# ML & Opt Lecture 6

- Practical considerations:
  - Train/valid/test splits
  - Cross-validation
  - Hyperparameter selection
  - Regularization (overfitting)
- Revisiting binary/multiclass logistic regression, linear separability

# Logistic Regression

$x \in \mathbb{R}^d$ features use to predict $y \in \{-1, 1\}$

Assume that the log odds of $y$ being 1 is linear in $x$ $(p = \mathbb{P}(Y=1|x))$

$$\ln\left(\frac{p}{1-p}\right) = \omega_0 + \sum_{i=1}^{d} \omega_i x_i = \omega_0 + \omega^T x$$

$$\equiv \tilde{\omega}^T \tilde{x}$$ where

$\uparrow$ equivalent

$$\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$$

and $$\tilde{\omega} = \begin{bmatrix} \omega \\ \omega_0 \end{bmatrix}$$

Interpretation: probability of $Y=1$

goes up if $\omega_i > 0$ and $x_i$ increases;

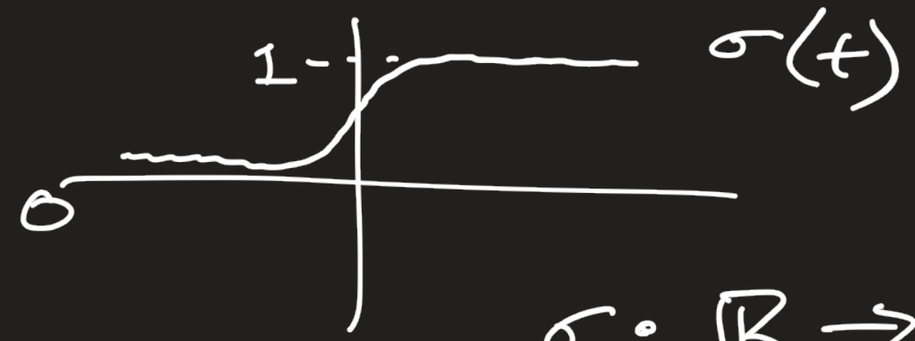
goes down if $\omega_i < 0$ and $x_i$ increases;

and $\omega_0$ determines the baseline prob of $Y=1$

$$\Rightarrow \frac{p}{1-p} = \exp(\omega_0 + \omega^T x) \Rightarrow p = \frac{1}{1 + e^{-\omega_0 + \omega^T x}}$$

$$\Rightarrow p = \sigma(\underbrace{\omega_0 + \omega^T x}) \quad \text{where} \quad \sigma(t) = \frac{1}{1 + e^{-t}}$$

called the logit




$\sigma : \mathbb{R} \to [0, 1]$

This is the same as saying

$$y \sim \text{Bern}(\sigma(\omega_0 + \omega^T x))$$

Goal: recover $\omega_0$ & $\omega$ given training data so in the future when we observe $x$ we can predict $y$

Notice we predict $y$ by choosing $c \in \{-1, 1\}$ that maximizes

$$p(y=c|x) = \begin{cases} \sigma(\tilde{w}^T \tilde{x}) & c=1 \\ 1-\sigma(\tilde{w}^T \tilde{x}) & c=-1 \end{cases}$$

