Classification

data: (X, y) : y: discrete set

$y = \{c_1, c_2\}$, $\{0,1\}$, $\{-1,1\}$

(Multiclass: $\{c_1, c_2, c_3 \dots\}$ ...)

$Y = \{c_1, c_2\}$: Binary classification

(no ordering on classes)

First example: use L-2 regression

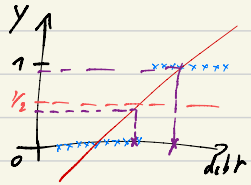
Binary classification Pb like $\{0,1\}$

Given some data S → fit a regression fct → if $f(x) > 1/2$ then predict 1

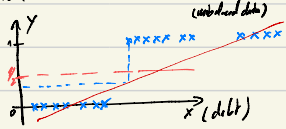
if $f(x) < 1/2$ then predict 0

i.e.: S = credit card data (X, Y)

X : amount outstanding debt, $Y = \begin{cases} 1 & \text{if the person default at the next payment} \\ 0 & \text{o.w} \end{cases}$

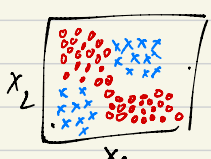


- However, classification is not only a special case of Lin. Reg:
  - The prediction is not continuous
  - Sensitivity to unbalanced data:



(Classification doesn't depend on L-2 loss, neither on accuracy)

Nearest Neighbor classifier: (KNN...)

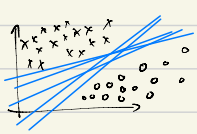


- The fundamental task of classification is to separate the space into decision regions

⇒ We can get some very complex boundaries

⇒ This will only work on low dimensions. (Neighbor in High dimension?)

Linear decision boundaries:



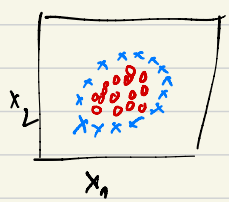
- We assume that the data is linearly separable

- minimal distance of the points to the hyperplane → margin

Idea: Maximize the margin (SVM...)

. What if the data is not linearly separable?

Non linear classifier:



- We can do feature augmentation $(x, x^2, x^3, x^4 \dots)$
- Then do linear classification
  → Can be computationally exp
- We can use the Kernel method too

Baye's prediction: (MAP rule)

$$\hat{y}(x) = \underset{y \in \mathcal{Y}}{\operatorname{arg\,max}}\ \mathbb{P}(y \mid X)$$

$$\left(\hat{y}(x) = \underset{g : \mathcal{X} \to \mathcal{Y}}{\operatorname{arg\,min}}\ \mathcal{L}_D(g)\right)$$

$$\mathbb{P}(\hat{y}(X) = y) = \int p(x)\, p(\hat{y}(x) \mid X)\, dx$$

↳ prob of correct guess is the average (over all x)

Thm: (Reg ↔ classif)

let $\eta : \mathcal{X} \to \mathbb{R}$ a reg. fct

$g_\eta : \mathcal{X} \to \{0,1\}$, $x \mapsto \mathbb{1}_{\eta(x) > 1/2}$

$$\mathcal{L}^c = \mathbb{E}_D[\mathbb{1}_{g(x) \neq y}], \quad \mathcal{L}^2 = \mathbb{E}_D[(Y - \eta(x))^2]$$

$$\mathcal{L}^c(g_\eta) - \mathcal{L}^c(g^*_\eta) \leq 2\sqrt{\mathcal{L}^2(\eta) - \mathcal{L}^2(\eta^*)}$$

Logistic Regression

Binary classification:

$(X, Y) \sim D$

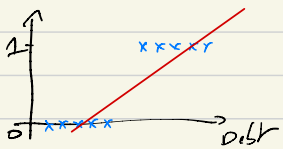
X features $\in \mathcal{X}$

Y Labels $\in \mathcal{Y} \in \{0, 1\}$

Data $S = \{(x_n, y_n)\}_{n=1}^{N}$ iid

Given a new X we want to predict y

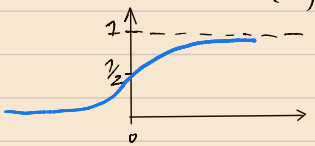
Motivation:



