# Machine Learning

## Linear Models

Fabio Vandin

October $29^{th}$, 2020

# Linear Predictors and Affine Functions

Consider $\mathcal{X} = \mathbb{R}^d$

**"Linear" (affine) functions**:

$$L_d = \{h_{\mathbf{w},b} : \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\}$$

where

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i x_i \right) + b$$

**Note**:

- each member of $L_d$ is a function $\mathbf{x} \rightarrow \langle \mathbf{w}, \mathbf{x} \rangle + b$
- $b$: *bias*

# Linear Models

Hypothesis class $\mathcal{H}$: $\phi \circ L_d$, where $\phi : \mathbb{R} \to \mathcal{Y}$

- $h \in \mathcal{H}$ is $h : \mathbb{R}^d \to \mathcal{Y}$

$\phi$ depends on the learning problem

**Example**

- binary classification, $\mathcal{Y} = \{-1, 1\} \Rightarrow \phi(z) = \text{sign}(z)$
- regression, $\mathcal{Y} = \mathbb{R} \Rightarrow \phi(z) = z$

# Equivalent Notation

Given $\mathbf{x} \in \mathcal{X}$, $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, define:

- $\mathbf{w}' = (b, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$
- $\mathbf{x}' = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$

Then:

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle \tag{1}$$

$\Rightarrow$ we will consider bias term as part of $\mathbf{w}$ and assume $\mathbf{x} = (1, x_1, x_2, \ldots, x_d)$ *when needed*, with $h_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$

# Linear Regression

$\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$

Hypothesis class:

$$\mathcal{H}_{reg} = L_d = \{\mathbf{x} \to \langle \mathbf{w}, \mathbf{x} \rangle + b : \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\}$$

**Note:** $h \in H_{reg} : \mathbb{R}^d \to \mathbb{R}$

Commonly used loss function: *squared-loss*

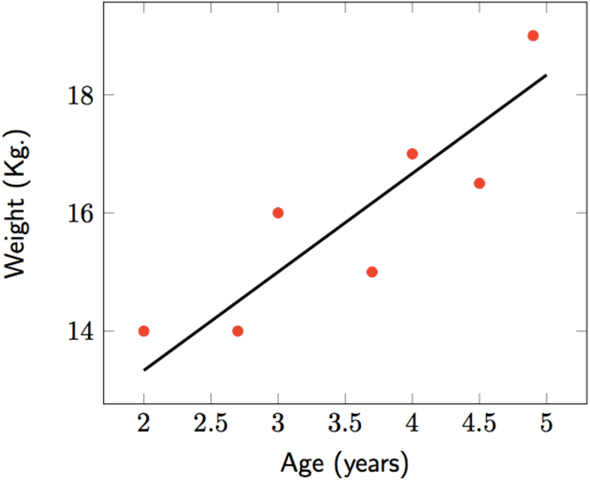
$$\ell(h, (\mathbf{x}, y)) \overset{def}{=} (h(\mathbf{x}) - y)^2$$

$\Rightarrow$ empirical risk function (training error): *Mean Squared Error*

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} (h(\mathbf{x}_i) - y_i)^2$$

# Linear Regression - Example

$d = 1$



# Least Squares

How to find a ERM hypothesis? *Least Squares* algorithm

Best hypothesis:

$$\arg\min_{\mathbf{w}} L_S(h_{\mathbf{w}}) = \arg\min_{\mathbf{w}} \frac{1}{m} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2$$

Equivalent formulation: $\mathbf{w}$ minimizing *Residual Sum of Squares* (RSS), i.e.

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2$$

# RSS: Matrix Form

Let

$$\mathbf{X} = \begin{bmatrix} \cdots & \mathbf{x}_1 & \cdots \\ \cdots & \mathbf{x}_2 & \cdots \\ \cdots & \vdots & \cdots \\ \cdots & \mathbf{x}_m & \cdots \end{bmatrix}$$

$\mathbf{X}$: *design matrix*

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$\Rightarrow$ we have that RSS is

$$\sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

Want to find $\mathbf{w}$ that minimizes RSS (=*objective function*):

$$\arg\min_{\mathrm{w}} RSS(\mathbf{w}) = \arg\min_{\mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

How?