notice

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

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

so

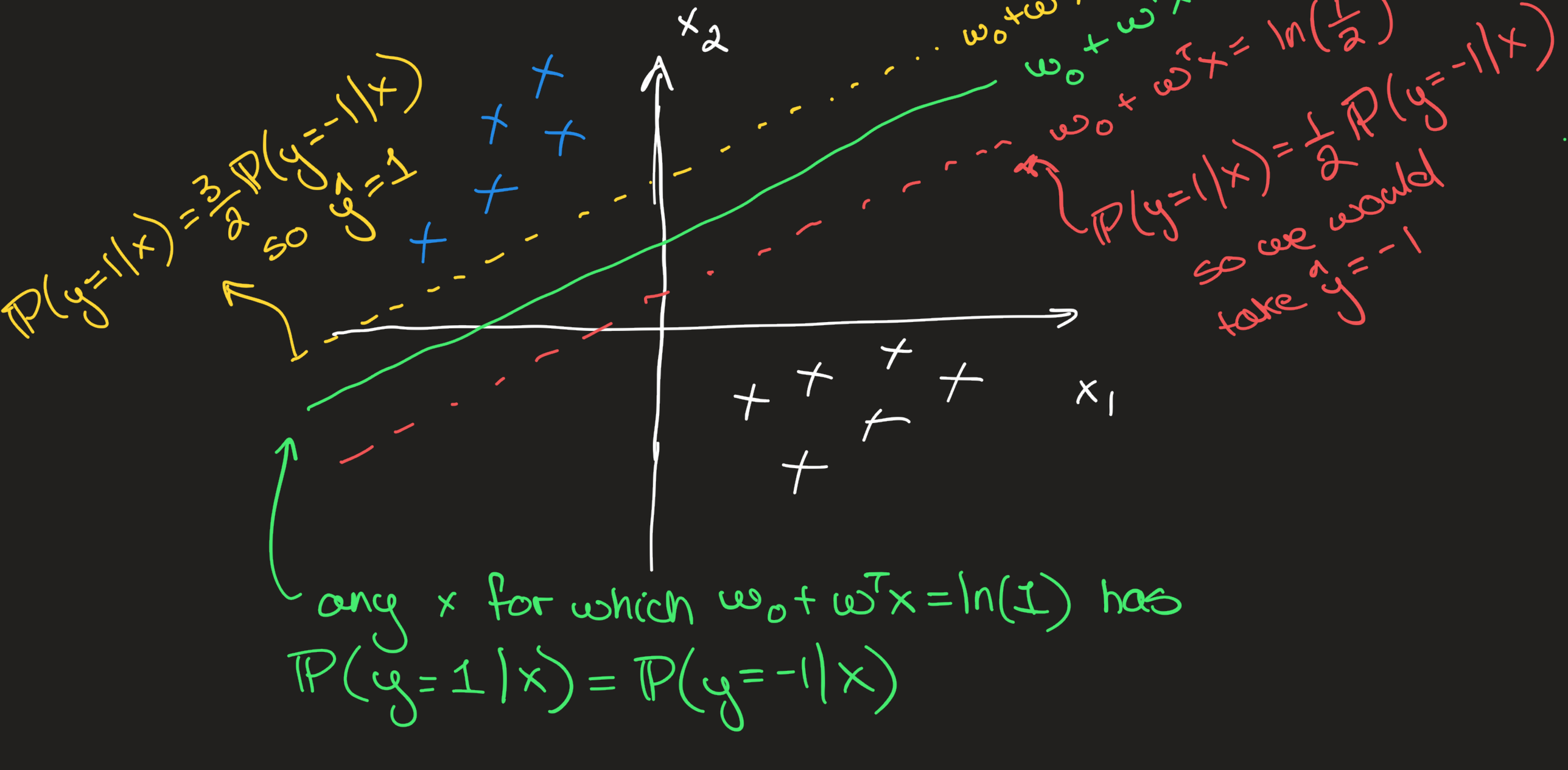
$$\boxed{1-\sigma(t) = \sigma(-t)}$$

This implies

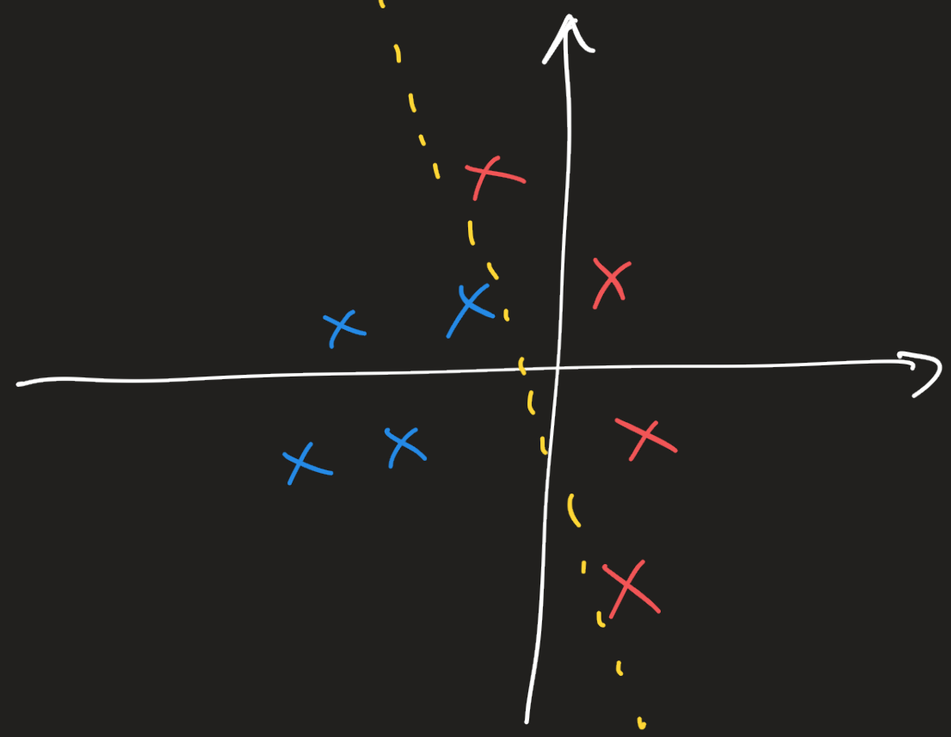
$$p(y=c|x) = \begin{cases} \sigma(\tilde{w}^T \tilde{x}) & c=1 \\ \sigma(-\tilde{w}^T \tilde{x}) & c=-1 \end{cases} = \sigma(c\tilde{w}^T \tilde{x})$$