Map $(-\infty, +\infty)$ to $[0, 1]$

Logistic function:

$$\sigma(x) = \frac{e^x}{1+e^x}$$



$$1 - \sigma(x) = (1+e^x)^{-1}$$

$$\sigma'(x) = \frac{e^x}{(1+e^x)^2} = \sigma(x) \cdot (1 - \sigma(x))$$

Logistic regression: $\mathbb{P}(y=1|x) = \sigma(x^T w)$

$\mathbb{P}(y=0|x) = 1 - \sigma(x^T w)$ } Regression step

. logistic because of σ, Regression because

- Quantize (step): if $\mathbb{P}(y=1|x) \geq 1/2$ predict 1

  if $\mathbb{P}(y=1|x) < 1/2$ predict 0

We assume that $x = \begin{pmatrix} 1 \\ x \end{pmatrix}$

(Scale w : $\|w\|$: fast or slow "slope"; w_0 : shift the transition (decision region) along w)

$\|w\| \to \infty$, $\sigma \to$ step fct

Likelihood criterion:

$S_{Train} = \{(x_i, y_i)\}_{i=1}^N$ iid ; $w^* = \arg\max_w \mathbb{P}(y, x \mid w)$

$= \arg\max_w \mathbb{P}(x \mid w)\,\mathbb{P}(y \mid x, w)$

Fact 1: $w \perp\!\!\!\perp x \Rightarrow \mathbb{P}(x \mid w) = \mathbb{P}(x)$

$w^* = \arg\max_w \mathbb{P}(y \mid x, w)$

$= \arg\max_w \prod_{i=1}^N \mathbb{P}(y_i \mid x_i, w)$ $\quad ((x_n, y_n)_{n=1}^N$ iid)

$= \arg\max_w \prod_{i=1}^N \sigma(x_i^T w)^{y_i} (1-\sigma(x_i^T w))^{1-y_i}$

- We prefer minimizing and summation $\Rightarrow$ minimize the negative log-likelihood

$w^* = \arg\min_w -\sum_{i=1}^N y_i \log(\sigma(x_i^T w)) + (1-y_i)\log(1-\sigma(x_i^T w))$

$= \arg\min_w \sum_{i=1}^N y_i \log\left(\frac{\sigma(x_i^T w)}{1-\sigma(x_i^T w)}\right) - \log(1-\sigma(x_i^T w))$

$\left(1-\sigma(y) = \frac{1}{1+e^y}\right)$ $\left(\frac{\sigma(y)}{1-\sigma(y)} = e^y\right)$

$w^* = \arg\min_w \sum_{i=1}^N -y_i \cdot x_i^T w + \log(1+e^{x_i^T w})$

$$\ell(w) = \sum_{i=1}^N \log(1+e^{x_i^T w}) - y_i x_i^T w$$ for $y \in \{0,1\}$

Gradient of $\ell(w)$:

$\nabla \ell(w) = \sum_{i=1}^N \frac{e^{x_i^T w}}{1+e^{x_i^T w}} \cdot x_i - y_i x_i = \sum_{i=1}^N x_i \underbrace{(\sigma(x_i^T w) - y_i)}_{D\times 1}$

$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix}_{N\times D}$ $Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}_{N\times 1}$

$$\nabla \ell(w) = X^T[\sigma(Xw) - Y]$$

- Same Gradient as LS but with $\sigma(\cdot)$ $(\nabla \ell(w) = X^T(Xw - Y))$
- No closed form sol: $\nabla \ell(w) = 0$ for LR
- Good news: ℓ is convex

How to prove it? . Compute the Hessian
. Look at oper. which preserves convexity (next slide)

Convexity: Rules of basic operations: . positive comb. ($\alpha f + \beta g$ is convex)
. composition: f is convex and g affine $\Rightarrow f \circ g$ is convex
. Inf. of f's both convex and concave

