

03/04/15

①

LOGISTIC REGRESSION

① Recall: $\hat{y}_{\text{Bayes OPT}} = \underset{y}{\text{argmax}} P(Y=y | \vec{X}=\vec{x})$

Generative Approach

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

Discriminative Approach

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

How?

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

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

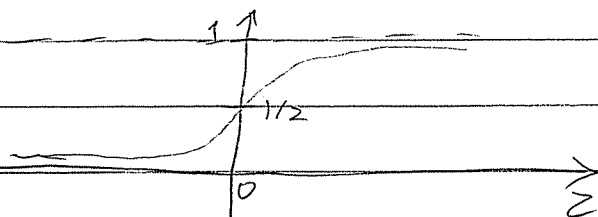
$$\theta_x \in [0, 1]$$

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

Problem: $\theta_x \in [0, 1]$

Solution: Meet the Sigmoid / Logistic Function

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



LR as a Probabilistic Model

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

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

Compare with probabilistic interpretation of linear regression (OLS)

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

③ LR is a linear classifier!

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

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

So how do we predict \hat{y} ?

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

equivalent
to checking

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

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

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

(2)

$$\Rightarrow \log \left[\frac{1/(1+e^{-w^T x})}{e^{-w^T x}/(1+e^{-w^T x})} \right] \geq 0$$

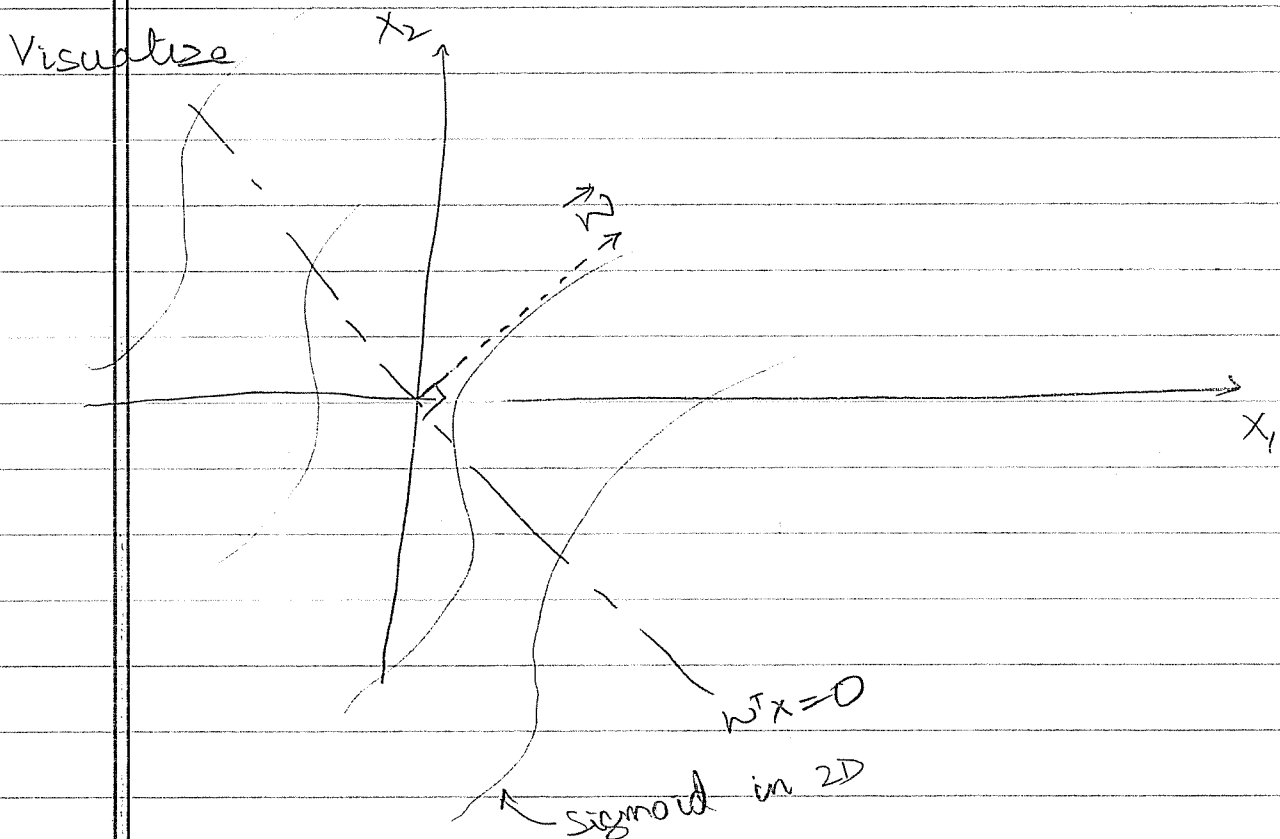
$$\Rightarrow \log(e^{w^T x}) \geq 0$$

$$\Rightarrow \boxed{w^T x \geq 0} \quad \text{linear classifier}$$

$$\begin{aligned} \text{If } w^T x \geq 0 &\Rightarrow y = 1 \\ w^T x < 0 &\Rightarrow y = 0 \end{aligned}$$

