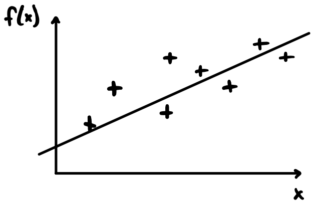
# Lecture 2 – Regression

Goal: learn real valued mapping $f: \mathbb{R}^d \to \mathbb{R}$

## Linear Regression




$x = [x_1, \dots, x_d]^T$, $f: \mathbb{R}^d \to \mathbb{R}$

$w = [w_1, \dots, w_d]^T$, $f(x) = \underline{\underline{w}}^T x$

1-dim: $y = f(x) = ax + b$

2-dim: $y = w_1 x_1 + w_2 x_2 + w_0$

d-dim: $y = w_1 x_1 + \dots + w_d x_d + w_0$

$= \sum_{i=1}^{d} w_i x_i + w_0$

$= w^T x + w_0$

## Homogeneous Representation

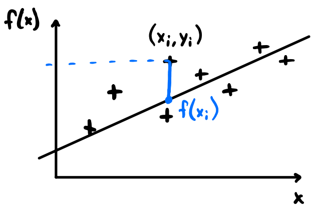
$\tilde{w} = [w_1, \dots, w_d, w_0] = [w^T, w_0]^T$

$\tilde{x} = [x_1, \dots, x_d, 1] = [x^T, 1]$

$f(\tilde{x}) = \tilde{w}^T \tilde{x}$

* this is just a way to simplify what we write by eliminating the need to separately tack on the $w_0$

## Quantifying goodness of fit




$D = \{(x_1, y_1), \dots, (x_n, y_n)\}$

$x_i \in \mathbb{R}^d \quad y_i \in \mathbb{R}$

Residual: $r_i = y_i - w^T x_i$

cost: $\hat{R}(\underline{\underline{w}}) = \sum_{i=1}^{n} r_i^2$

* notice

↳ 1: we square each residual

↳ 2: we sum the squares

least-squares...!

Least - squares linear regression optimization

↳ given data set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ $\quad x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$

↳ how do we find optimal weight vector?

## Method 1: Closed form solution

$$w^* = \arg\min_w \underbrace{\sum_{i=1}^{n} (y_i - w^T x_i)^2}_{\hat{R}(w)}$$

Empirical Risk Minimizer

derivation

1. $R(w) = \sum_{i=1}^{n} (y_i - w^T x_i)^2$
2. Take gradient with respect to $w$ and set to 0.

$$\nabla_w R(w) = -2X^T(Y - Xw)$$

$$-2X^T(Y - Xw) = 0$$

3. Rearrange...

$$X^T X w = X^T Y$$

4. Now, if $X^T X$ is invertible...

$$(X^T X)^{-1}(X^T X) w = (X^T X)^{-1}(X^T Y)$$

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

Time Complexity of $w^* = (X^T X)^{-1} X^T y$

$$\underbrace{(X_{n\times d}^T X_{n\times d})}_{O(nd^2)}{}^{\overset{O(d^3)}{-1}} \underbrace{X_{n\times d} \, y_{n\times 1}}_{O(nd)}$$

$$\Longrightarrow O(nd^2 + d^3)$$

Gradient Descent

**Method 2:** Optimization

The objective function $\hat{R}(w) = \sum_i (y_i - w^T x_i)^2$ is convex!

## Gradient Descent

initial guess for weight vector

- Start at arbitrary $w_0 \in \mathbb{R}^d$
- For $t = 1, 2, \ldots$ do $w_{t+1} = w_t - \eta_t \nabla \hat{R}(w_t)$

for each iteration, update the weight vector by taking a step in the opposite direction of the gradient of the objective function

learning rate (e.g. $\frac{1}{2}$ for least squares)

## Convergence

- converges to stationary point $\nabla = 0$
- optimal!
- $O(\log \frac{1}{\varepsilon})$ to get within $\varepsilon$ of the minimum of objective function.

**Time Complexity:** $O(nd)$ to compute gradient

## Adaptive Step Size

- line search
- "bold driver" heuristic
  - if function $\downarrow$, step size $\uparrow$
  - if function $\uparrow$, step size $\downarrow$

## Closed form vs Gradient Descent?

- computational complexity
- don't always need 100% optimal
- not all problems of closed form solution

Linear regression for polynomials

↳ we can fit non-linear functions via linear regression using non-linear features of our data

$$f(x) = \sum_{i}^{D} w_i \phi_i(x) \qquad x \xrightarrow{\phi} \hat{x} = \phi(x) \in \mathbb{R}^D$$

1-d: $\underset{z}{\phi(x)} = [1, x, x^2, \dots x^k]$, $k = D-1$

2-d: $\phi(x) := \phi([x_1, x_2]) = [1, x_1, x_2, x_1 x_2, x_1^2 x_2 \dots]$

⋮

p-dim $\phi(x) = O(p^k)$ vs $O(k^p)$