Lecture 04 | Logistic Regression

Readings: Murphy: 10–10.2.4, 10.3–10.3.3.
Abu-Mostafa: 3.3

Classification

Learn $f: X \to Y$ $\quad X \in \mathbb{R}^d$ features; $Y = \{1, 2, \dots, K\}$ target classes.

"0-1 Loss Function": $\boxed{\ell(f(x), y) = 1\{f(x) \neq y\}}$

Expected Loss:

$$E_{XY}[1\{f(x) \neq y\}] \overset{\text{law of total expectation (tower property)}}{=} E_X\left[E_{Y|X}[1\{f(x) \neq y\} \mid X = x]\right]$$

$$\overset{\text{Expectation of indicator is probability of event.}}{=} E_X\left[P(f(x) \neq y \mid X = x)\right]$$

$$= 1 - P(f(x) = y \mid X = x)$$

$$= E_X\left[1 - P(f(x) = y \mid X = x)\right]$$

To minimize expected loss, we must maximize:

$$\boxed{f^*(x) = \underset{y \in Y}{\operatorname{argmax}}\ P(Y = y \mid X = x), \quad \forall x}$$

Bayes optimal classifier

Bayes optimal classifier

$$f^*(x) = \arg\max_y \underbrace{P(Y=y \mid X=x)}$$

In practice, we do not know it.
We only get n iid samples $\{(x_i, y_i)\}_{i=1}^n$

- Suppose X is **discrete** so that $x \in \{1, 2, \dots, m\}$
  What is a natural estimator for $P(Y=y \mid X=x)$?

$$\hat{f}(x) = \arg\max_y \frac{\sum_{i=1}^n 1\{x_i=x, y_i=y\}}{\sum_{i=1}^n 1\{x_i=x\}}$$

- if X is **continuous**, we need a **model** to explain observations

Maximum Likelihood Estimation (MLE) for Classification

$$f^*(x) = \arg\max_y P(Y=y \mid X=x)$$
$$\hat{f}(x) = \arg\max_y P_w(Y=y \mid X=x)$$

General MLE procedure:

(1) Parameterize $P_w(Y=y \mid X=x)$ as a function of w

(2) Learn w on iid training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$

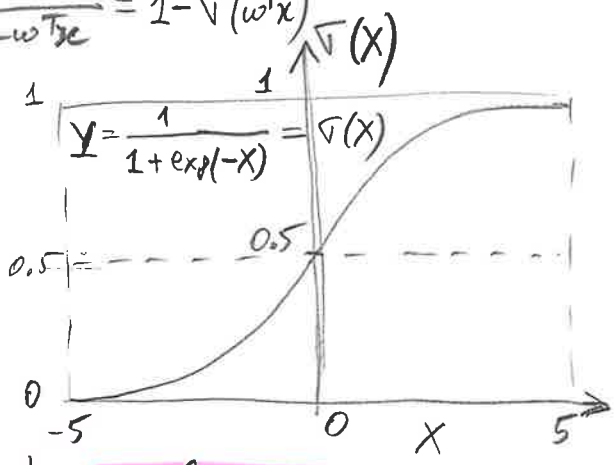
$$\hat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P_w(y_i \mid x_i)$$
$$= \arg\max_w \sum_{i=1}^n \log\left(P_w(y_i \mid x_i)\right)$$

maximize Likelihood of training data!

Modeling conditional probabilities

- Recall **linear regression**:
  - $P_w(Y=y|X=x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(y-w^Tx)}{2\sigma^2}}$
  - Prediction: $\mathbb{E}[Y|X=x] = w^Tx$
- In **Logistic regression**, we use a **specialized model** for **binary** classification:

Posterior for class "1" →
- $P_w(Y=1|X=x) = \frac{1}{1+e^{-w^Tx}} = \sigma(w^Tx)$: pass w^Tx through **sigmoid** function σ.
- $P_w(Y=0|X=x) = \frac{e^{-w^Tx}}{1+e^{-w^Tx}} = 1-\sigma(w^Tx)$



- Prediction: just round sigmoid(w^Tx)

Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ iid, $x_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$ for notation convenience!!!

