Cody Martin

Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- **Linear Regression**
  - Over-fitting
  - Regularization
  - L_1-norm regularization

Least-Squares Regression

- **Linear regression minimizes mean squared error for a set of sampling points**

$$\text{M samples}\left\{\underbrace{\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}}_{\text{N coefficients}}\right. \cdot \alpha = \begin{bmatrix} \\ B \\ \\ \end{bmatrix}$$

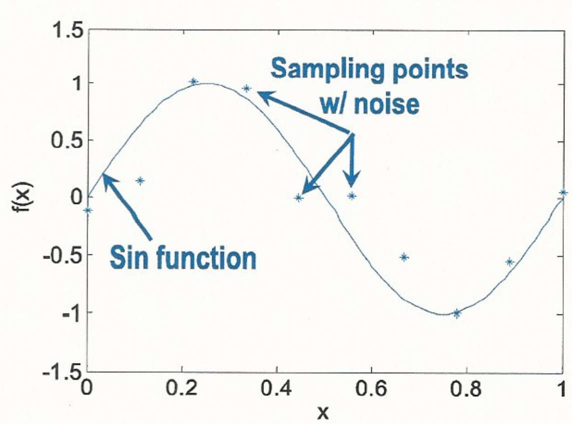
$\min_{\alpha} \|A\alpha - B\|_2^2$

- **In practice, M must be substantially larger than N (i.e., M >> N) to avoid over-fitting**

A Simple Over-Fitting Example

- **Approximate sinusoidal function by polynomial model**

$$f(x) = \sin(2\pi \cdot x) \approx \sum_{k=0}^{K} \alpha_k x^k \quad 0 \le x \le 1$$

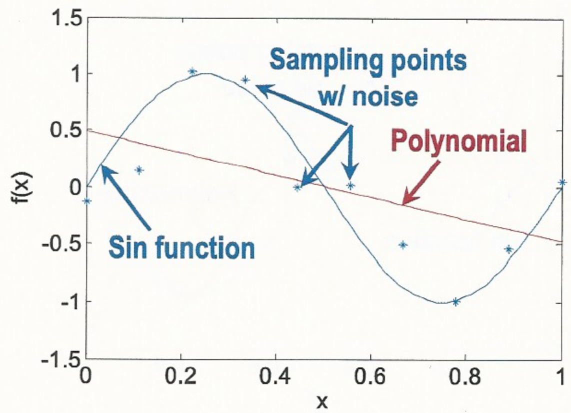


A Simple Over-Fitting Example

- **First-order polynomial model (K = 1)**

$$f(x) = \sin(2\pi \cdot x) \approx \sum_{k=0}^{K=1} \alpha_k x^k \quad 0 \le x \le 1$$

$\alpha_0 + \alpha_1 x$

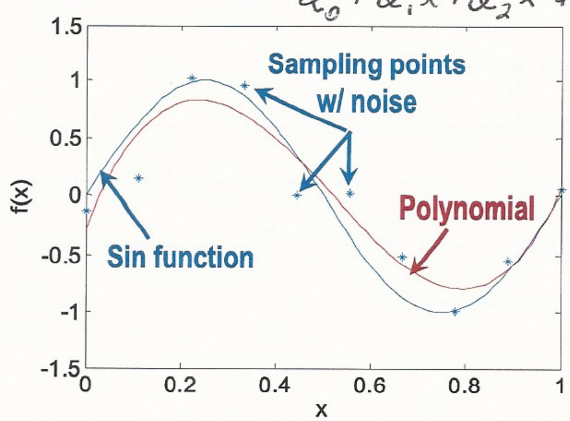


A Simple Over-Fitting Example

- **3rd-order polynomial model (K = 3)**

$$f(x) = \sin(2\pi \cdot x) \approx \sum_{k=0}^{K=3} \alpha_k x^k \quad 0 \le x \le 1$$

