03/04/15

LOGISTIC REGRESSION

(1) Recall: $\hat{y}_{BayesOPT} = \underset{y}{\operatorname{argmax}}\ P(Y=y \mid \vec{X}=\vec{x})$

Generative Approach

estimate $P(\vec{X}|Y), P(Y)$ $\Longrightarrow$ Bayes Rule $\Rightarrow P(Y|\vec{X}) \Rightarrow$ predict

Discriminative Approach

estimate

$P(Y|\vec{X}) \Rightarrow$ predict

↑ How?

(2) LR for Binary classification ($y \in \{0,1\}$)

We know $Y|\vec{X}=\vec{x} \sim Ber(\theta_{\vec{x}})$

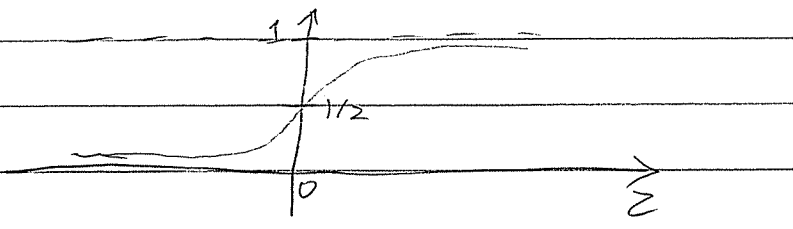
$\theta_{\vec{x}} \in [0\ 1]$

Silly idea: Can we regress from $\vec{x} \rightarrow \theta_{\vec{x}}$?

Problem: $\theta_{\vec{x}} \in [0\ 1]$

Solution: Meet the Sigmoid / Logistic Function

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



LR as a Probabilistic Model

$$P(Y=1 \mid \vec{X}=\vec{x}) = \frac{1}{1+e^{-\vec{w}^T\vec{x}}} = \sigma(\vec{w}^T\vec{x})$$

$$\iff Y \mid \vec{X}=\vec{x} \sim \text{Ber}(\sigma(w^Tx))$$

Compare with probabilistic interpretation of linear regression (OLS)

$$Y \mid \vec{X}=\vec{x} \sim N(\vec{w}^T\vec{x}, \sigma^2)$$

(3) LR is a linear classifier!

Note: $P(Y=1 \mid \vec{X}=\vec{x}) = \frac{1}{1+e^{-w^Tx}}$

$$\Rightarrow P(Y=0 \mid \vec{X}=\vec{x}) = 1 - \frac{1}{1+e^{-w^Tx}} = \frac{e^{-w^Tx}}{1+e^{-w^Tx}} = \frac{1}{1+e^{+w^Tx}}$$

So how do we predict $\hat{y}$?

$$\hat{y}_{MAP} = \underset{y \in \{0,1\}}{\text{argmax}}\ P(Y=y \mid \vec{X}=\vec{x})$$

equivalent to checking

$$\frac{P(Y=1 \mid \vec{X}=\vec{x})}{P(Y=0 \mid \vec{X}=\vec{x})} \overset{?}{\geq} 1$$

if yes $\hat{y}=1$, else $\hat{y}=0$

$$\Rightarrow \log\left[\frac{P(Y=1 \mid \vec{X}=\vec{x})}{P(Y=0 \mid \vec{X}=\vec{x})}\right] \overset{?}{\geq} 0$$

$$\Rightarrow \log\left[\frac{1/(1+e^{-w^Tx})}{e^{-w^Tx}/(1+e^{-w^Tx})}\right] \gtrless^{?} 0$$