Now we see that we can predict $y$, given $x$, with

$$\hat{y} = \underset{c \in \{-1,1\}}{\operatorname{argmax}} \; p(c|x) = \underset{c \in \{-1,1\}}{\operatorname{argmax}} \; \sigma(c\tilde{w}^T\tilde{x})$$



Question: given training data, how to estimate $w_0, w$



Use MLE to find best separating hyperplane

$$\mathcal{L}(w_0, w) = \frac{1}{n} \sum_{i=1}^{n} -\log p(y_i | x_i ; w_0, w)$$

↑ NLL of the model parameters $w_0, w$ given the observed (training) data

Notice

$$\mathcal{L}(w_0, w) = \frac{1}{n} \sum_{i=1}^{n} -\log \sigma(y_i \tilde{w}^T \tilde{x}_i)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \log \left( \frac{1}{\sigma(y_i \tilde{w}^T \tilde{x}_i)} \right)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \log \left( 1 + e^{-y_i \tilde{w}^T \tilde{x}_i} \right)$$

take $\ell(v, y) = \log \left( 1 + e^{-vy} \right)$ called the logistic loss

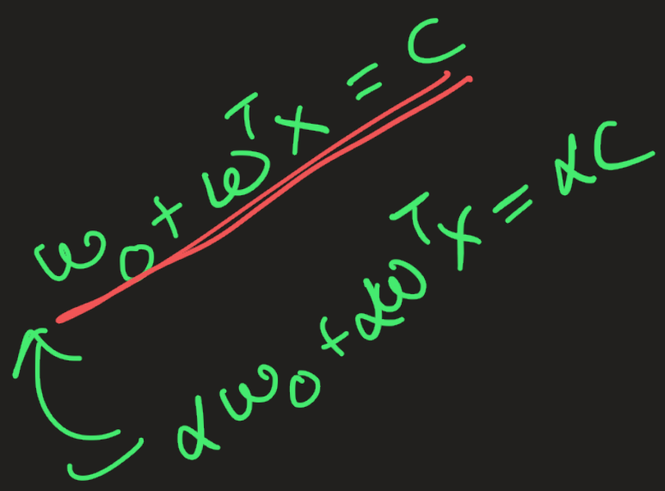
$$= \frac{1}{n} \sum_{i=1}^{n} \ell(\tilde{w}^T \tilde{x}_i, y_i)$$

So fitting the model parameters for logistic regression using MLE is done by solving

$$\hat{w_0}, \hat{w} = \underset{\substack{w_0 \in \mathbb{R} \\ w \in \mathbb{R}^d}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(w_0 + w^T x_i, y_i)$$

where $\ell(v, y) \equiv \log(1 + e^{-vy})$ is the logistic loss fn.

