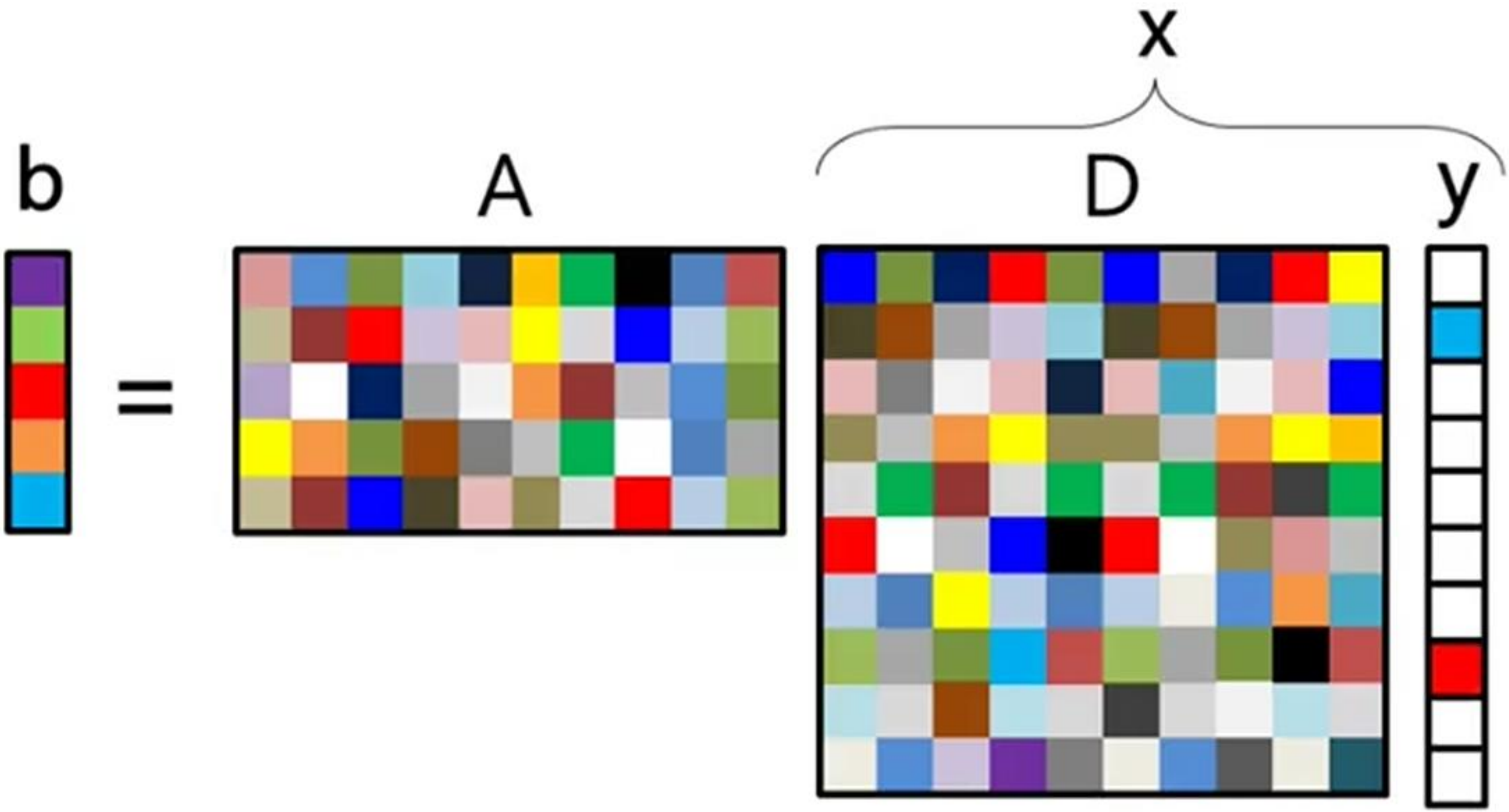
- Signal is local, measurements are global
- Each measurement picks up a little information about each component