$$\Rightarrow \log\left(e^{w^Tx}\right) \gtrless^{?} 0$$

$$\Rightarrow \boxed{w^Tx \gtrless^{?} 0}$$ linear classifier

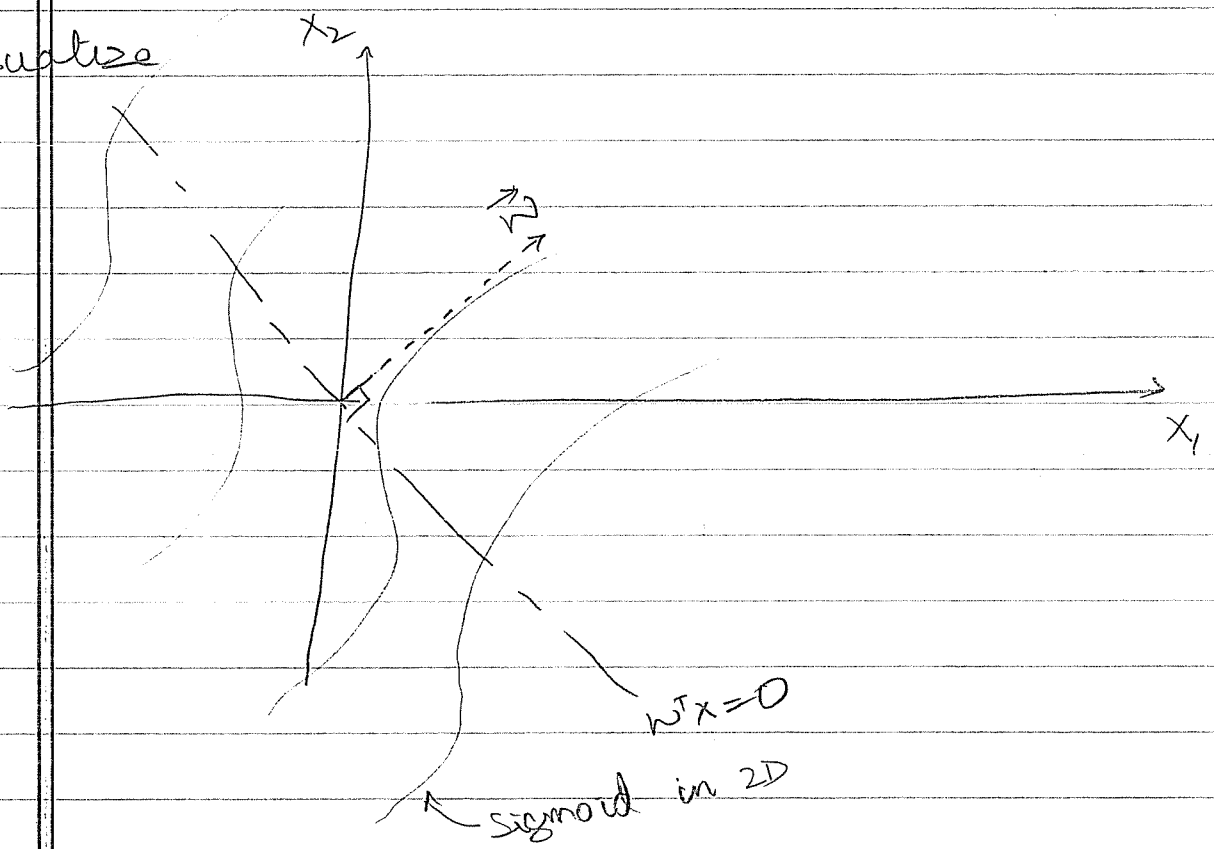
If $w^Tx \geq 0 \Rightarrow \hat{y} = 1$

$w^Tx < 0 \Rightarrow \hat{y} = 0$

So $w^Tx \equiv$ Score for class 1

$\frac{1}{1+e^{-w^Tx}} \equiv$ Prob of class 1.

Visualize



④ Estimating LR parameter: How do we learn $\vec{w}$?

