Lecture 8: Logistic Regression

02 February 2024 11:01

$$\begin{bmatrix} P(y_i=1 \mid x_i, w) \\ P(y_i=0 \mid x_i, w) \end{bmatrix} \quad \text{for class 1, } w_1^T x_i \quad \text{for class 2, } w_2^T x_i$$

$$P(y_i \mid x_i, w) = \frac{e^{w_1^T x_i}}{e^{w_1^T x_i} + e^{w_2^T w_i}} \qquad \begin{bmatrix} \tau\, e^{w_1^T x_i} \\ \tau\, e^{w_2^T x_i} \end{bmatrix} \quad \text{where } \tau = \frac{1}{e^{w_1^T x_i} + e^{w_2^T x_i}}$$

$$\Rightarrow f(w, x_i) = \frac{1}{1 + e^{-w^T x_i}} \quad \text{where } w = w_1 - w_2$$

$$= \sigma(w^T x_i) \quad \left(\text{sigmoid fn} : \sigma(z) = \frac{1}{1-e^{-z}}\right)$$

Binary Logistic Regression

We've $w_{LR}^* \in \mathbb{R}^d$ such that $w_{LR}^* = \arg\max_w \prod_{i=1}^n P(y_i \mid x_i, w)$

$$= \arg\max_w \sum_{i=1}^n \log P(y_i \mid x_i, w)$$

$$= \arg\min_w \left(-\sum_{i=1}^n \log P(y_i \mid x_i, w)\right)$$

We've Negative log likelihood,

$$NLL_i = -\log P(y_i \mid x_i, w)$$

$$= -\log P(y_i=1 \mid x_i, w), \; y_i = 1 \quad \text{or} \quad -\log\left(1 - P(y_i=1 \mid x_i, w)\right), \; y_i = 0$$

$$= -y_i \log\left(\sigma(w^T x_i)\right) - (1-y_i) \log\left(1 - \sigma(w^T x_i)\right)$$

(CROSS ENTROPY LOSS)

$$H(p, q) = -\sum_{x \in X} P(x) \log q(x)$$

$$\therefore NLL_i(w) = y_i \log\left(1 + e^{-w^T x_i}\right) - (1-y_i) \log\left(\frac{e^{-w^T x_i}}{1 + e^{-w^T x_i}}\right)$$

$$= y_i \log\left(1 + e^{-w^T x_i}\right) + (1-y_i)\, w^T x_i + (1-y_i) \log\left(1 + e^{-w^T x_i}\right)$$

$$= \log\left(1 + e^{-w^T x_i}\right) + (1-y_i)\, w^T x_i$$

We are going to apply gradient descent on NLL_i to minimize it.

$$\nabla NLL_i(w) = \frac{e^{-w^T x_i}}{}$$

We are going to apply gradient descent on NLL_i to minimize it.

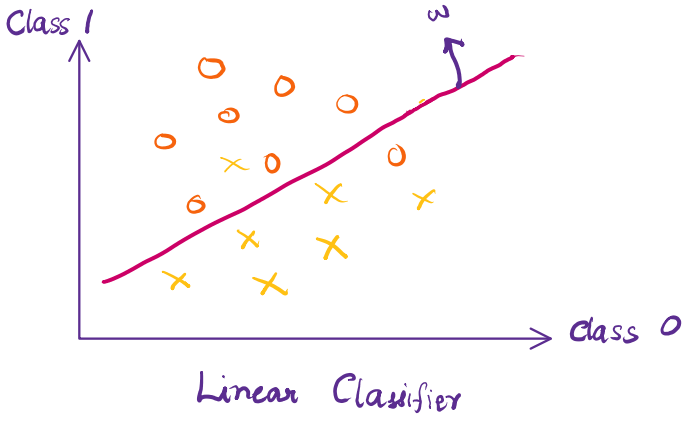
$$\therefore \nabla_w NLL_i(w) = \frac{-e^{-w^T x_i}}{1+e^{-w^T x_i}} x_i + (1-y_i) x_i$$

$$= \frac{1}{1+e^{-w^T x_i}} x_i - x_i y_i \in \mathbb{R}^d$$

$$= -(y_i - \sigma(w^T x_i)) x_i$$

Applying gradient descent: $w_{t+1} \leftarrow w_t - \eta \sum_{i=1}^{n} \nabla_w NLL_i(w)$

Recall that $\frac{P(y=1|x,w)}{P(y=0|x,w)} > 0 \rightarrow \hat{y}=1$ or, $\hat{y}=0$ $\Rightarrow w^T x > 0 \rightarrow \hat{y}=1$ or, $\hat{y}=0$



Linear Classifier

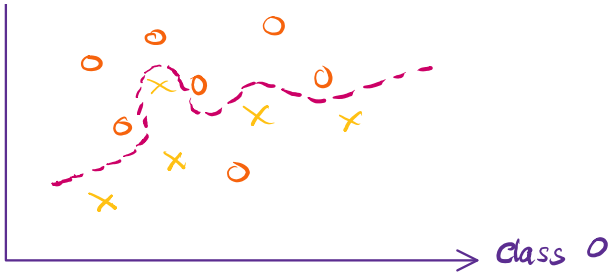
To accomodate a better classification, (give place to outliers) we can use a basis function,

