# Applications of Optimization to Machine learning

Linear Regression : Dataset: $\underline{\overbrace{\begin{Bmatrix} x_1, x_2, \dots, x_n \\ y_1, y_2, \dots, y_n \end{Bmatrix}}^{\text{Training data.}}}$

$x_i \in \mathbb{R}^d \quad \forall i$

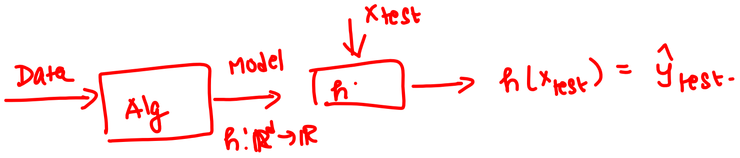
$y_i \in \mathbb{R}$

Goal: $\boxed{h : \mathbb{R}^d \to \mathbb{R}}$

$\hat{y}_{test} = h(x_{test})$ $\in \mathbb{R}^d.$

$\uparrow$ $\in \mathbb{R}.$

$\longrightarrow$ given input during test time.



$h$ is linear $\boxed{h_w(x) = \underline{w^T x}}$ for some $w \in \mathbb{R}^d$.

how to get "Best" $h$ from training data?

Performance measure : sum of squares error.

$$\min_{\underline{w} \in \mathbb{R}^d} \underbrace{\sum_{i=1}^{n} (\underbrace{w^T x_i - y_i}_{})^2}_{f(w)}$$

↳ error of $x_i$ made by $w$.

Specific Goal of linear regression

$$\boxed{\min_{w \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{n} (w^T x_i - y_i)^2}$$

$$f(w) = \underline{\sum_{i=1}^{n}} \underline{(w^T x_i - y_i)^2} \quad ?$$

$$\downarrow$$

$$\sum_{i=1}^{n} \underline{h_i(w)}$$

If $h_i(w)$ is convex for all $i$, then $f(w)$ is convex [∵ Property 1

Sums of cvx
fn is cvx]

Linear + Constant.

$$\underline{h_i(w)} = (w^T x_i - y_i)^2$$

$$= f(g(w))$$

$$g(w) = \underline{w^T x_i - \underline{y_i}} \qquad g: \mathbb{R}^d \to \mathbb{R}$$

$$f(z) = \underline{z}^2 \qquad f: \mathbb{R} \to \mathbb{R}$$

Convex.

$\Rightarrow$ Convex [Composition of
Linear + const and
in w
Convex].

**Conclusion**: $f$ is convex.

$$f(w) = \frac{1}{2} \underbrace{\sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2}$$

$$X = \underset{n \times d}{\begin{bmatrix} — x_1 — \\ — x_2 — \\ \vdots \\ — x_n — \end{bmatrix}} \qquad w = \underset{d \times 1}{\begin{bmatrix} | \\ w \\ | \end{bmatrix}} \qquad \begin{bmatrix} | \\ y \\ | \end{bmatrix} = \underset{n \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}$$

$$= \frac{1}{2} \left\| \begin{bmatrix} (w^T x_1 - y_1) \\ \vdots \\ (w^T x_n - y_n) \end{bmatrix} \right\|_2^2$$

$$z^T z = \|z\|^2 = \sum_{i=1}^{n} z_i^2$$

$$f(w) = \frac{1}{2} \| Xw - y \|_2^2$$

$$f(w) = \frac{1}{2} (Xw - y)^T (Xw - y) = \frac{1}{2} \left( \underbrace{w^T \overset{\overset{d \times d.}{\downarrow}}{X^T X} w} - 2w^T(\underline{X^T y}) + \underline{y^T y} \right)$$

$$\longrightarrow w^T A w$$

$$\underline{2Aw}$$

$$\nabla f(w) = \frac{1}{2} \left( 2 \, (X^T X) w - 2 (X^T y) \right) = (X^T X) w - (X^T y)$$

$$\boxed{\nabla f(w) = (X^TX)w - X^Ty}$$

$$(X^TX)w^* = X^Ty \quad \Rightarrow \quad \boxed{w^* = (\underline{X^TX})^{+}(X^Ty)}$$

$(\underline{X^TX})^{-1}(X^Ty)$

But $X^TX$ may not be invertible.

Advantages :- Analytical solution. $\boxed{(X^TX)^{-1}(X^Ty)}$

issues/concerns : Needs an "inverse" computation $\sim \underline{O(d^3)}$

↳ Can use iterative procedures [Gradient descent !!]

$$w^{t+1} = w^t - \eta_t \underbrace{\nabla f(w^t)} \longrightarrow (X^TX)w - X^Ty$$

[does not involve an inverse]

$$\underline{\nabla F(w^t)} = \boxed{(\underline{X^T}X)w - \underline{X^T}y}$$

$\boxed{\underline{\text{Approximation of Gradient}}}$

- $\boxed{\text{Stochastic Gradient descent}}$
  - Samples a small set of data points uniformly at random.
  - Pretend the points sampled form the new dataset and compute gradient w.r.t w
- Can show that $\frac{1}{T}\sum_{t=1}^{T} w_t \longrightarrow w^*$.