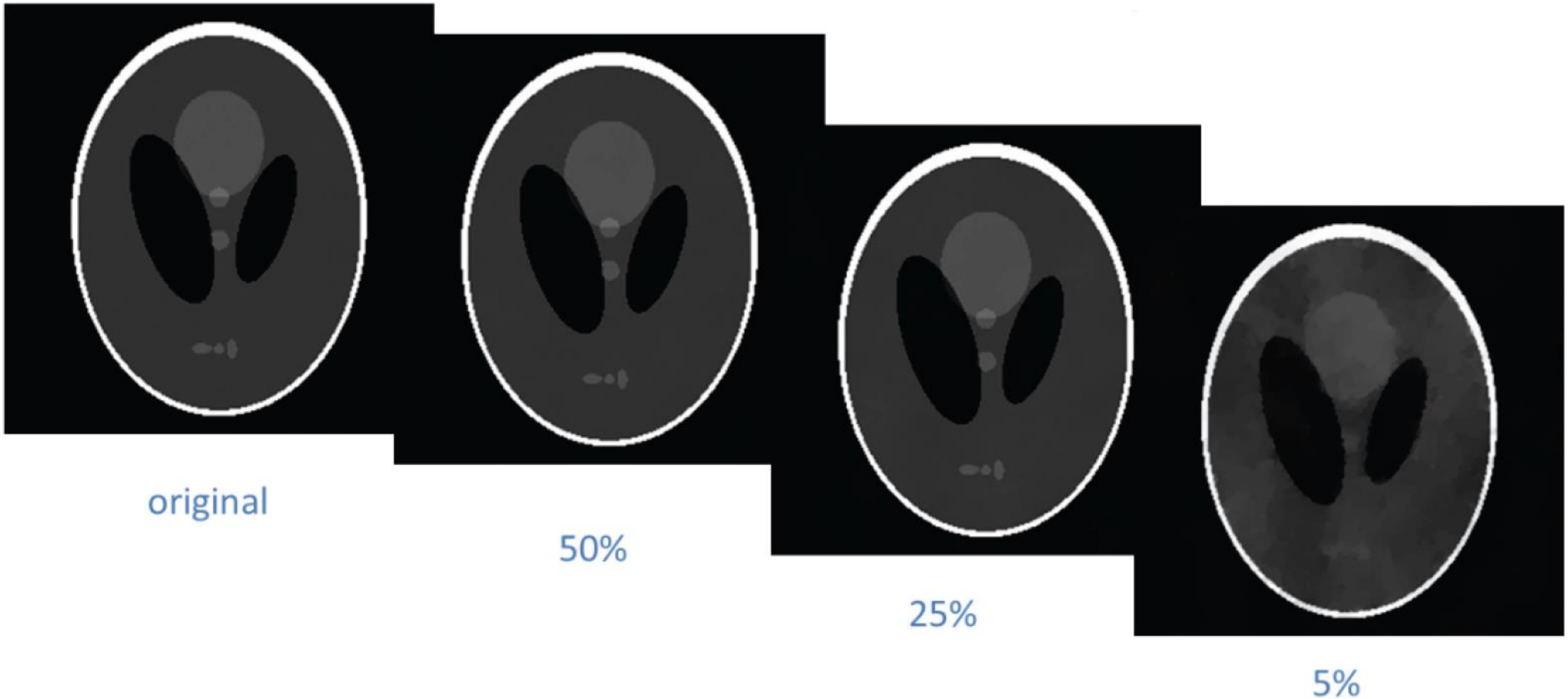


Universality

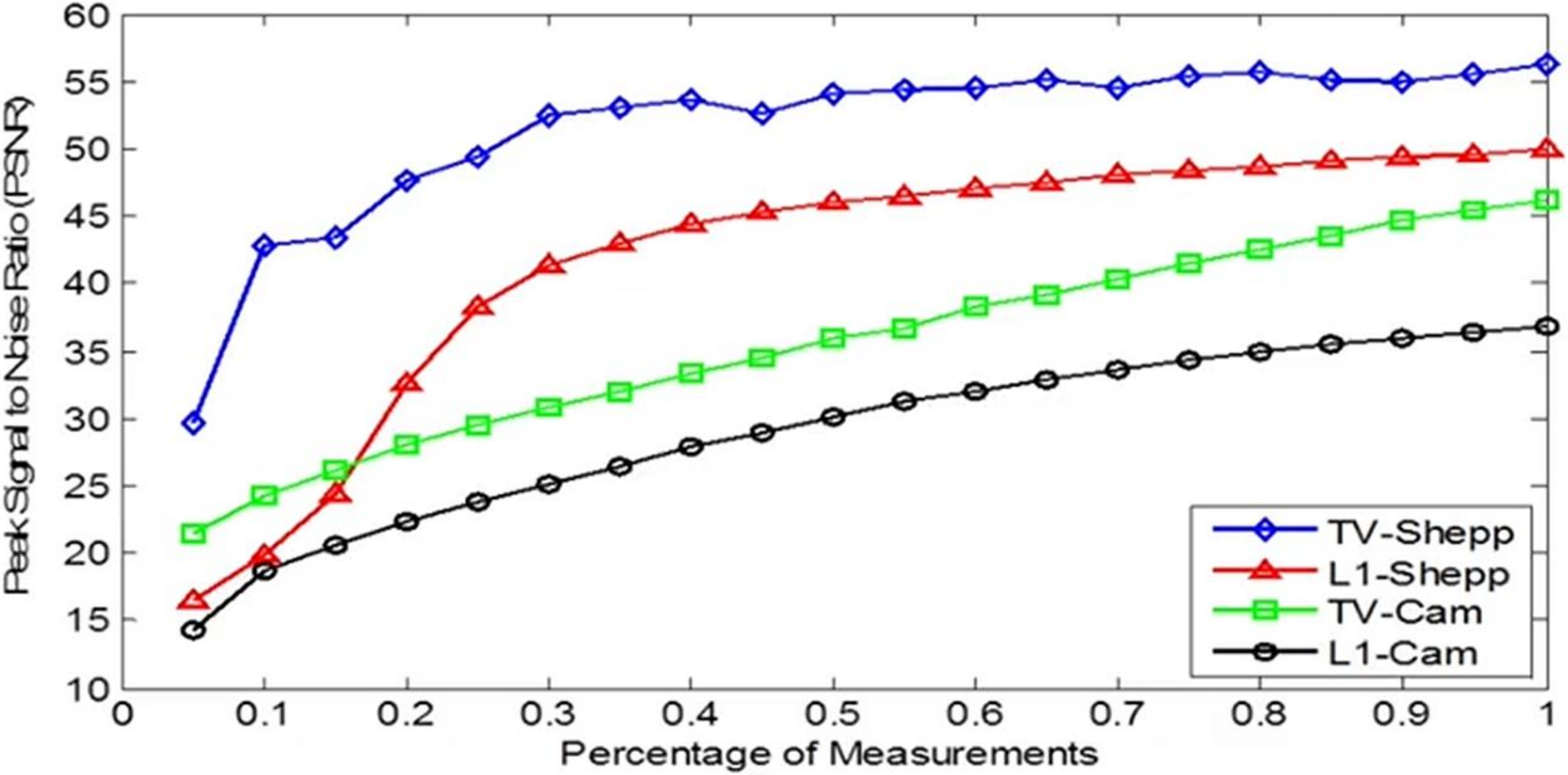
- Random measurements can be used for signals sparse in any basis



Results Compressive Sensing



Results



Peak Signal to Noise Ratio (PSNR)
Percentage of Measurements
TV-Shepp
L1-Shepp
TV-Cam
L1-Cam

Outline

- Examples of Applications of Sparsity
- L_2, L_1, and L_0 Norms
  - Linear Inverse Problems
  - Minimum L_2, L_1, and L_0 Norm Solution
- Solution Approaches
  - Matching Pursuit
  - Smooth Reformulations
  - Dictionary Learning
- Sparse Solutions to Some Applications
  - Image Denoising, Image Inpainting, Image Super-Resolution, Robust Face Recognition, Video Surveillance, Compressive Sensing

Sparsity, Lasso: http://www.stat.cmu.edu/~ryantibs/statml/lectures/sparsity.pdf