Hessian of ℓ:

$\nabla \ell(w) = \sum_{i=1}^N x_i(\sigma(x_i^T w) - y_i)$

$\nabla^2 \ell(w) = \sum_{i=1}^N x_i \nabla(\sigma(x_i^T w))^T$

$= \sum_{i=1}^N x_i x_i^T \nabla(x_i^T w)(1-\sigma(x_i^T w))$

$= \sum_{i=1}^N x_i \sigma(x_i^T w)(1-\sigma(x_i^T w)) x_i^T$

$= X^T S X$ where $S = \mathrm{diag}[\sigma(x_i^T w)(1-\sigma(x_i^T w))]_{n,n}$

since $S \geq 0$, $\nabla^2 \ell(w) \geq 0$ (ℓ is convex)

- $y \mapsto \log(1+e^y)$ is convex?
- $(\log(1+e^y))' = \frac{e^y}{1+e^y} = \sigma(y)$
- $\frac{\partial^2 \log(1+e^y)}{\partial y^2} = \sigma(y)' = \sigma(y)(1-\sigma(y)) \geq 0$
- $\Rightarrow \log(1+e^y)$ is convex

$\Rightarrow \ell$ is convex

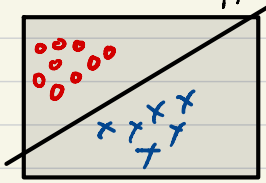
• How to minimize $\mathcal{L}$? => Gradient descent:

$w_0 \in \mathbb{R}^D$

$$w_{k+1} = w_k - \gamma_{k+1} \nabla \mathcal{L}(w_k)$$

SGD: $w_{k+1} = w_k - \gamma_{k+1} \nabla \mathcal{L}_n(w_k)$

-> What will happen if the data are linearly separable?



$$\mathcal{L}(w) = \sum_{i=1}^{N} \log(1 + e^{x_i^T w}) - y_i x_i^T w$$

$\min \mathcal{L}(w)$?

$\inf \mathcal{L}(w) = 0 = \lim_{|w| \to \infty} \mathcal{L}(w)$

The limit is not obtained by a finite w

=> To avoid this we add L-2 regularization

$$\mathcal{L}(w) = \sum_{n=1}^{N} \log(1 + e^{x_n^T w}) - y_n x_n^T w + \lambda \|w\|_2^2$$

Newton alg. (2nd order Alg.) (for small N, D)

$\hat{w}$: current estimate

$\mathcal{L}(w) \sim \underbrace{\mathcal{L}(\hat{w}) + \nabla \mathcal{L}(\hat{w})(w - \hat{w}) + (w - \hat{w})^T \nabla^2 \mathcal{L}(\hat{w})(w - \hat{w})}_{P(w)} + O(\|w - \hat{w}\|^3)$

next pt: w^+ = arg min quad. approx.

$\nabla P(w) = \nabla \mathcal{L}(\hat{w}) + \nabla^2 \mathcal{L}(\hat{w})(w - \hat{w}) = 0$

-> $w^+ = \hat{w} - \nabla^2 \mathcal{L}(\hat{w})^{-1} \nabla \mathcal{L}(\hat{w})$ $\Rightarrow$ $w_{k+1} = w_k - \gamma_{k+1} \nabla^2 \mathcal{L}(w)^{-1} \nabla \mathcal{L}(\hat{w})$

• If our fct is quad. we will minimize it with one step of Newton Method

$$\mathcal{L}(w) = \sum_{n=1}^{N} \log(1 + e^{x_n^T w}) - y_n x_n^T w \qquad \text{if } y_n' \in \{-1, 1\} \hookrightarrow y_n \in \{0, 1\}$$

$$= \sum_{n=1}^{N} \log\left(\frac{1 + e^{x_n^T w}}{e^{y_n' x_n^T w}}\right) = \sum_{n=1}^{N} \log\left(1 + e^{-y_n' x_n^T w}\right) \qquad y_n' = 2y_n - 1, \quad y_n = \tfrac{1}{2}(y_n' + 1)$$