Basis function: $\Phi(x)$

Logistic function : $\sigma(w^T \Phi)$

Now we check, $w^T \phi > 0$ instead of $w^T x > 0$

This may be linear in a higher dimensions, but non linear in lower



Classification using basis

We can accomodate for overfitting by using a regularizer:

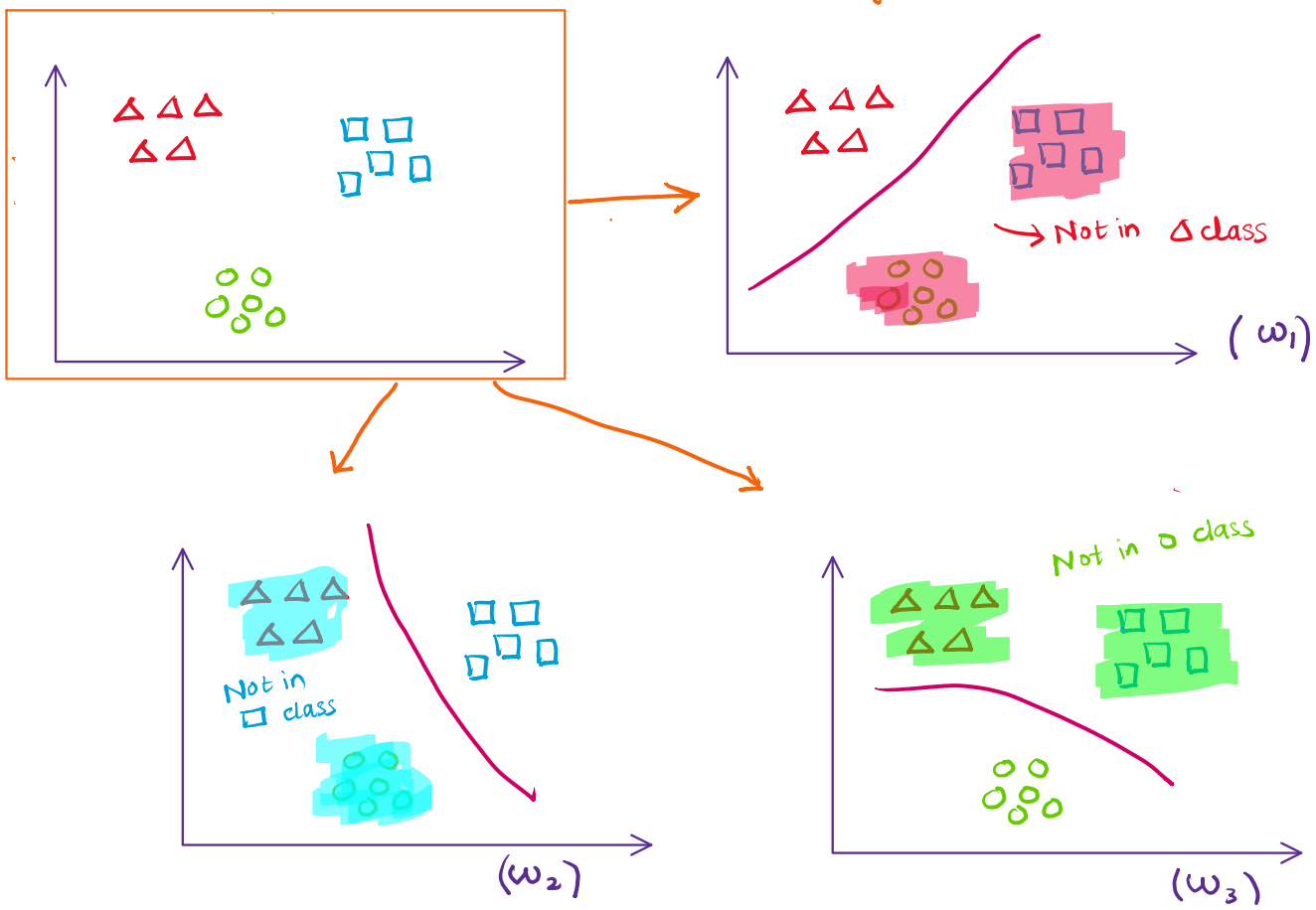
$$NLL_{reg}(w) = NLL_{LR}(w) + \lambda \|w\|^2$$

(Minimize using gradient descent)

Multi-class Classification

① One-v/s-rest Classifier

We can use any binary classifier



Let $\hat{x}$ be test point

$\sigma(\omega_1^T \hat{x}) \rightarrow$ class 1

$\sigma(\omega_2^T \hat{x}) \rightarrow$ class 2

$\sigma(w_1^T \hat{x}) \rightarrow$ class 1

$\sigma(w_2^T \hat{x}) \rightarrow$ class 2

$\sigma(w_3^T \hat{x}) \rightarrow$ Class 3

$\therefore$ Final prediction, $\hat{y} = \underset{k}{\text{argmax}} \ \sigma(w_k^T \hat{x})$

