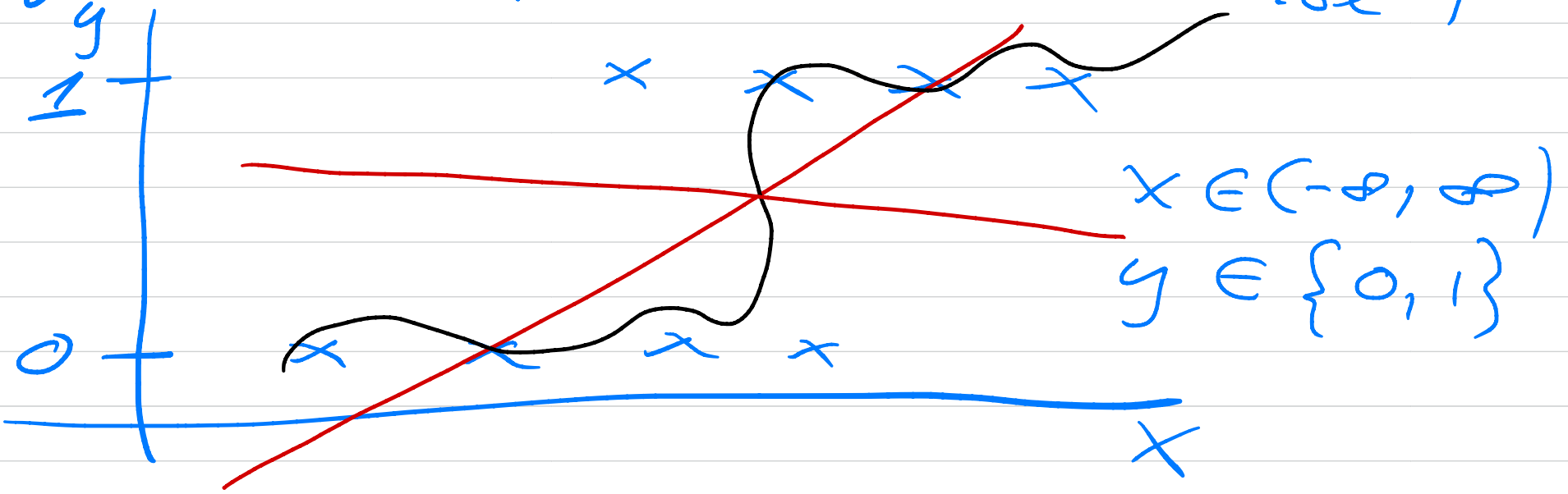
Erasmus+ lecture on machine learning, November 6, 2023

Logistic Regression (Binary case)



linear regression

$$y_i \cong \sum_j x_{ij} \beta_j + \varepsilon_i$$

continuous

$$\varepsilon_i \sim N(0, \sigma^2)$$

$$\sum_j x_{ij} \beta_j = f(x_i) \Rightarrow y_i = f(x_i) + \varepsilon_i$$

In linear regression

$$y_i \in (-\infty, \infty) \wedge x_i \in (-\infty, \infty)$$

Logistic regression & classification problems, we want $f(x_i)$ to represent discrete outputs, e.g. binary

$$y_i = \{0, 1\}$$

$$f(x_i) \rightarrow p(x_i) = p_i$$

$$p_i = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} = \frac{1}{1 + e^{-\beta_0 - \beta_1 x_i}}$$

Sigmoid function

if $\underline{p(x)} = \sigma(x) = \frac{1}{1+e^{-x}}$

if $p(x) > 0.5$, then $\hat{y} = \underline{1}$

else $p(x) \leq 0.5$, then $\hat{y} = 0$

$$\int_{x \in D} p(x)\,dx = \underline{1}$$

our assumption

$$y(x) = p(x) + \varepsilon$$

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

$$D = \{(x_0, y_0), (x_1, y_1) \dots (x_{n-1}, y_{n-1})\}$$

$$\left\{ p(x) = \frac{e^{\beta_0 + \beta_1 x + \beta_2 x + \dots}}{1 + e^{\beta_0 + \beta_1 x + \dots}} \right\}$$

$$p(x) \rightarrow P(y_i' = 1, x_i' / \beta)$$

$$= \frac{e^{\beta_0 + \beta_1 x_i'}}{1 + e^{\beta_0 + \beta_1 x_i'}} = p_i'$$

$$P(y_i = 0, x_i \mid \beta) = 1 - p_i$$

$$\sum_{i=0,1} P(y_i, x_i) = 1$$

Assume that y_i are independent and identically distributed $(i.i.d.)$

$$P(D \mid \beta) = \prod_{i=0}^{n-1} P(y_i, x_i \mid \beta)$$

what probability distribution does ε follow?