$\alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3$

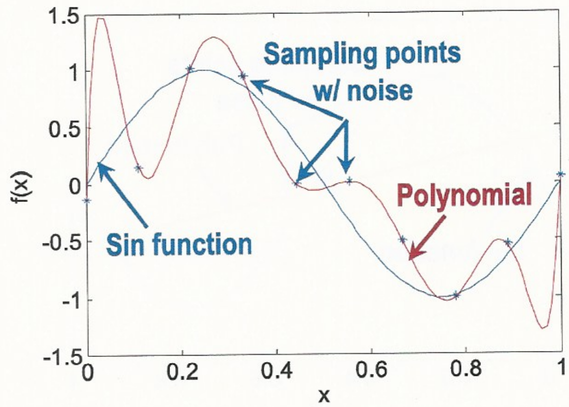


A Simple Over-Fitting Example

- **9rd-order polynomial model (K = 9)**

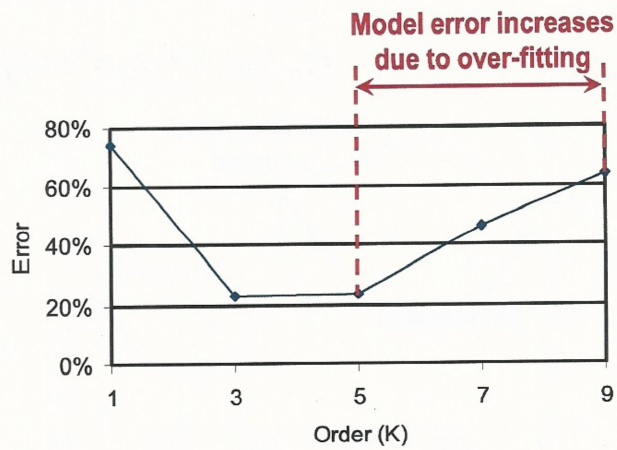
$$f(x) = \sin(2\pi \cdot x) \approx \sum_{k=0}^{K=9} \alpha_k x^k \quad 0 \le x \le 1$$

$\alpha_0 + \alpha_1 x + \alpha_2 x^2 + \ldots + \alpha_9 x^9$



A Simple Over-Fitting Example

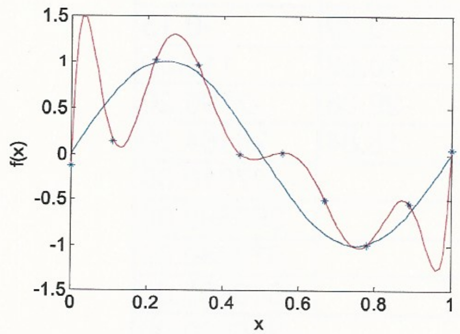
- **Model order vs. model error**
  - Increasing model complexity does not necessarily increase model accuracy



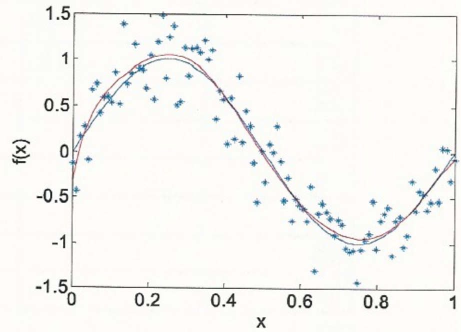
Over-Fitting

- **Increasing the number of samples helps to reduce over-fitting**

$$f(x)=\sin(2\pi\cdot x)\approx\sum_{k=0}^{K=9}\alpha_k x^k \quad 0\le x\le 1$$



9-th order model fitted from 10 sampling points



9-th order model fitted from 100 sampling points

Regularization

- **In practice, additional sampling points may be difficult and/or expensive to collect**
  - E.g., a single sampling point may be collected by a physical experiment that takes several months to finish

- **Regularization is a useful technique to minimize over-fitting**
  - What is regularization?
  - How does it work?

Regularization

- **Our previous example:**

$$f(x)=\sin(2\pi\cdot x)\approx\sum_{k=0}^{K}\alpha_k x^k \qquad 0\le x\le 1$$

	K = 1	K = 3	K = 9
α0	0.48	-0.29	-0.13
α1	-0.96	10.58	116.89
α2		-29.08	-2796.20
α3		18.84	26454.00
α4			-127610.00
α5			350590.00
α6			-572280.00
α7			549450.00
α8			-286530.00
α9			62610.00