$$\hat{w}_{MLE} = \arg\max_w \sum_{i=1}^n \log P_w(y_i|x_i)$$

use fact that: $1-\sigma(s) = \sigma(-s)$

$$P_w(y_i|x_i) = \sigma(y_i w^T x_i)$$

can be ± 1

$$\hat{w}_{MLE} = \arg\min_w \sum_{i=1}^n \log\left(1 + e^{-y_i w^T x_i}\right)$$

To compute, there is no closed form.
But, it is a smooth convex problem!
⇒ Compute via **gradient descent!** (GD)

(See Abu-Mostafa 3.3)

Multi-class classification

- So for binary: $y \in \{-1, 1\}$
- Generally we can have k classes: $y \in \{c_1, c_2, \ldots, c_k\}$

Classes / Labels

- A k-class classifier predicts y given x.

Encoding Categorical Labels c_j

- Need to embed raw c_j into real-valued vectors
- One-hot encoding / embedding is very common: each class is a standard basis vector in $\mathbb{R}^k$

example:

Classes country of Birth:
{Argentina, Brazil, USA, ...}
196 categories

1-hot encoding ⟹

x	$h_1(x)$	$h_2(x)$	...	$h_{195}(x)$	$h_{196}(x)$
Argentina	0	0		0	1
Brazil	0	0		1	0
USA	0	0	1	0	0
⋮					
Zimbabwe	1	0		0	0

Multi-class Logistic Regression

Data: $x_i \in \mathbb{R}^d$, categorical $y \in \{c_1, c_2, \ldots, c_k\}$
One hot encoding s.t. $y = [1\ 0\ 0 \ldots 0] \in \mathbb{R}^k$
means/implies $y = c_1$

Model: Linear prediction: $\hat{y}_i = f(x_i) = \text{softmax}(w^T x_i) \in \mathbb{R}^k$
Parameter matrix $w \in \mathbb{R}^{d \times k}$

$$\begin{bmatrix} — & w_1^T & — \\ — & w_2^T & — \\ & \vdots & \\ — & w_k^T & — \end{bmatrix} \cdot \begin{bmatrix} | \\ x_i \\ | \end{bmatrix} = \begin{bmatrix} | \\ \text{logits} \\ | \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} 0.4 \\ 0.3 \\ \vdots \end{bmatrix}$$

normalized that sum = 1

$w^T \in \mathbb{R}^{k \times d}$ $\quad x_i \in \mathbb{R}^{d \times 1}$ $\quad \text{logits} \in \mathbb{R}^{k \times 1}$ $\quad \hat{y} \in \mathbb{R}^k$

Softmax Classification

Binary: 2 classes $y \in \{-1, 1\}$

Multi-class: k classes $y \in \{c_1, c_2, \cdots, c_k\}$

Conditional Probabilities:

$$P(y=1|x) = \frac{1}{1+e^{-\omega^T x}} = \frac{e^{\omega^T x}}{1+e^{\omega^T x}}$$

$$P(y=-1|x) = \frac{1}{1+e^{\omega^T x}}$$

$$P(y=c_j|x) = \frac{e^{\omega_j^T x}}{\sum_{j'} e^{\omega_{j'}^T x}}$$

MLE:

$$\arg\max_{\omega} \sum_{i=1}^{n} \log\left(\frac{1}{1+e^{-y_i \omega^T x_i}}\right)$$

$$\arg\max_{\omega} \sum_{i=1}^{n} \sum_{j=1}^{k} 1\{y_i = c_j\} \cdot \log\left(\frac{e^{\omega_j^T x_i}}{\sum_{j'=1}^{k} e^{\omega_{j'}^T x_i}}\right)$$

Why Softmax – can be easily interpreted as probabilities

$$\begin{bmatrix} 1 \\ 10 \\ 2 \end{bmatrix} \to \begin{bmatrix} \text{small} \\ \text{big} \\ \text{small} \end{bmatrix} \tilde{\to} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} \to \begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \end{bmatrix}$$

Regression and Classification

ML paradigm: { define predictor $f_w(x)$ and Loss/Cost $\ell(f_w(x), y)$
Then, optimize $\hat{\omega} = \arg\min_{\omega} \sum_{i=1}^{n} \ell(f_w(x_i), y_i)$

Regression – squared error loss: $\ell(f_w(x), y) = (y - f_w(x))^2$

Logistic loss: $\ell(f_w(x), y) = \log\left[1 + \exp(-y f_w(x))\right]$