Softmax Regression:

$w_i \rightarrow P(y=i \mid x, w)$ (Propability that y belong to i^{th} class)

Like logistic regression we've

$$P(y=i \mid x, w) = \tau \, e^{w_i^T x} \quad : \text{where } \tau = \frac{1}{\sum_{j=1}^{k} e^{w_j^T x}} \quad \text{(Assume k classes)}$$

$$= \frac{e^{w_i^T x}}{\sum_{j=1}^{k} e^{w_j^T x}}$$

Softmax fn

$$f(x, w) = \begin{bmatrix} P(y=1 \mid x, w) \\ P(y=2 \mid x, w) \\ \vdots \\ P(y=k \mid x, w) \end{bmatrix} = \begin{bmatrix} \tau \, e^{w_1^T x} \\ \tau \, e^{w_2^T x} \\ \vdots \\ \tau \, e^{w_k^T x} \end{bmatrix} \quad \text{where, } w = \begin{bmatrix} - \, w_1^T \, - \\ - \, w_2^T \, - \\ \vdots \\ - \, w_k^T \, - \end{bmatrix}_{k \times d}$$

Here, $NLL(w) = -\sum_{i=1}^{n} P(y_i \mid x, W) = -\sum_{i=1}^{n} \sum_{k=1}^{k} \mathbb{I}\{y_i = k\} \log \frac{e^{w_k^T x}}{\sum_{j=1}^{k} e^{w_j^T x}}$ (Indicator fn.)

$$NLL_i(w) = -\sum_{k=1}^{K} \left[\mathbb{I}\{y_i = k\} \left[w_k^T x - \log \left(\sum_{j=1}^{K} e^{w_j^T x} \right) \right] \right]$$

Again, we will perform gradient descent:

$$-\nabla_{w_k} NLL_i(w) = \begin{cases} x - \dfrac{e^{w_k^T x}}{\sum_j e^{w_j^T x}}, & y_i = k \\ - \dfrac{e^{w_k^T x}}{\sum_j e^{w_j^T x}}, & y_i \neq k \end{cases}$$

$$\therefore \quad \nabla_{w_k} NLL_i(w) = -\left[\mathbb{I}\{y_i = k\} - f_k(x, w)\right] x$$

> One-Hot Representation:
> Instead of indicator function,
> we can represent y_i as a vector with k entries such that if $y_i \in$ class α, everywhere in the vector it is zero except α^{th} position
>
> $$y_i = [0 \; 0 \cdots \underset{\alpha}{1} \cdots 0]$$

Gradient Descent Step:

$$\begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_k^T \end{bmatrix}_{t+1} \leftarrow \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_k^T \end{bmatrix}_{t} - \eta \begin{bmatrix} -\nabla_{w_1} NLL(w) - \\ -\nabla_{w_2} NLL(w) - \\ \vdots \\ -\nabla_{w_k} NLL(w) - \end{bmatrix}$$

where,

$$\nabla_{w_k} NLL(w) = \sum_{i=1}^{n} \nabla_{w_k} NLL_i(w)$$

NB v/s LR

Naive Bayes : $\arg\max_y P(Y=y) \prod_{i=1}^{d} P(x_i | Y=y)$

- $P(Y=y)$: inter class distribution
- $P(x_i | Y=y)$: infer data distribution

Models like NB are called generative model which uses data distribution to infer class distribution

Logistic Regression : $\arg\max_y P(y | x, w)$ → given by $\sigma(w^T x)$

Models like LR are called discriminative models.

Gaussian Naive Bayes: Special case of LR

$$\prod_{i=1}^{d} (\ldots)$$

Gaussian Naive Bayes: Special case of LR

$$\underset{y_k}{\operatorname{argmax}}\ P(Y=y_k)\prod_{i=1}^{d} P(x_i \mid Y=y_k)$$

where, $P(x_i \mid y_k) \sim \mathcal{N}(\mu_{ik}, \sigma_{ik}^2)$

with following assumptions:

1. x_i, x_j are conditionally independent
2. $P(y=1)=\pi$, $P(y=0)=1-\pi$
3. $P(x_i \mid y=0) \sim \mathcal{N}(\mu_{i0}, \sigma_i^2)$
   $P(x_i \mid y=1) \sim \mathcal{N}(\mu_{i1}, \sigma_i^2)$

$\therefore$ We've $P(y_i=1 \mid x) = \dfrac{P(x \mid y_i=1)\,P(y_i=1)}{P(x \mid y_i=1)\,P(y_i=1) + P(x \mid y_i=0)\,P(y_i=0)}$

$$= \frac{1}{1+\exp\{w_0+\sum w_j x_j\}}$$

(What are w_0, w_j's? (HW))