Coefficients become extremely large due to over-fitting

Regularization

- **Key idea: large coefficient value should be penalized**

$$\text{M samples}\left\{\underbrace{\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}}_{\text{N coefficients}}\cdot\alpha=\begin{bmatrix} \\ B \\ \\ \end{bmatrix}\right.$$

$$\min_{\alpha}\ \|A\cdot\alpha-B\|_2^2$$

Ordinary least-squares

$$\min_{\alpha}\ \|A\alpha-B\|_2^2+\lambda\|\alpha\|_2^2$$

- λ determines how much $\|\alpha\|_2$ should be penalized

Regularization

- **Regularization can be re-written as a least-squares problem**

$$\|A\alpha - B\|_2^2 + \lambda \cdot \|\alpha\|_2^2 = \|A\alpha - B\|_2^2 + \lambda \cdot \alpha_1^2 + \lambda \cdot \alpha_2^2 + \cdots$$

$$\begin{cases} A\alpha = B \\ \sqrt{\lambda}\alpha_1 = 0 \\ \sqrt{\lambda}\alpha_2 = 0 \\ \vdots \\ \sqrt{\lambda}\alpha_N = 0 \end{cases}$$

$$\left\| \begin{bmatrix} A \\ \hline \sqrt{\lambda} & & \\ & \ddots & \\ & & \sqrt{\lambda} \end{bmatrix} [\alpha] - \begin{bmatrix} B \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix} \right\|_2^2 = \|A\alpha - B\|_2^2 + (\sqrt{\lambda}\alpha_1 - 0)^2 + (\sqrt{\lambda_N}\alpha_N - 0)^2$$

$$= \|A\alpha - B\|_2^2 + \lambda\alpha_1^2 + \lambda\alpha_2^2 + \ldots + \lambda\alpha_N^2$$

Regularization

$$\min_{\alpha} \quad \|A \cdot \alpha - B\|_2^2 + \lambda \cdot \|\alpha\|_2^2$$

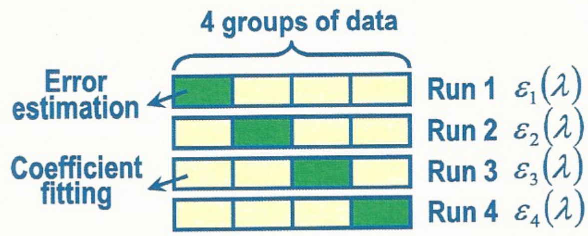
- **Regularization result depends on λ**
  - λ = 0 → ordinary least-squares regression (over-fitting)
  - λ = inf → α = 0 (over-penalized)

- **Optimal λ value is case-dependent**
  - Require a smart algorithm to automatically determine λ to achieve minimal modeling error
  - Question: how to estimate modeling error by using a set of training samples?

Cross Validation

- **λ is often determined by cross validation**
  - Calculate coefficients from training set
  - Estimate error from testing set

- **Example: 4-fold cross validation**



$$Error(\lambda) = [\varepsilon_1(\lambda) + \varepsilon_2(\lambda) + \varepsilon_3(\lambda) + \varepsilon_4(\lambda)]/4$$

Cross Validation

- **Example: 4-fold cross validation**
  - Solve the regularization problem with different λ values
  - Estimate error Error(λ) for each λ
  - Find the optimal λ to minimize Error(λ)

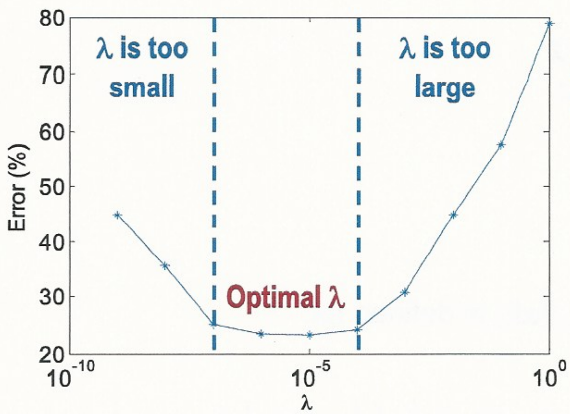
$$\min_{\alpha} \quad \|A \cdot \alpha - B\|_2^2 + \lambda \cdot \|\alpha\|_2^2$$

$$\min_{\lambda} \ Error(\lambda) = [\varepsilon_1(\lambda) + \varepsilon_2(\lambda) + \varepsilon_3(\lambda) + \varepsilon_4(\lambda)]/4$$