- notice logistic regression always has one solution (when it has a finite solution)




ASIDE: if the training data is actually linearly separable, and $w_0, w$ separate the data, then $\alpha w_0, \alpha w$ also separate the data, and I can choose $\alpha \to \pm\infty$ so that $\mathcal{L}(\alpha w_0, \alpha w) \to 0$

# Multiclass Logistic Regression

Given features $x \in \mathbb{R}^d$, predict class $y \in [k] = \{1, \dots, k\}$

Ex: given pixel image $x \in [256]^{m \times n}$ determine the type of image

Model: assume the log-odds (logits or scores) of each class, given $x$, relative to class $k$ are linear:

$$\log \frac{\mathbb{P}(Y=1 \mid X=x)}{\mathbb{P}(Y=k \mid X=x)} = b_1 + w_1^T x$$

$$\vdots$$

$$\log \frac{\mathbb{P}(Y=k-1 \mid X=x)}{\mathbb{P}(Y=k \mid X=x)} = b_{k-1} + w_{k-1}^T x$$

Parameters

$b \in \mathbb{R}^{k-1}$

$w \in \mathbb{R}^{k-1 \times d}$ with rows $w_1^T, \dots, w_{k-1}^T$

These $k-1$ equations plus the fact that $\sum_{i=1}^{k} \mathbb{P}(Y=i \mid x) = 1$ gives us

$$\mathbb{P}(Y=i \mid X=x) = \frac{\exp(b_i + w_i^T x)}{1 + \sum_{l=1}^{k-1} \exp(b_l + w_l^T x)} \qquad i=1,\dots,k-1$$

$$\mathbb{P}(Y=k \mid X=x) = \frac{1}{1 + \sum_{l=1}^{k-1} \exp(b_l + w_l^T x)}$$

For ease of use we introduce nonzero logits for class $K$, i.e. note that

$$\mathbb{P}(Y=k|X) = \frac{1}{1+\sum_{i=1}^{k-1}\exp(b_i+\omega_i^T x)} \cdot \frac{\exp(b_k+\omega_k^T x)}{\exp(b_k+\omega_k^T x)}$$

$$= \frac{\exp(b_k+\omega_k^T x)}{\exp(b_k+\omega_k^T x)+\sum_{i=1}^{k-1}\exp((b_i+b_k)+(\omega_i+\omega_k)^T x)}$$

similarly we have

$$\mathbb{P}(Y=i|X) = \frac{\exp((b_i+b_k)+(\omega_k+\omega_i)^T x)}{\exp(b_k+\omega_k^T x)+\sum_{i=1}^{k-1}\exp((b_i+b_k)+(\omega_i+\omega_k)^T x)}$$

We relabel the shifted biases and weights

$$b_K \leftarrow b_K$$
$$w_K \leftarrow w_K$$
$$\left.\begin{aligned} b_i &\leftarrow b_i + b_K \\ w_i &\leftarrow w_i + w_K \end{aligned}\right\} \text{ for } i \neq K$$

Then we have

$$\mathbb{P}(y = i \mid x) = \frac{\exp(b_i + w_i^T x)}{\sum_{l=1}^{K} \exp(b_l + w_l^T x)} \quad \text{for } i = 1, \ldots, K$$

($b_i + w_i^T x$: logit for class $i$; $\sum_{l=1}^{K} \exp(b_l + w_l^T x)$: sum exp of the logits)

$$= \frac{\left(e^{b + w^T x}\right)_i}{1^T e^{b + w^T x}}$$

where $\text{softmax}(v) = \frac{e^v}{1^T e^v}$

$$= \text{softmax}\left(b + w^T x\right)_i$$

Recall $\sigma : \mathbb{R} \to [0,1]$

logits → probability $\gamma = I$

Now for multiclass logistic regression

$$\text{softmax} : \mathbb{R}^k \longrightarrow \text{set of probability vectors over the } k \text{ classes}$$

logit vector for $k$ classes

softmax is a continuous approximation of the function that maps a vector $v$ to a vector of zeros and ones where there is a one corresponding to the largest entry in $v$, and a zero everywhere else

Ex:

$$\begin{bmatrix} 30 \\ 1 \\ 0 \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} 1 \\ 2.54 \times 10^{-13} \\ 9.36 \times 10^{-14} \end{bmatrix}$$

logits → class probs

Ex:

$$\begin{bmatrix} 0.9 \\ 0.05 \\ 0.05 \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} 0.539 \\ 0.230 \\ 0.230 \end{bmatrix}$$