[Pretty much the same way we learn coin-toss / Bernoulli parameters]

Compare & Contrast

"Coin-Toss"	Logistic Regression
$y \in \{0,1\}$	$y \in \{0,1\}$ $\vec{x} \in \mathbb{R}^d$
Given Dataset	
$D = \{y_1, \dots, y_N\}$	$D = \{(\vec{x}_1, y_1) \dots (\vec{x}_n, y_N)\}$
Model: $Y \sim Ber(\theta)$	Model: $Y \mid \vec{X}=\vec{x} \sim Ber(\underbrace{\sigma(w^T x)}_{\theta_x})$
$P(Y=1) = \theta$	$P(Y=1 \mid \vec{X}=\vec{x}) = \theta_x$
$P(Y=0) = 1-\theta$	$P(Y=0 \mid \vec{X}=\vec{x}) = 1-\theta_x$
Likelihood of 1 sample	
$L(\theta) = \theta^y (1-\theta)^{1-y}$	$L(w) = \theta_x^y (1-\theta_x)^{1-y}$
Likelihood of Dataset	
$L(\theta) = \prod_{i=1}^{N} \theta^{y_i} [1-\theta]^{(1-y_i)}$	$L(w) = \prod_{i=1}^{N} \theta_{x_i}^{y_i} [1-\theta_{x_i}]^{(1-y_i)}$
Notice this is same as	
$= \theta^{\alpha_H} [1-\theta]^{\alpha_T}$	$= \prod_{i=1}^{N} [\sigma(w^T x_i)]^{y_i} [1-\sigma(w^T x_i)]^{(1-y_i)}$
where	
$\alpha_H = \sum_{i=1}^{N} y_i$	
$\alpha_T = N - \alpha_H$	

So for LR, the log-likelihood is

$$LL(\vec{w}) = \sum_{i=1}^{N} \left[y_i \log \sigma(w^T x_i) + (1-y_i) \log(1-\sigma(w^T x_i)) \right]$$

$$= \sum_{i=1}^{N} \left[y_i \log\left(\frac{e^{w^T x_i}}{1+e^{w^T x_i}}\right) + (1-y_i) \log\left(\frac{1}{1+e^{w^T x_i}}\right) \right]$$

$$= \sum_{i=1}^{N} \left[y_i \log(e^{w^T x_i}) + \log\left(\frac{1}{1+e^{w^T x_i}}\right) \right]$$

$$= \sum_{i=1}^{N} \left[\underbrace{y_i w^T x_i}_{\text{linear in } \vec{w}} - \underbrace{\log(1+e^{w^T x_i})}_{\text{Convex in } \vec{w}} \right]$$

[Why? Exercise!]

Good News: $LL(\vec{w})$ is concave in $\vec{w}$
so we can maximize it efficiently!

Bad News: $\frac{\partial LL(\vec{w})}{\partial \vec{w}} = 0$ won't have a closed form solution. Let's try.

$$\frac{\partial LL(\vec{w})}{\partial \vec{w}} = \sum_{i=1}^{N} \left[y_i \vec{x}_i^T - \frac{1}{1+e^{w^T x_i}} \cdot e^{w^T x_i} \cdot x_i^T \right]$$

$$= \sum_{i=1}^{N} \left[y_i - \frac{1}{1+e^{-w^T x_i}} \right] \cdot x_i^T$$

$$\Rightarrow \quad \frac{\partial LL(\vec{w})}{\partial \vec{w}} = \sum_{i=1}^{N} \left[y_i - P(Y=1 \mid \vec{X}=\vec{x}_i, \vec{w}) \right] \cdot \vec{x}_i^T$$

Not a linear system of equations in $\vec{w}$
can't solve directly.

→ Solution: Gradient Descent!

Goal: $\min_{\vec{w}} f(\vec{w})$

Algorithm: Initialize $\vec{w}^{(0)} = \vec{0}$ (say)

$$\vec{w}^{(t+1)} = w^{(t)} - \eta \left. \frac{\partial f(\vec{w})}{\partial \vec{w}} \right|_{w^{(t)}}$$

Update Rule (in optimization)
Learning Rule (in ML)

η: Step-size (in optimization) or "learning Rate" (in ML)

$\left. \frac{\partial f(\vec{w})}{\partial \vec{w}} \right|_{w^{(t)}}$: Gradient evaluated at time(t)