A Simple Regularization Example

- 9-th order polynomial model fitted from 10 sampling points

$$f(x) = \sin(2\pi \cdot x) \approx \sum_{k=0}^{K=9} \alpha_k x^k \quad 0 \le x \le 1 \qquad \min_{\alpha} \quad \|A \cdot \alpha - B\|_2^2 + \lambda \cdot \|\alpha\|_2^2$$



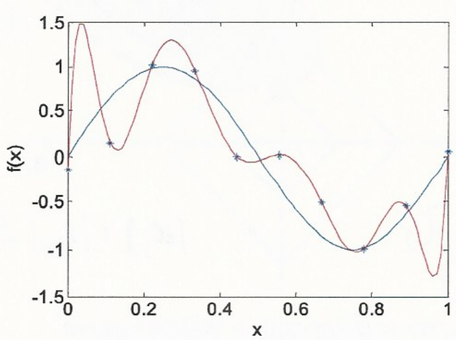
$\lambda = 0 \rightarrow$ ordinary least-squares

$\lambda = \text{inf} \rightarrow \alpha = 0$

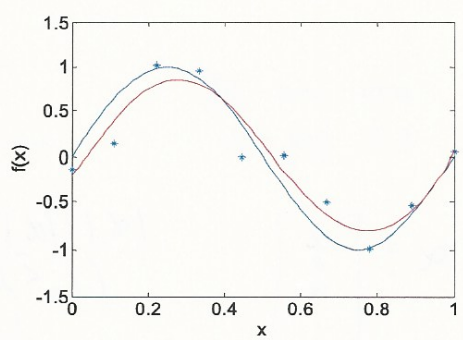
A Simple Regularization Example

- 9-th order polynomial model fitted from 10 sampling points

$$f(x) = \sin(2\pi \cdot x) \approx \sum_{k=0}^{K=9} \alpha_k x^k \quad 0 \le x \le 1 \qquad \min_{\alpha} \quad \|A \cdot \alpha - B\|_2^2 + \lambda \cdot \|\alpha\|_2^2$$



Without regularization



With regularization ($\lambda = 10^{-5}$)

Regularization

- **Several other possible forms of regularization**

$$\min_{\alpha} \quad \|A\cdot\alpha - B\|_2^2 + \lambda\cdot\|\alpha\|_2^2$$

$$\min_{\alpha} \|A\alpha - B\|_2^2 + \lambda\|\alpha\|_2^2$$

$$\min \|A\alpha - B\|_2^2$$
$$\text{S.T. } \|\alpha\| \le \lambda$$

- For a vector $\alpha \in R^N$, $\|\alpha\|_1$ is defined as:

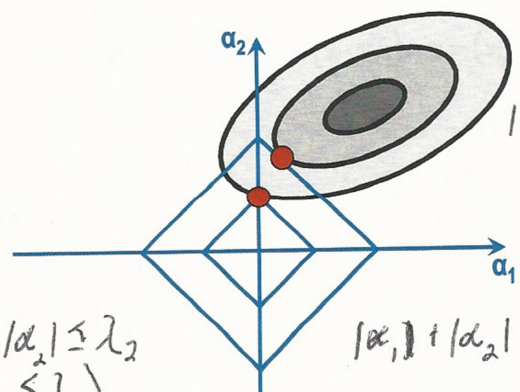
$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \end{bmatrix} \qquad \|\alpha\|_1 = |\alpha_1| + |\alpha_2| + \ldots$$

L_1-Norm Regularization

- **L_1-norm regularization is often used to find sparse solution of a linear equation**

$$\min_{\alpha} \quad \|A\alpha - B\|_2^2$$
$$\text{S.T.} \quad \|\alpha\|_1 \le \lambda$$

$$\alpha = \begin{bmatrix} \alpha \\ 0 \\ \alpha \\ \vdots \end{bmatrix}$$



L_1-norm regularization yields sparse solution if λ is sufficiently small

L_1-Norm Regularization

- L_1-norm regularization can be solved by convex programming

$$\min_{\alpha} \ \|A\alpha - B\|_2^2$$
$$\text{S.T.} \ \ \|\alpha\|_1 \le \lambda$$

