Today:  
- LS review & Examples
- Intro to Matching Pursuit.

**Summary of LS:**

The set of solutions to the LS problem:

\[
\min_{\hat{x}} \| A \hat{x} - \hat{b} \|^2
\]

is given by the set of solutions to:

\[
(A^T A) \hat{x} = A^T \hat{b} \quad \text{(normal Eqs)}
\]

In particular, if \( N(A) = \emptyset \), then

\[
\hat{x}^* = (A^T A)^{-1} A^T \hat{b}.
\]

Ex: \( A = \hat{a} \), \( \hat{x}^* = \frac{\langle \hat{a}, \hat{b} \rangle}{\langle \hat{a}, \hat{a} \rangle} \leftarrow A^T \hat{b} \)
\[ \|A\hat{x} - b\|_2^2 = \|A\hat{x}^* - b\|_2^2 + \|A\hat{x} - A\hat{x}^*\|_2^2 \geq \|A\hat{x}^* - b\|_2^2 \]

To verify desired orthogonality, just use normal equations:

\[ \langle b - A\hat{x}^*, A\hat{x} \rangle = \hat{x}^T A^T b - \hat{x}^T A^T A\hat{x}^* = \hat{x}^T A^T b - \hat{x}^T A^T b = 0. \]

**Ex 2:** \( y = ax + b \)
\[ \text{LS problem:} \]
\[
\min_{a,b} \left\| \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \right\|^2
\]
\[
= \sum_{i=1}^{m} (ax_i + b - y_i)^2
\]
vertical distances between data points and line \( ax + b \).

**Remark:** To get some intuition about what LS problems look like when \( \mathbb{N}(A) \neq 0 \), consider Linear Regression example. In this case, all \( x_i \)'s are equal, 
\[
\begin{array}{c}
\text{y}_i \\
\text{x}_i \\
\end{array}
\]
\[
\begin{array}{c}
\text{y}_i \\
\text{x}_i \\
\end{array}
\]
Ex: Hubble Space Telescope

New Examples:

Common Misconception: LS problems look "linear", i.e., like linear regression.
Ex: polynomial regression.
\[ p(x) = p_0 + p_1 x + p_2 x^2 + p_3 x^3 \]

Want:
\[ p(x) = p_0 + p_1 x_i + p_2 x_i^2 + p_3 x_i^3 \approx y_i \quad \forall i. \]

Variables

System of LEQs in variables \( p_0, \ldots, p_3 \).

LS problem:
\[ \min_{p_0, p_1, p_2, p_3} \sum_{i=1}^{m} (p(x_i) - y_i)^2 \]

To solve for \( p_i \)'s, just find solutions to:
\[ A^T A \hat{p} = A^T \hat{y} \]
Ex: Piazzi discovery of Ceres (dwarf planet)

Kepler's law of planetary motion: planets have elliptical orbits.

So, in (x, y) - plane

\[ \alpha x^2 + \beta y^2 + \gamma xy + dx + ey = 1 \]

Eqn for ellipse - coefficients (unknowns) determine shape of ellipse.

? (? , y)

(\(x_i, y_i\))

?
LS problem:

\[ \text{min} \quad \| \begin{bmatrix} x_i^2 & y_i^2 & x_i y_i & x_i & y_i \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} - \frac{1}{n} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \|_2^2 \]

\[ \sum_{i=1}^{n} (a x_i^2 + b y_i^2 + c x_i y_i + d x_i + e y_i - 1)^2 \]

\underline{Ex: Prediction of stock prices.}

Let \( x_i \), \( i = 1 \ldots T \) be stock price on day \( i \).

From these, we want to predict \( x_{T+1} \).
Propose model:

\( a_1 x_{i+1} + a_2 x_{i+2} + \ldots + a_k x_{i+k} \approx x_{i+k+1} \)

i.e. stocks can be predicted by looking at prev k days (but there is noise!)

Goal: learn model i.e., \( a_i's \) from past data.

Set up LS problem:

\[
\min_{a_1 \ldots a_k} \left\| \begin{bmatrix} x_1 & \ldots & x_k \\ x_2 & \ldots & x_{k+1} \\ \vdots & & \vdots \\ x_{T-k} & \ldots & x_T \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix} - \begin{bmatrix} x_{k+1} \\ \vdots \\ x_T \end{bmatrix} \right\|^2
\]
\[
\sum_{i=1}^{T-k} \left( a_i x_{i+\cdots} a_k x_{i+k-1} - x_{i+k} \right)^2
\]

Bottom line: Trick is to recognize your problem can be cast as a LS problem. Then just use generic LS solution to solve.

Matching Pursuit:

Often times we want to solve a LS problem \( A \hat{x} = \hat{b} \) in a way that gives a "sparse" solution.

(\text{sparse} = \text{mostly zero entries} ).

Ex:
\[
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
\ddot{x}_1 \\
\ddot{x}_2 \\
\ddot{x}_3 \\
\ddot{x}_4 \\
\ddot{x}_5
\end{pmatrix} = \begin{pmatrix}
1 \\
1 \\
1 \\
1 \\
1
\end{pmatrix}
\]

Many solutions: 

Examples of solutions:

\[
\begin{pmatrix}
\ddot{x}_1 \\
\ddot{x}_2 \\
\ddot{x}_3 \\
\ddot{x}_4 \\
\ddot{x}_5
\end{pmatrix} = \begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

Sparse solution

More formally, want to solve "constrained" LS problem:

\[
\min \| A \ddot{x} - \ddot{b} \|_2 \quad \text{subject to} \quad \| \ddot{x} \|_0 \leq K.
\]

\[\| \ddot{x} \|_0 = \# \text{ of nonzero entries in } \ddot{x}.
\]

This class of problems is computationally difficult to solve optimally.
So, we resort to heuristics that give good performance in practice. Matching Pursuit (next time) is one such heuristic.

Before introducing MP, let's see why sparsity is motivated in real-life.

*Ex: Internet of Things.*

\[ \vec{r} = \sum_{i=1}^{n} x_i \vec{s}_i \]

*signature uniquely assigned to devices* all of same length
Assumption: a very small # of devices is Tx-ing at any given time.

Hence, to determine $\hat{x}$'s, can solve

$$\min_{\hat{x} \in \mathbb{R}^n} \| A \hat{x} - \hat{r} \|_2^2 \quad \text{s.t.} \quad \| \hat{x} \|_0 \leq k.$$ 

The above matrix

$$\begin{bmatrix}
\frac{1}{s_1} & \cdots & \frac{1}{s_n} \\
1 & \cdots & 1
\end{bmatrix}$$

in general signatures will be linearly dependent.

Ex: Classification Problem.

A proprietary pharmaceutical has chemical composition

$$\begin{bmatrix}
\frac{b_1}{b_2} & \cdots & \frac{b_1}{b_n}
\end{bmatrix}, \quad b_i = \text{concentration of chem i.}$$
There are a variety of generic compounds we can purchase, with compositions

\[ \vec{a}_k = \begin{bmatrix} a_{1k} \\ \vdots \\ a_{nk} \end{bmatrix} \]

then composition of compound \( k \).

Goal: solve:

\[
\min \| \begin{bmatrix} \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \vec{b} \|_2 \\
\text{subject to } \begin{bmatrix} \vec{1} \\ \vec{a}_1 \\ \vdots \\ \vec{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{bmatrix}
\]

to find a few ingredients we can combine to make \( \vec{b} \).