placeholder

1 Regression

1.1 Supervised Learning

Supervised Learning is the task of 'learning' a function relationship, based on a given set of inputs/outputs.

Some terminology:

$x \in \mathbb{R}^d$	Inputs (Attributes/Covariates)
$\phi(x) \in \mathbb{R}^p$	Features
$y \in \mathbb{R}$	Outputs (Targets/Labels)
$D = \{(x_i, y_i)\}_{i=1}^n$	Training Set
D'	Test Set
$f : \mathbb{R}^p \to \mathbb{R}$	Predictor (Model)
$l(f(x), y)$	Loss

Machine Learning Pipelines can often be classified using:

F	Function Class
$L(f)$	Training Loss
	Optimization Method

The function class F is a set of parametrized functions. We are looking for the $f \in F$ that minimizes $L(f)$.

D. **Training Loss**

$$L(f) := \frac{1}{n} \sum_{i=1}^{n} l\big(f(x_i), y\big)$$

1.2 Multiple Linear Regression

Multiple Linear Regression directly uses the $x \in \mathbb{R}^d$.
Here, $F_{\text{affine}} = \big\{ f(x) = w^\top x + w_0 \big| w \in \mathbb{R}^d, w_0 \in \mathbb{R} \big\}$.

Why are we using linear functions instead?

Any estimator $f \in F_{\text{affine}}$ can be rewritten as $f\big((x,1)\big) = (w, w_0)^\top \cdot (x, 1)$, thus we can augment the inpurs $x \mapsto (x, 1)$ and instead search in $F_{\text{linear}} = \{f(x) = \hat{w}^\top x | \hat{w} \in \mathbb{R}^{d+1}\}$

Loss Functions

D. **Squared Loss** $\quad l\big(f(x), y\big) := \big(f(x) - y\big)^2$

Most common Loss Function, but sensitive to outliers.

D. **Absolute Loss** $\quad l_{\text{abs}}\big(f(x), y\big) := |f(x) - y|$

Less sensitive to outliers, but not differentiable.

D. **Huber Loss**

$$l_{\text{huber}}\big(f(x), y\big) := \begin{cases} \frac{1}{2}\big(f(x) - y\big)^2 & |f(x) - y| \le \delta \\ \delta\big(|f(x) - y| - \frac{1}{2}\delta\big) & |f(x) - y| > \delta \end{cases}$$

Using parameter δ, the penalization of outliers can be controlled

Assymetric Loss: In some cases it is desirable to penalize overestimation harder than underestimation, or vice versa.

D. **Quantile Loss**

$$l_\tau\big(f(x), y\big) := \tau \max\Big\{y - f(x), 0\Big\} + (1-\tau) \max\Big\{f(x) - y, 0\Big\}$$

Using parameter τ, over/underestimation can be penalized

Linear Regression

To find $\hat{f} := \underset{f \in F_{\text{linear}}}{\arg\min} L(f)$ we just look for $w \in \mathbb{R}^d$.

$$\hat{w} := \underset{w \in \mathbb{R}^d}{\arg\min} L(f_w) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\Big(y_i - w^\top x_i\Big)^2}_{l\left(f(x_i), y_i\right)}$$

A natural abuse of notation here is $L(w) := L(f_w)$.

This can be rewritten in matrix notation:

$$\sum_{i=1}^{n} \Big(y_i - w^\top x_i\Big)^2 = \big\|y - Xw\big\|^2$$

The factor $\frac{1}{n}$ is irrelevant for Optimization, it doesn't depend on w

So we find the usual problem:

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\arg\min} \big\|y - Xw\big\|^2$$

The solution is a stationary point, so:

$$\nabla_w \big\|y - Xw\big\|^2 = 2X^\top (X\hat{w} - y) = 0$$

Which yields the known **Normal Equation**

$$X^\top X \hat{w} = X^\top y$$