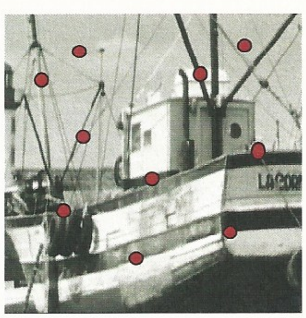
$$\underset{\alpha}{\text{Min}} \ \|A\alpha - B\|_2^2$$
$$\text{S.T.} \ \ t_1 + t_2 + \dots t_n \le \lambda$$
$$|\alpha_n| \le t_n \ \leftarrow \ -t_n \le \alpha_n \le t_n$$
$$(n = 1, \dots, N)$$

  - Convex quadratic cost function
  - Linear constraints

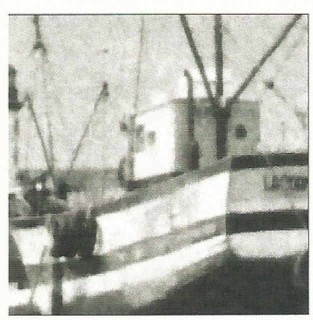
L_1-norm regularization has been applied to a large number of practical problems, e.g., image processing

Image Processing Example

- Image re-construction (compressed sensing)
  - Sample an image at a small number of locations
  - Recover full image by a numerical algorithm
  - More details in next lecture…



Original image



Recovered image

Summary

- **Linear regression**
  - Over-fitting
  - Regularization
  - L_1-norm regularization

Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- Compressed Sensing
  - Orthogonal matching pursuit (OMP)

L_1-Norm Regularization

- L_1-norm regularization is often used to find sparse solution of a linear equation

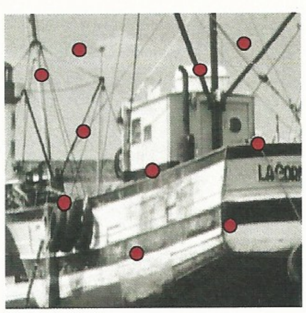
$$\min_{\alpha} \quad \|A\alpha - B\|_2^2$$
$$\text{S.T.} \quad \|\alpha\|_1 \le \lambda$$

(Convex optimization)

- L_1-norm regularization has been applied to a large number of practical problems
  - In this lecture, we will focus on the compressed sensing problem for image processing

Compressed Sensing

- Our focus: image re-construction
  - Sample an image at a small number of spatial locations
  - Recover full image by a numerical algorithm



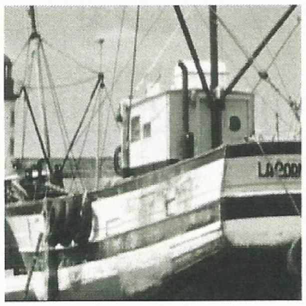
Original image

Recovered image

Compressed Sensing

- A 2-D image can be mapped to frequency domain by discrete cosine transform (DCT)

$$\underbrace{G(u,v)}_{\text{DCT coefficient}}=\sum_{x=1}^{P}\sum_{y=1}^{Q}a_u\cdot b_v\cdot \underbrace{g(x,y)}_{\text{Image pixel}}\cdot\cos\frac{\pi(2x-1)(u-1)}{2\cdot P}\cdot\cos\frac{\pi(2y-1)(v-1)}{2\cdot Q}$$



Original image

$$x,u\in\{1,2,\cdots,P\}$$

$$y,v\in\{1,2,\cdots,Q\}$$

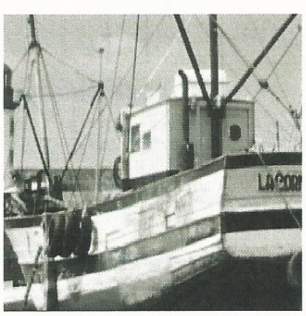
$$a_u=\begin{cases}\sqrt{1/P} & (u=1)\\ \sqrt{2/P} & (2\le u\le P)\end{cases}$$

$$b_v=\begin{cases}\sqrt{1/Q} & (v=1)\\ \sqrt{2/Q} & (2\le v\le Q)\end{cases}$$

Compressed Sensing

- A 2-D image can be uniquely determined by inverse discrete cosine transform (IDCT), if all DCT coefficients are known

$$\underbrace{g(x,y)}_{\text{Image pixel}}=\sum_{u=1}^{P}\sum_{v=1}^{Q}a_u\cdot b_v\cdot \underbrace{G(u,v)}_{\text{DCT coefficient}}\cdot\cos\frac{\pi(2x-1)(u-1)}{2\cdot P}\cdot\cos\frac{\pi(2y-1)(v-1)}{2\cdot Q}$$



