Important Announcements and Notes (Nov 25)

- Review cross entropy loss intuition

Important Announcements and Notes (Nov 21)

- No tutorial today

Remaining topics:

- Nov 21: Binary Classification
- Nov 26: Multiclass Classification

} assignment 7

- Nov 28: Neural Networks
- Dec 3: Language Models
- Dec 5: Exam/Course Review

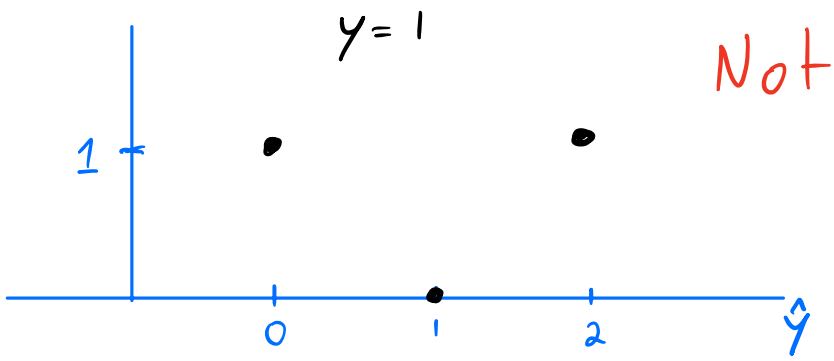
Classification

Labels are unordered (and usually finite)

Ex: Types of wine $\mathcal{Y} = \{\text{Barolo}, \text{Grignolino}, \text{Barbera}\}$
$= \{0, 1, 2\}$

Loss ℓ is 0-1 loss

$$\ell(\hat{y}, y) = \begin{cases} 0 & \text{if } \hat{y} = y \\ 1 & \text{otherwise} \end{cases}$$



Not continuous

$$\hat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\vec{x}_i), y_i)$$

Not continuous

$\Rightarrow$ Hard to optimize

$$f_{Bayes} = \underset{f \in \{f \mid f: \mathcal{X} \to \mathcal{Y}\}}{\operatorname{argmin}} \; L(f)$$

$$L(f) = \mathbb{E}[\ell(f(\vec{X}), Y)]$$

$$f_{Bayes}(\vec{x}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \; p(y \mid \vec{x})$$

Binary Classification $y = \{0, 1\}$

MLE to estimate pmf $p(y|\vec{x})$

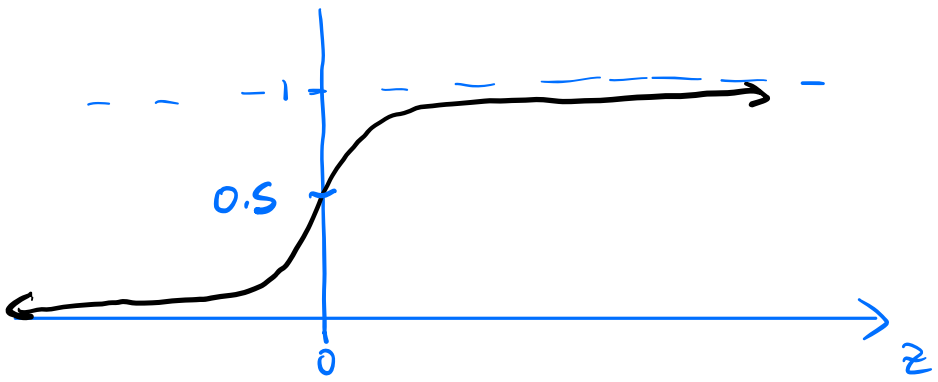
$D = ((\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n))$

$(\vec{X}_i, Y_i)$ are i.i.d. with $\mathbb{P}_{\vec{X},Y}$, $p_{\vec{X},Y}$

Assume $Y_i | \vec{X}_i = \vec{x}_i \sim \text{Bernoulli}(\alpha^*(\vec{x}_i)) = \text{Bernoulli}(\sigma(\vec{x}_i^T \vec{w}^*))$