→ Gradient Ascent for LR

$$\vec{w}^{(t+1)} = w^{(t)} + \eta \left. \frac{\partial LL}{\partial \vec{w}} \right|_{\vec{w}^{(t)}}$$

$$= w^{(t)} + \eta \sum_{i=1}^{N} \left[\underbrace{y_i}_{\text{truth } \{0,1\}} - \underbrace{P(Y=1 \mid \vec{X}=\vec{x}_i, \vec{w}^{(t)})}_{\text{what our model believes } [0\ 1]} \right] \cdot \vec{x}_i$$

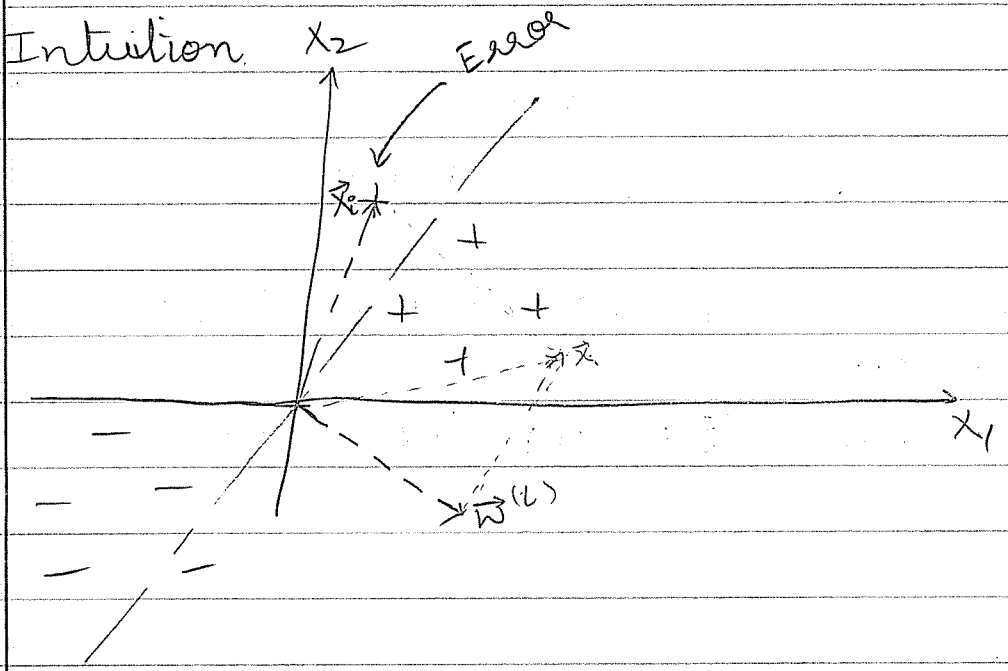
Error

Pattern/data

So gradient ascent says

"make $\vec{w}$ more like the $\vec{x}_i$ we are currently incorrect on"

Intuition



$w^T x \geq 0$

$w^T x = 0$

③ MAP Estimation of $\vec{w}$

Model $\begin{cases} Y \mid X, \vec{w} \sim \text{Ber}(\sigma(\vec{w}^T \vec{x})) \\ \vec{w} \sim N(0, t^2 I) \Leftrightarrow w_j \sim N(0, t^2) \quad \text{I.I.D} \end{cases}$

$$\hat{w}_{MAP} = \underset{\vec{w}}{\text{argmax}} \log P(\vec{w} \mid D)$$

$$= \underset{\vec{w}}{\text{argmax}} \log \left[\frac{P(D \mid \vec{w}) P(\vec{w})}{P(D)} \right]$$

$$= \underset{\vec{w}}{\text{argmax}} \log P(D \mid \vec{w}) + \log P(\vec{w})$$

$$= \underset{\vec{w}}{\text{argmax}} \underbrace{\sum_{i=1}^{N} \log P(Y_i = y_i \mid \vec{X}_i = \vec{x}_i, \vec{w})}_{LL} + \underbrace{\sum_{j=0} \log P(w_j)}_{\text{log-prior}}$$

$$\frac{\partial [\circ]}{\partial \vec{w}} = \sum_{i=1}^{N} \underbrace{\left[y_i - P(Y=1 \mid \vec{x}_i, \vec{w}) \right] \vec{x}_i}_{\text{focus on mistake}} - \underbrace{\lambda \vec{w}}_{\text{but try to keep norm of } \vec{w} \text{ small by moving towards } \vec{0}}$$