So $w^T x \equiv \text{Score for class 1}$

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



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

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

Compare to Contrast

"Coin-Toss"	Logistic Regression
$y \in \{0, 1\}$	$y \in \{0, 1\} \quad \vec{x} \in \mathbb{R}^d$
Given Dataset	
$D = \{y_1, \dots, y_N\}$	$D = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\}$
Model: $Y \sim \text{Ber}(\theta)$	Model: $Y X = \vec{x} \sim \text{Ber}(\underbrace{\sigma(\vec{w}^T \vec{x})}_{\theta_x})$
$P(Y=1) = \theta$ $P(Y=0) = 1 - \theta$	$P(Y=1 X = \vec{x}) = \theta_x$ $P(Y=0 X = \vec{x}) = 1 - \theta_x$
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	
$= \theta^{\alpha_H} [1 - \theta]^{\alpha_T}$	$= \prod_{i=1}^N [\sigma(\vec{w}^T \vec{x}_i)]^{y_i} [1 - \sigma(\vec{w}^T \vec{x}_i)]^{(1-y_i)}$
where	
$\alpha_H = \sum_{i=1}^N y_i$	
$\alpha_T = N - \alpha_H$	

So for LR, the log-likelihood is

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

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

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

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

linear
in \vec{w}

Convex in \vec{w} [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.

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

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

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

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)

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

Step-size (in optimization) or "Learning Rate" (in ML) η
 Gradient evaluated at time t
 Update Rule (in optimization)
 Learning Rule (in ML)

→ Gradient Ascent for LR

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

$$= \vec{w}^{(t)} + \eta \sum_{i=1}^N [y_i - P(Y=1 | \vec{X}=\vec{x}_i, \vec{w}^{(t)})] \cdot \vec{x}_i$$

(with $\{0, 1\}$)

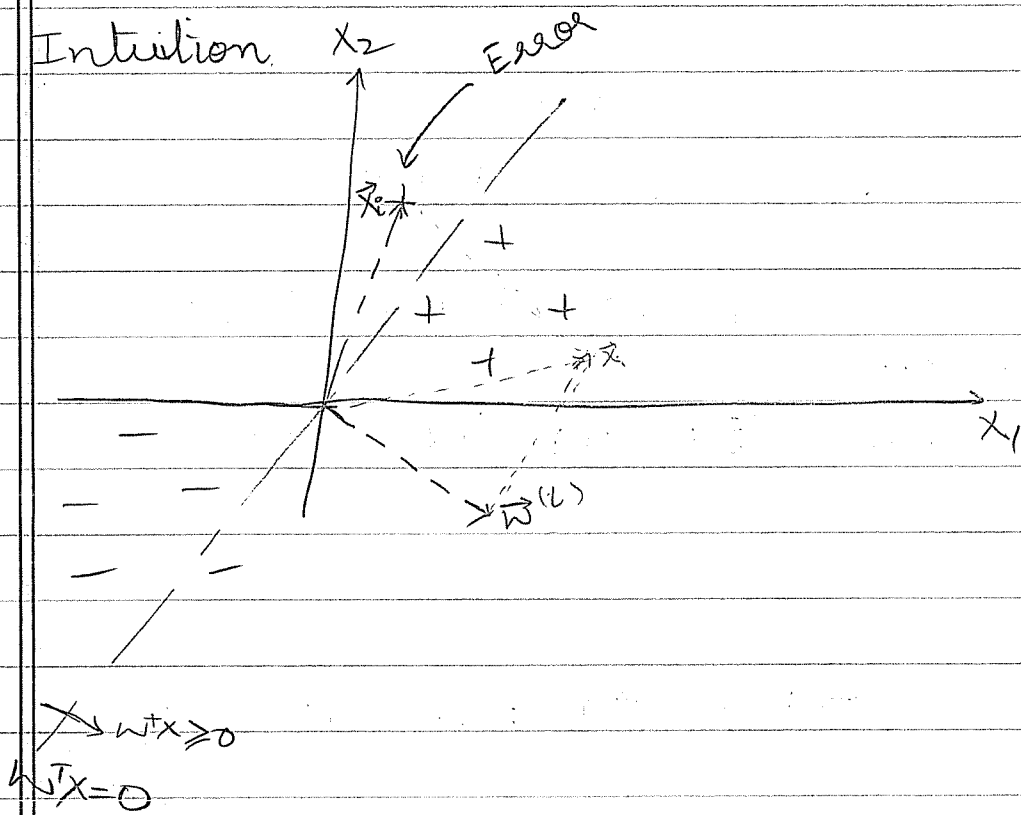
What our model believes $\{0, 1\}$

Error

Pattern/data

So gradient ascent says

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



③ MAP Estimation of \vec{w}

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

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

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

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

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

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