Compute gradient $\frac{\partial RSS(\mathbf{w})}{\partial \mathbf{w}}$ of objective function w.r.t $\mathbf{w}$ and compare it to 0.

$$\frac{\partial RSS(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})$$

Then we need to find $\mathbf{w}$ such that

$$-2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

$$-2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

is equivalent to

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

If $\mathbf{X}^T\mathbf{X}$ is invertible $\Rightarrow$ solution to ERM problem is:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

# Complexity Considerations

We need to compute

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Algorithm:

1. compute $\mathbf{X}^T\mathbf{X}$: product of $(d+1) \times m$ matrix and $m \times (d+1)$ matrix
2. compute $(\mathbf{X}^T\mathbf{X})^{-1}$ inversion of $(d+1) \times (d+1)$ matrix
3. compute $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$: product of $(d+1) \times (d+1)$ matrix and $(d+1) \times m$ matrix
4. compute $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$: product of $(d+1) \times m$ matrix and $m \times 1$ matrix

Most expensive operation? Inversion!

$\Rightarrow$ done for $(d+1) \times (d+1)$ matrix

# $\mathbf{X}^T\mathbf{X}$ not invertible?

How do we get $\mathbf{w}$ such that

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

if $\mathbf{X}^T\mathbf{X}$ is not invertible?
Let

$$\mathbf{A} = \mathbf{X}^T\mathbf{X}$$

Let $\mathbf{A}^+$ be the *generalized inverse* of $\mathbf{A}$, i.e.:

$$\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$$

**Proposition**

If $\mathbf{A} = \mathbf{X}^T\mathbf{X}$ is not invertible, then $\hat{w} = \mathbf{A}^+\mathbf{X}^T\mathbf{y}$ is a solution to $\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$.

# Computing the Generalized Inverse of $\mathbf{A}$

Note $\mathbf{A} = \mathbf{X}^T\mathbf{X}$ is symmetric $\Rightarrow$ eigenvalue decomposition of $\mathbf{A}$:

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T$$

with

- $\mathbf{D}$: diagonal matrix (entries = eigenvalues of $\mathbf{A}$)
- $\mathbf{V}$: orthonormal matrix ($\mathbf{V}^T\mathbf{V} = \mathbf{I}_{d \times d}$)

Define $\mathbf{D}^+$ diagonal matrix such that:

$$\mathbf{D}^+_{i,i} = \begin{cases} 0 & \text{if } \mathbf{D}_{i,i} = 0 \\ \frac{1}{\mathbf{D}_{i,i}} & \text{otherwise} \end{cases}$$

Let $\mathbf{A}^{+} = \mathbf{V}\mathbf{D}^{+}\mathbf{V}^{T}$

Then

$$\begin{aligned}
\mathbf{A}\mathbf{A}^{+}\mathbf{A} &= \mathbf{V}\mathbf{D}\mathbf{V}^{T}\mathbf{V}\mathbf{D}^{+}\mathbf{V}^{T}\mathbf{V}\mathbf{D}\mathbf{V}^{T} \\
&= \mathbf{V}\mathbf{D}\mathbf{D}^{+}\mathbf{D}\mathbf{V}^{T} \\
&= \mathbf{V}\mathbf{D}\mathbf{V}^{T} \\
&= \mathbf{A}
\end{aligned}$$

$\Rightarrow$ $\mathbf{A}^{+}$ is a generalized inverse of $\mathbf{A}$.

**In practice**: the Moore-Penrose generalized inverse $\mathbf{A}^{\dagger}$ of $\mathbf{A}$ is used, since it can be efficiently computed from the Singular Value Decomposition of $\mathbf{A}$.