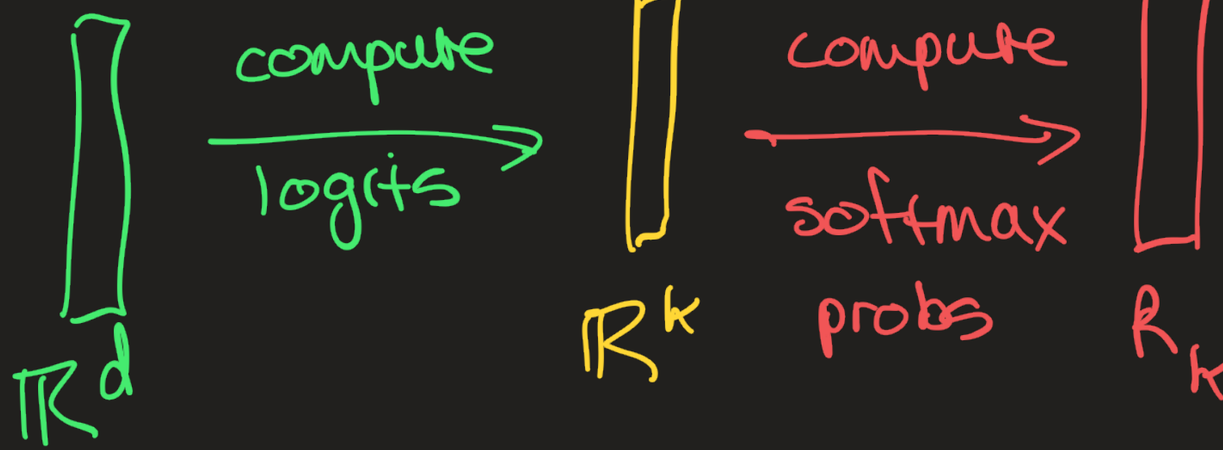
logits → class probs

Given that my data follows the label dist

$$\mathbb{P}(y=i \mid x) = \text{softmax}(Wx+b)_i$$

how do I predict the class label given $x$?

Take $x$ $\quad Wx+b$




$$\mathbb{P}(y=i \mid x) = \text{softmax}(Wx+b)_i$$

and take $x$ as having class

$$\hat{y} = \underset{i \in [K]}{\text{argmax}} \; \mathbb{P}(y=i \mid x)$$

$$= \underset{i \in [K]}{\text{argmax}} \; (Wx+b)_i$$

We determine $w$ & $b$ given training data by minimizing the NLL (i.e. MLE):

$$\mathcal{L}(w, b) := \frac{1}{n} \sum_{i=1}^{n} -\log \mathbb{P}[Y = y_i \mid X = x_i ; w, b]$$

and take

$$\hat{w}, \hat{b} = \underset{w, b}{\operatorname{argmin}} \mathcal{L}(w, b)$$

To write the NLL more conveniently, encode the class observations as <u>one-hot vectors</u>. Instead of $y_i \in [k]$, consider $y_i \in \{e_1, \dots, e_k\} \subseteq \mathbb{R}^k$.

E.g. if training example $i$ is of class 2, we take

$$y_i = e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^k$$

Now we see that if the $i$th training example has class $j$, then $y_i = e_j$ and

$$\log \mathbb{P}[y = y_i \mid X = x_i] = \log \mathbb{P}(y = e_j \mid X = x_i)$$

$$= \log \operatorname{softmax}(W x_i + b)_j$$

$$= \log e_j^T \operatorname{softmax}(W x_i + b)$$

$$= y_i^T \log \operatorname{softmax}(W x_i + b)$$

$$= y_i^T \log \left( \operatorname{softmax}(W x_i + b) \right)$$

$$= y_i^T \log \left( \frac{\exp(W x_i + b)}{1^T \exp(W x_i + b)} \right)$$

$$\Rightarrow \log \mathbb{P}[Y=y_i \mid X=x_i] = y_i^T (W x_i + b) - \log\left(\sum_{l=1}^{k} \exp(W x_i + b)_l\right)$$

$$\text{logsumexp}(W x_i + b)$$

where $\text{logsumexp} : v \mapsto \log(1^T e^v)$

$$= y_i^T (W x_i + b) - \text{logsumexp}(W x_i + b)$$

Therefore we get the MCE estimates for the model parameters by solving

$$\hat{w}, \hat{b} = \underset{w,b}{\operatorname{argmin}} -\frac{1}{n}\sum_{i=1}^{n}\left[ y_i^T (wx_i + b) - \log\left(\sum_{l=1}^{K} \exp(wx_i + b)_l\right)\right]$$

question: what is the loss function so that

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, wx_i + b) = \mathcal{L}(w, b) \; ?$$