

## Introduction to Machine Learning

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
$f: \mathbb{R}^p \rightarrow \mathbb{R}$	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

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

## 1.2 Multiple Linear Regression

**Multiple Linear Regression** directly uses the  $x \in \mathbb{R}^d$ .

Here,  $F_{\text{affine}} = \{f(x) = w^\top x + w_0 \mid w \in \mathbb{R}^d, w_0 \in \mathbb{R}\}$ .

Why are we using linear functions instead?

Any estimator  $f \in F_{\text{affine}}$  can be rewritten as  $f((x, 1)) = (w, w_0)^\top \cdot (x, 1)$ ,

thus we can augment the inputs  $x \mapsto (x, 1)$  and

instead search in  $F_{\text{linear}} = \{f(x) = \hat{w}^\top x \mid \hat{w} \in \mathbb{R}^{d+1}\}$

## Loss Functions

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

Most common Loss Function, but sensitive to outliers.

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

Less sensitive to outliers, but not differentiable.

**D. Huber Loss**

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

Using parameter  $\delta$ , 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**

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

Using parameter  $\tau$ , over/underestimation can be penalized

## Linear Regression

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

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

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

This can be rewritten in matrix notation:

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

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

So we find the usual problem:

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

The solution is a stationary point, so:

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

Which yields the **Normal Equation** from linear algebra.

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

## 2 Classification

In regression, we search an  $\hat{f} : \mathbb{R}^d \rightarrow \mathbb{R}$ , i.e.  $y, \hat{y} \in \mathbb{R}$ .  
In classification, we want  $\hat{y} \in \mathcal{Y} \subset \mathbb{R}$ , s.t.  $\mathcal{Y}$  is discrete.

### 2.1 Binary Classification

We generally use  $\mathcal{Y} = \{+1, -1\}$  and set  $\hat{y} = \text{sgn}(\hat{f}(x))$ .  
So, a linear classifier where  $\hat{f}(x) = w^\top x$  takes the form:

$$x \mapsto \begin{cases} 1 & w^\top x > 0 \\ -1 & w^\top x < 0 \end{cases}$$

**D. Decision Boundary**  $\{x \in \mathbb{R}^d \mid \hat{f}(x) = 0\}$

Like in regression, using features is again possible.

### 2.2 Surrogate Loss

We'd like to reuse the loss minimization from regression.  
A natural metric for accuracy is simply checking if  $\hat{y} = y$ .

**D. Zero-One Loss**

$$l_{0-1}(\hat{y}, y) := \mathbb{I}_{\hat{y} \neq y} = \begin{cases} 1 & \hat{y} \neq y \\ 0 & \hat{y} = y \end{cases}$$

We could try minimizing this:

$$\sum_{(x,y) \in \mathcal{D}} l_{0-1}(\hat{y}, y) = \sum_{(x,y) \in \mathcal{D}} \mathbb{I}_{f_w(x) \cdot y < 0}$$

Unfortunately,  $l_{0-1}$  is non-continuous and non-convex.  
We introduce *surrogate loss* to still apply GD.

Note how  $\mathbb{I}_{\hat{y} \neq y} = \mathbb{I}_{\hat{y} \cdot y < 0}$ , so  $l_{0-1}$  only depends on  $z := \hat{y} \cdot y$ .  
We thus define losses over  $z$ , that are cont. and convex.

**D. Surrogate Loss**

$$l_{\text{exp}} = e^{-z} \quad l_{\log} = \log(1 + e^{-z})$$

A notable difference is that  $l'_{\text{exp}}$  is unbounded,  
while  $l'_{\log} = \frac{1}{1+e^z} \in (-\frac{1}{2}, -1)$  for  $z < 0$ .  
This is better for outliers, thus  $l_{\log}$  is usually preferred.

## 2.3 Logistic Regression

We assume  $w_0 = 0$

We try to minimize  $l_{\log} = \log(1 + e^{-z})$ , so:

$$L(w) = \frac{1}{n} \sum_{i=1}^n l_{\log}(z_i) = \frac{1}{n} \sum_{i=1}^n \log\left(1 + e^{-\overbrace{y_i \cdot w^\top x_i}^{z_i}}\right)$$

Assume  $\{x_i, y_i\}_{i=1}^n$  is linearly separable, i.e.

$$\exists w \in \mathbb{R}^d : \underbrace{y_i \cdot w^\top x_i}_{z_i} > 0 \quad \forall i \leq n$$

Then there are multiple valid decision boundaries.

the distance  $x_0$  to the decision boundary is:  $\|x_0\|_2 \cdot |\cos(\theta)|$ .

$\theta$  between  $w, x_0 \in \mathbb{R}^d$

$$\|x_0\|_2 \cdot |\cos(\theta)| = \|x_0\|_2 \cdot \frac{|w^\top x_0|}{\|w\|_2 \cdot \|x_0\|_2} = \frac{|w^\top x_0|}{\|w\|_2}$$

Note if  $w$  is a unit-vector, this is just  $|w^\top x_0|$

**D. Margin**  $\text{margin}(w) := \min_{1 \leq i \leq n} y_i \cdot w^\top x_i$

## 2.4 Solutions

**D. Maximum Margin Solution**

$$w_{\text{MM}} := \max_{\|w\|_2=1} \min_{1 \leq i \leq n} (y_i \cdot w^\top x_i)$$

If  $\mathcal{D}$  is linearly separable, this is convex.

**D. Support Vector Machine**

$$w_{\text{SVM}} := \min_{w \in \mathbb{R}^d} \|w\|_2 \quad \text{s.t.} \quad y_i \cdot w^\top x_i \geq 1 \quad \forall i \leq n$$

Solving these problems is actually equivalent, up to scaling:

**L.**  $\frac{w_{\text{SVM}}}{\|w_{\text{SVM}}\|_2} = w_{\text{MM}}$  (This also holds for the case  $w_0 \neq 0$ )