Original image

$$x,u\in\{1,2,\cdots,P\}$$

$$y,v\in\{1,2,\cdots,Q\}$$

$$a_u=\begin{cases}\sqrt{1/P} & (u=1)\\ \sqrt{2/P} & (2\le u\le P)\end{cases}$$

$$b_v=\begin{cases}\sqrt{1/Q} & (v=1)\\ \sqrt{2/Q} & (2\le v\le Q)\end{cases}$$

Compressed Sensing

- **Sample a 2-D image at a number of (say, M) spatial locations**
  - Result in a set of (i.e., M) linear equations

$$g(x_1,y_1)=\sum_{u=1}^{P}\sum_{v=1}^{Q} a_u \cdot b_v \cdot G(u,v)\cdot \cos\frac{\pi(2x_1-1)(u-1)}{2\cdot P}\cdot \cos\frac{\pi(2y_1-1)(v-1)}{2\cdot Q}$$

$$g(x_2,y_2)=\sum_{u=1}^{P}\sum_{v=1}^{Q} a_u \cdot b_v \cdot G(u,v)\cdot \cos\frac{\pi(2x_2-1)(u-1)}{2\cdot P}\cdot \cos\frac{\pi(2y_2-1)(v-1)}{2\cdot Q}$$

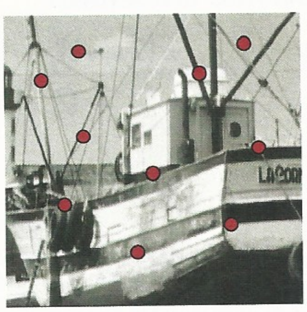
$$\vdots$$

$$\underbrace{g(x_M,y_M)}_{\text{Image pixel}}=\sum_{u=1}^{P}\sum_{v=1}^{Q} a_u \cdot b_v \cdot \underbrace{G(u,v)}_{\text{DCT coefficient}}\cdot \cos\frac{\pi(2x_M-1)(u-1)}{2\cdot P}\cdot \cos\frac{\pi(2y_M-1)(v-1)}{2\cdot Q}$$

Our goal is to solve all DCT coefficients from these linear equations

Compressed Sensing

- **Re-write the linear equation in matrix form**



Original image

DCT transform

$$\begin{bmatrix}\times\\ \times\\ \times\end{bmatrix}=\begin{bmatrix}\times&\times&\times&\times&\times\\ \times&\times&\times&\times&\times\\ \times&\times&\times&\times&\times\end{bmatrix}\cdot\begin{bmatrix}\times\\ \times\\ \times\\ \times\\ \times\end{bmatrix}$$

Sampling data

DCT coefficients

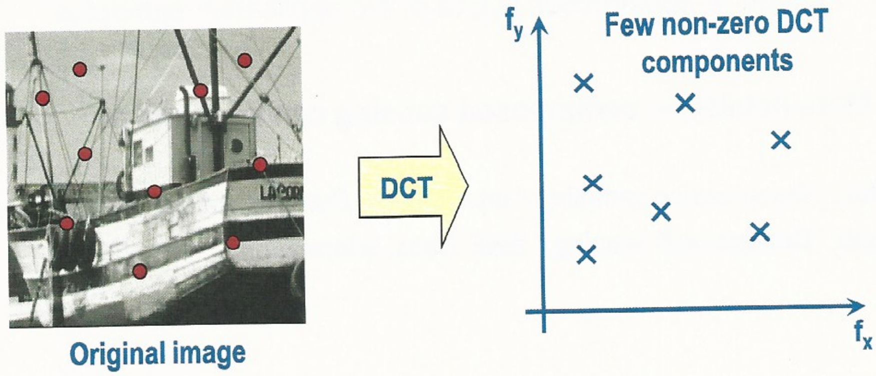
16×16 256 coefficients

M=100 equations

Result in an under-determined linear equation, since we have less sampling locations than unknown DCT coefficients

Compressed Sensing

- Additional information is required to uniquely solve under-determined linear equation