⑥. Logistic Regression & Gaussian Naive Bayes

$x \in \mathbb{R}^d \quad y \in \{0, 1\}$

Assume Naive Bayes Classifier with

$P(X_i \mid Y=k) = N(\mu_k, \sigma_i)$ ← mean for each class / feature; ↖ variance only w.r.t feature

$P(Y) = \text{Bernoulli}(\theta)$

and derive $P(Y \mid X) \Rightarrow$ logistic regression

$$P(Y=1 \mid \vec{X}=\vec{x}) = \frac{P(\vec{X}=\vec{x} \mid Y=1)\, P(Y=1)}{P(\vec{X}=\vec{x})}$$

$$= \frac{P(\vec{X}=x \mid Y=1)\, P(Y=1)}{P(\vec{x} \mid y=1)P(y=1) + P(\vec{x} \mid y=0)P(y=0)}$$

marginalization & chain rule →

divide by num →

$$= \frac{1}{1 + \frac{P(\vec{X}=\vec{x} \mid Y=0)\, P(Y=0)}{P(\vec{X}=\vec{x} \mid Y=1)\, P(Y=1)}}$$

$e^{\log z} = z$ →

$$= \frac{1}{1 + \exp\left[\log \frac{P(\vec{X}=\vec{x} \mid Y=0)}{P(\vec{X}=x \mid Y=1)} + \underbrace{\log \frac{P(Y=0)}{P(Y=1)}}\right]}$$

doesn't depend on $\vec{X}$, looks like a bias term

naive assumption → $= \frac{1}{1 + \exp\left[\sum_{j=1}^{d} \log \frac{P(X_j = x_j | Y=0)}{P(X_j = x_j | Y=1)} + \log \frac{P(Y=0)}{P(Y=1)}\right]}$

Let's look at this term a bit

$$\log \frac{P(X_j = x_j | Y=0)}{P(X_j = x_j | Y=1)} = \log \frac{\frac{1}{\sqrt{2\pi\sigma_j}} e^{\frac{-(x_j - \mu_{j0})^2}{2\sigma_j^2}}}{\frac{1}{\sqrt{2\pi\sigma_j}} e^{\frac{-(x_j - \mu_{j1})^2}{2\sigma_j^2}}}$$

$$= \frac{-(x_j - \mu_{j0})^2}{2\sigma_j^2} + \frac{(x_j - \mu_{j1})^2}{2\sigma_j^2}$$

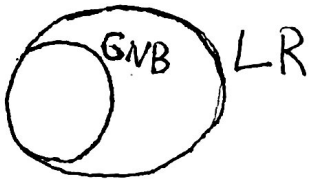
$$= \frac{2\mu_{j0}x_j - \mu_{j0}^2 - 2\mu_{j1}x_j + \mu_{j1}^2}{2\sigma_j^2}$$

$$= \underbrace{x_j (\mu_{j0} - \mu_{j1}) 4\sigma_j^2}_{x_j \cdot w_j} + \overbrace{\left(\frac{\mu_{j1}^2 - \mu_{j0}^2}{2\sigma_j^2}\right)}^{\text{another bias term } w_{0j}}$$

$$w_j = (\mu_{j0} - \mu_{j1}) 4\sigma_j^2$$

Going back

$$P(Y=1 | \vec{X} = \vec{x}) = \frac{1}{1 + \exp\left[\sum_{j=1}^{d} x_j \cdot w_j + w_0\right]} = LR!!!$$



GNB ⊂ LR