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

"sigmoid" or "logistic"



$$p(y|\vec{x}) = \left(\sigma(\vec{x}^T \vec{w}^*)\right)^y \left(1 - \sigma(\vec{x}^T \vec{w}^*)\right)^{(1-y)}$$

$$\vec{w}_{MLE} = \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmax}} \; p(D \mid \vec{w})$$

$$= \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmax}} \; \prod_{i=1}^{n} p(\vec{x}_i, y_i \mid \vec{w})$$

$$= \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \; -\sum_{i=1}^{n} \log\left(p(\vec{x}_i, y_i \mid \vec{w}) \right)$$

$$= \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \; -\sum_{i=1}^{n} \left[\log\left(p(y_i \mid x_i, \vec{w}) \right) + \log\left(p(x_i) \right) \right]$$

$$= \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \; -\sum_{i=1}^{n} \log\left(p(y_i \mid x_i, \vec{w}) \right)$$

$$= \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \; -\sum_{i=1}^{n} \log\left(\left(\sigma(\vec{x}_i^T \vec{w}) \right)^{y_i} \left(1 - \sigma(\vec{x}_i^T \vec{w}) \right)^{(1-y_i)} \right)$$

$$= \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \; -\sum_{i=1}^{n} \left[\log\left(\left(\sigma(\vec{x}_i^T \vec{w}) \right)^{y_i} \right) + \log\left(\left(1 - \sigma(\vec{x}_i^T \vec{w}) \right)^{(1-y_i)} \right) \right]$$

$\log(x^a) = a \log(x)$

$= g(\vec{w})$ convex

$$= \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \; -\sum_{i=1}^{n} \left[y_i \log\left(\sigma(\vec{x}_i^T \vec{w}) \right) + (1-y_i) \log\left(1 - \sigma(\vec{x}_i^T \vec{w}) \right) \right]$$

$= h_i(v_i)$ $\qquad$ $r_i(v_i)$

$u_i = \vec{x}_i^T \vec{w}$, $v_i = \sigma(u_i)$

$$\frac{\partial g}{\partial w_j}(\vec{w}) = -\sum_{i=1}^{n} \left[\frac{dh_i}{dv_i} \frac{dv_i}{du_i} \frac{\partial u_i}{\partial w_j} + \frac{dr_i}{dv_i} \frac{dv_i}{du_i} \frac{\partial u_i}{\partial w_j} \right]$$

$$\frac{dh_i}{dv_i} = y_i \frac{1}{v_i} = \frac{y_i}{\sigma(u_i)}, \quad \frac{dr_i}{dv_i} = -(1-y_i)\frac{1}{1-v_i} = -\frac{1-y_i}{1-\sigma(u_i)}$$

$$\frac{dv_i}{du_i} \overset{\text{exercise}}{=} \sigma(u_i)(1-\sigma(u_i)), \quad \frac{\partial u_i}{\partial w_j} = x_{ij}$$

$$= -\sum_{i=1}^{n}\left[\frac{y_i}{\sigma(u_i)}\sigma(u_i)(1-\sigma(u_i))x_{ij} - \frac{(1-y_i)}{(1-\sigma(u_i))}\sigma(u_i)(1-\sigma(u_i))x_{ij}\right]$$

$$= -\sum_{i=1}^{n}\left[y_i(1-\sigma(u_i))x_{ij} - (1-y_i)\sigma(u_i)x_{ij}\right]$$

$$= -\sum_{i=1}^{n}\left[y_i x_{ij} - y_i\sigma(u_i)x_{ij} - \sigma(u_i)x_{ij} + y_i\sigma(u_i)x_{ij}\right]$$

$$= -\sum_{i=1}^{n}\left(y_i - \sigma(u_i)\right)x_{ij}$$

$$\frac{\partial g}{\partial w_j}(\vec{w}) = \sum_{i=1}^{n}\left(\sigma(\vec{x}_i^T\vec{w}) - y_i\right)x_{ij} \qquad j \in \{0,\ldots,d\}$$

No closed form solution

Use gradient descent

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - \eta^{(t)}\nabla g(\vec{w}^{(t)})$$

$$\nabla g(\vec{w}) = \sum_{i=1}^{n} (\sigma(\vec{x}_i \vec{w}) - y_i)\, \vec{x}_i \in \mathbb{R}^{d+1}$$

$$\vec{w}_{MLE} \approx \vec{w}^{(T)}$$

$$f_{MLE}(\vec{x}) = \sigma(\vec{x}^T \vec{w}_{MLE}) \approx \alpha^*(\vec{x})$$

Binary Classification Learner

$$\mathcal{A}(D) = \hat{f}_{Bin}$$

$$f_{Bayes}(\vec{x}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\ p(y|\vec{x})$$

$$\approx \underset{y \in \{0,1\}}{\operatorname{argmax}}\ p(y|\vec{x}, \vec{w}_{MLE})$$