Explore sparsity to find unique, deterministic solution from under-determined equation

Compressed Sensing

- We know that α is sparse – but we do *not* know the exact location of zeros
  - Find the sparse solution α (i.e., DCT coefficients) for $A\alpha = B$
  - Apply inverse DCT transform to recover full image

$$\underbrace{\begin{bmatrix} \times \\ \times \\ \times \end{bmatrix}}_{B} = \overbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}^{A} \cdot \underbrace{\begin{bmatrix} 0 \\ \times \\ \times \\ 0 \\ 0 \end{bmatrix}}_{\alpha \text{ (Sparse)}}$$

$$\text{Min } \|A\alpha - B\|_2^2$$
$$\text{S.T. } \|\alpha\|_1 \le \lambda$$

(convex optimization!)

Compressed Sensing

- Compressed sensing is a general technique that is applicable to many other practical problems
  - Image re-construction is one of the application examples

- More details on compressed sensing can be found at

Candes, "Compressive sampling," International Congress of Mathematicians, 2006

Donoho, "Compressed sensing," IEEE Trans. Information Theory, 2006.

Orthogonal Matching Pursuit (OMP)

- Another way to find sparse solution is L_0-norm regularization

$$\begin{aligned} \min_{\alpha} \quad & \|A\alpha - B\|_2^2 \\ \text{S.T.} \quad & \underline{\|\alpha\|_0} \le \lambda \end{aligned}$$

(VERY difficult to solve)

Number of non-zeros in α

- Efficient numerical algorithm exists to find an approximate (i.e., sub-optimal) solution
  - E.g., orthogonal matching pursuit

Orthogonal Matching Pursuit (OMP)

- **Goal:**
  - Identify a subset of DCT coefficients that are non-zero
- **Approach:**
  - Find important DCT coefficients by checking the inner product between A_i and B
  - Assume that each A_i is normalized (i.e., has unit length)

$A \cdot \alpha = B$

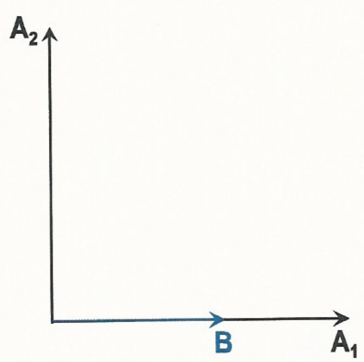
$\langle A_i, B \rangle = A_i^T \cdot B$

$A = [A_1 | A_2 | A_3 | \cdots] \cdot B = [\ ]$

$\|A_i\|_2 = 1$

Orthogonal Matching Pursuit (OMP)

- **Inner product <A_i, B> implies the importance of A_i when approximating B**
- **2-D example**



$\alpha_1 \cdot A_1 + \alpha_2 \cdot A_2 \approx B$

$\langle A_1, B \rangle \neq 0$

$\langle A_2, B \rangle = 0$

$\Rightarrow |\langle A_1, B \rangle| > |\langle A_2, B \rangle|$

A_1 is important

Orthogonal Matching Pursuit (OMP)

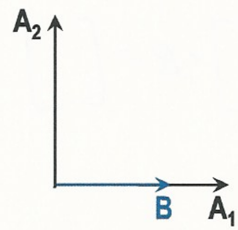
- **2-D example**

$$\alpha_1 \cdot A_1 + \alpha_2 \cdot A_2 \approx B$$

  - Step 1: Calculate $<A_1, B>$ and $<A_2, B>$
  - Step 2: Select A_i that corresponds to the largest inner product magnitude (i.e., A_1 in this example)
  - Step 3: Solve the coefficient α_1 by least-squares fitting

$$\min_{\alpha_1} \quad \left\| \alpha_1 \cdot A_1 - B \right\|_2^2$$

  - Step 4: Set $\alpha_2 = 0$ (B is independent of A_2 in this example)

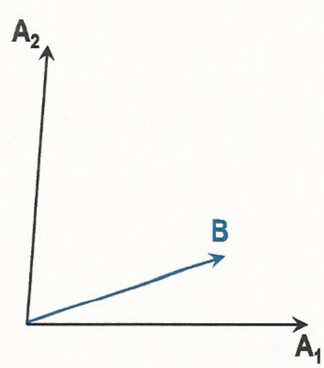


Orthogonal Matching Pursuit (OMP)