$$y_i = 1 = p_i + \varepsilon_i$$

$$\varepsilon_i = 1 - p_i$$

$$y_i = 0 \Rightarrow \varepsilon_i = -p_i \quad (p_i \Rightarrow p)$$

$$\mathbb{E}[\varepsilon] = \sum_i \varepsilon_i (p_i)$$

$$= (1-p)p + (-p)(1-p)$$

$$= 0$$

$$\text{var}[\varepsilon] = (1-p)^2 p + (-p)^2 (1-p)$$

$$= p(1-p) \Rightarrow$$

$\varepsilon \sim$ Binomial distribution with mean 0 and variance $p(1-p)$

$$P(D|\beta) = \prod_{i=0}^{n-1} p_i^{y_i} (1-p_i)^{1-y_i}$$

Maximum likelihood estimation (MLE)

$$\hat{\beta} = \arg\max_{\substack{\beta \\ \beta \in \mathbb{R}^p}} P(D|\beta)$$

cost function $C(\beta) = P(D|\beta)$

$$\vec{\nabla}_\beta C(\beta) = 0$$

Equivalent problem

$$\hat{\beta} = \arg\max_{\beta} \log P(D|\beta)$$

Or

$$\hat{\beta} = \arg\min_{\beta} \; -\log P(D/\beta)$$

cross entropy

$$C(\beta) = -\sum_{i=0}^{n-1} \left[y_i' \log p_i' + (1-y_i') \log (1-p_i') \right]$$

$$\left(p_i' = \frac{e^{\beta_0 + \beta_1 x_i'}}{1 + e^{\beta_0 + \beta_1 x_i'}} \right)$$

$$= -\sum_{i=0}^{n-1} \left[y_i' (\beta_0 + \beta_1 x_i') - \log (1 + e^{\beta_0 + \beta_1 x_i'}) \right]$$

$$C(\beta) = -\sum_{i=0}^{n-1} \left[y_i (\beta_0 + \beta_1 x_i) - \log(1 + e^{\beta_0 + \beta_1 x_i}) \right]$$

$$\frac{\partial C}{\partial \beta_0} = 0 = -\sum_{i=0}^{n-1} (y_i - p_i) = g_0$$

$$\frac{\partial \log(1 + e^{\beta_0 + \beta_1 x_i})}{\partial \beta_0} = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} = p_i$$

$$\frac{\partial C}{\partial \beta_1} = 0 = -\sum_{i=0}^{n-1} x_i (y_i - p_i) = g_1$$

$$\frac{\partial C}{\partial \beta} = -X^T(y - p) = X^T(p - y) = g$$

$X^T \in \mathbb{R}^{p \times n}$ $\quad y, p \in \mathbb{R}^n$

$g \in \mathbb{R}^p$

$$\frac{\partial^2 C}{\partial \beta \partial \beta^T} = H = X^T W X$$

$$W = \begin{cases} p_i(1 - p_i) = w_{ij} & i = j \\ w_{ij} = 0 & i \neq j \end{cases}$$

Linear regression (OLS)

$$\beta = (X^T X)^{-1} X^T y$$

$$g = X^T(p - y) = 0$$

no β-dependence

$$p(\beta) = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)}$$

non-linear dependence on β.

NO analytical solution for β

Numerical solutions using iterative schemes like Newton-Raphson

Taylor expand $C(\beta)$ around the optimal value $\hat{\beta}$, $\hat{\beta} - \beta^{(n)}$

$$C(\hat{\beta}) = C(\beta^{(n)}) + (g^{(n)})^T(\hat{\beta} - \beta^{(n)}) + \frac{1}{2}(\hat{\beta} - \beta^{(n)})^T H^{(n)}(\hat{\beta} - \beta^{(n)}) + \dots$$

$$\left.\frac{\partial C}{\partial \beta}\right|_{\beta = \beta^{(n)}} = -g^{(n)}$$

$$\left.\frac{\partial^2 C}{\partial \beta \partial \beta^T}\right|_{\beta = \beta^{(n)}} = H^{(n)}$$

$$b = \hat{\beta} - \beta^{(n)}$$

$$C(\vec{\beta}) \simeq C(\beta^{(n)}) + (g^{(n)})^T b$$

$$+ \frac{1}{2} b^T H^{(n)} b$$

$$\left(f(x) = \text{const} + \gamma^T x + \frac{1}{2} x^T A x \right)$$

$$\frac{\partial f}{\partial x} = 0 \Rightarrow Ax = \gamma$$

$$\frac{\partial C}{\partial b} = g^{(n)} + H^{(n)} b = 0$$

$$b = \vec{\beta} - \beta^{(n)}$$

$$\vec{\beta} = \beta^{(n+1)} = \beta^{(n)} - \left[H(\beta^{(n)}) \right]^{-1}$$

Newton-Raphson iterative scheme $\times$ $g(\beta^{(n)})$

$$\beta^{(n+1)} = \beta^{(n)} - \left[H(\beta^{(n)})\right]^{-1} \left(X^T(p(\beta^{(n)}) - y\right)$$

Bottleneck of all optimization methods resides in $H(\beta^{(n)})$ and its inverse,

In Neural networks it is common to have at least 10^5 or more parameters $\Rightarrow$

$$H \in \mathbb{R}^{10^5 \times 10^5}$$

$H \rightarrow$ replace with constant

η = Learning rate

$$\beta^{(n+1)} = \beta^{(n)} - \gamma^{(n)} g^{(n)}$$

constant

Adagrad

RMSprop

ADAM — 10^5 citations

⋮