$$= \hat{f}_{Bin}(\vec{x})$$

$$p(y=1|\vec{x}, \vec{w}_{MLE}) = f_{MLE}(\vec{x})$$

$$p(y=0|\vec{x}, \vec{w}_{MLE}) = 1 - f_{MLE}(\vec{x})$$

$$= \begin{cases} 1 & \text{if } f_{MLE}(\vec{x}) \geq 0.5 \\ 0 & \text{if } f_{MLE}(\vec{x}) < 0.5 \end{cases}$$

$$= \begin{cases} 1 & \text{if } \vec{x}^T \vec{w}_{MLE} \geq 0 \\ 0 & \text{if } \vec{x}^T \vec{w}_{MLE} < 0 \end{cases}$$

$\vec{x}^T \vec{w}_{MLE} = w_0 + x_1 w_1 + \ldots + x_d w_d = 0$

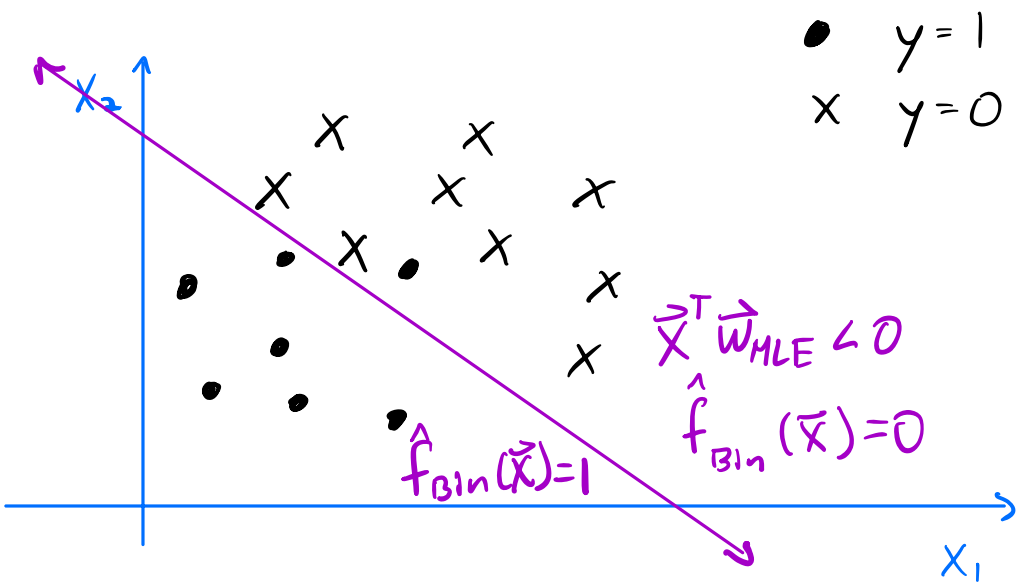
is a d-1 dimensional hyperplane

Ex: $d = 2$

$\vec{x}^T \vec{w}_{MLE} = 0$

$w_0 + x_1 w_1 + x_2 w_2 = 0$

$x_2 = -x_1 \frac{w_1}{w_2} - \frac{w_0}{w_2}$



Polynomial features apply like before

MAP implying Regularization like before

Logistic Regression

different y from below

$\mathcal{Y} = [0,1]$ representing values of $p(y=1 \mid \vec{x}) = \alpha^*(\vec{x})$

$\ell(f(\vec{x}), y) = -\left[y \log(f(\vec{x})) + (1-y)\log(1-f(\vec{x}))\right]$ "cross-entropy loss"

$\mathcal{F} = \{f \mid f : \mathbb{R}^{d+1} \to [0,1] \text{ and } f(\vec{x}) = \sigma(\vec{x}^T \vec{w}) \text{ where } \vec{w} \in \mathbb{R}^{d+1}\}$

Learner: $\mathcal{A}(D) = \hat{f}$

$$\hat{f} = \operatorname{argmin}_{f \in \mathcal{F}} \hat{L}(f)$$

$$= \operatorname{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} -\left[y_i \log(f(\vec{x}_i)) + (1-y_i)\log(1-f(\vec{x}_i))\right]$$

$$= \operatorname{argmin}_{f \in \mathcal{F}} -\sum_{i=1}^{n} \left[y_i \log(f(\vec{x}_i)) + (1-y_i)\log(1-f(\vec{x}_i))\right]$$

$$= f_{MLE}$$

Why not just learn $p(y=1|\vec{x})$ using linear regression?

Ex:

$d=1$