- **2-D example**

$$\alpha_1 \cdot A_1 + \alpha_2 \cdot A_2 \approx B$$

  - A_1 and A_2 are not orthogonal
  - B is not orthogonal to A_1 or A_2



Orthogonal Matching Pursuit (OMP)

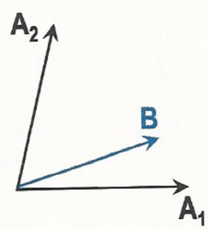
- **2-D example**

$$\alpha_1 \cdot A_1 + \alpha_2 \cdot A_2 \approx B$$

  - Step 1: Calculate $<A_1, B>$ and $<A_2, B>$
  - Step 2: Select A_i that corresponds to the largest inner product magnitude (i.e., A_1 in this example)
  - Step 3: Solve the coefficient α_1 by least-squares fitting

$$\min_{\alpha_1} \quad \left\|\alpha_1 \cdot A_1 - B\right\|_2^2$$

$$|<A_1, B>| > |<A_2, B>|$$



Orthogonal Matching Pursuit (OMP)

- **2-D example**

$$\alpha_1 \cdot A_1 + \alpha_2 \cdot A_2 \approx B$$

  - Step 4: Calculate the residue

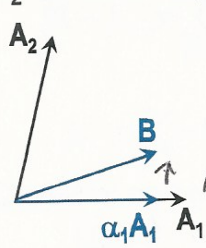
$$F = B - \alpha_1 \cdot A_1$$ **(F is orthogonal to A_1)**

  - Step 5: Calculate $<A_1, F>$ and $<A_2, F>$
  - Step 6: Select A_i that corresponds to the largest inner product magnitude (i.e., A_2 in this example)

$<A_1, F> = 0$

$<A_2, F> \neq 0$

$|<A_1, F>| < |<A_2, F>|$



$F = B - \alpha_1 A_1$

Orthogonal Matching Pursuit (OMP)

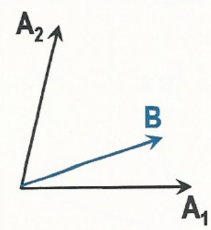
- **2-D example**

$$\alpha_1 \cdot A_1 + \alpha_2 \cdot A_2 \approx B$$

- Step 7: Solve the coefficients α_1 and α_2 by least-squares fitting

$$\min_{\alpha_1, \alpha_2} \quad \left\| \alpha_1 \cdot A_1 + \alpha_2 \cdot A_2 - B \right\|_2^2$$

(α_1 is re-calculated)



Orthogonal Matching Pursuit (OMP)

- **General OMP algorithm to solve underdetermined linear equation**

$$\min_{\alpha} \quad \left\| A\alpha - B \right\|_2^2$$
$$\text{S.T.} \quad \left\| \alpha \right\|_0 \le \lambda$$

- Step 1: Set F = B, $\Omega = \{\}$ and p = 1
- Step 2: Calculate the inner product values $\theta_i = <A_i, F>$
- Step 3: Identify the index s for which $|\theta_s|$ takes the largest value
- Step 4: Update Ω by $\Omega = \Omega \cup \{s\}$
- Step 5: Approximate B by the linear combination of $\{A_i; i \in \Omega\}$

$$\min_{\alpha_i, i \in \Omega} \quad \left\| \sum_{i \in \Omega} \alpha_i \cdot A_i - B \right\|_2^2$$

- Step 6: Update F

$$F = B - \sum_{i \in \Omega} \alpha_i \cdot A_i$$

- Step 7: If $p < \lambda$, p = p+1 and go to Step 2. Otherwise, stop.

$$\alpha_i = 0 \quad (i \notin \Omega)$$

Orthogonal Matching Pursuit (OMP)

$$\begin{array}{ll} \min\limits_{\alpha} & \|A\alpha - B\|_2^2 \\ \text{S.T.} & \|\alpha\|_0 \le \lambda \end{array}$$

- **Orthogonal matching pursuit is a heuristic algorithm to solve the L_0-norm regularization problem**
- **A number of other heuristic algorithms exist to solve under-determined linear equation**
  - More details can be found in compressed sensing papers

Summary

- **Compressed sensing**
  - Orthogonal matching pursuit (OMP)