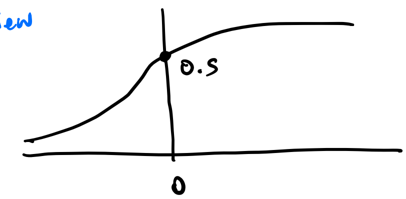
Perceptron

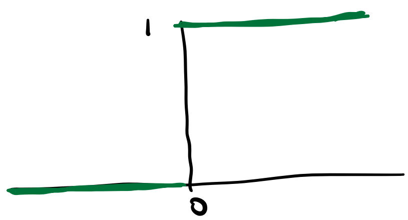
Log Reg review



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

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

Perceptron function



$$g(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Algorithm:

Repeatedly:

$$\theta_j := \theta_j + \alpha\underbrace{\left(y^{(i)} - h_\theta(x^{(i)})\right)}\, x_j^{(i)}$$

0: algo got it right

+1/-1: +1 if wrong, $y^{(i)} = 1$

-1 if wrong, $y^{(i)} = 0$

Based on this: update decision boundary.

Exponential families.

Exponential Prob. Density Fn (PDF): standard form

$$p(y;\eta) = b(y)\exp[\eta^T T(y) - a(\eta))$$

y: data

η: natural parameter

$T(y)$: sufficient statistic

$b(y)$: Base measure

$a(\eta)$: log-partition function

$$p(y;\eta) = \frac{b(y)\exp(\eta^T T(y))}{\exp(a(\eta))}$$

Bernoulli (for binary data)

ϕ: probability of event

$$p(y;\phi) = \phi^y (1-\phi)^{1-y}$$

$$= \exp(\log(\phi^y (1-\phi)^{1-y}))$$

$$= \exp(\underbrace{\log(\tfrac{\phi}{1-\phi})}_{\eta}\underbrace{y}_{T(y)} + \underbrace{\log(1-\phi)}_{a(\eta)})$$

$b(y) = 1$; $T(y) = y$

$\eta = \log(\frac{\phi}{1-\phi}) \Rightarrow \phi = \frac{1}{1+e^{-\eta}}$ (sigmoid fn)

$a(\eta) = -\log(1-\phi) = -\log(1 - \frac{1}{1+e^{-\eta}}) = \log(1+e^{\eta})$

Gaussian (w. fixed variance)

Assume $\sigma^2 = 1$

$$p(y;\mu) = \frac{1}{\sqrt{2\pi}}\exp\left(\frac{-(y-\mu)^2}{2}\right)$$

$$= \underbrace{\frac{1}{\sqrt{2\pi}} e^{-y^2/2}}_{b(y)} \exp(\mu y - \tfrac{1}{2}\mu^2)$$

μ: η $\quad$ y: $T(y)$ $\quad$ $\tfrac{1}{2}\mu^2$: $a(\eta)$

$b(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}$

$T(y) = y$ $\qquad a(\eta) = \frac{\mu^2}{2} = \frac{\eta^2}{2}$

$\eta = \mu$

Properties

1. MLE w.r.t. η is concave
   negative log likelihood (NLL) is convex
2. $E[y;\eta] = \frac{\partial}{\partial \eta} a(\eta)$
3. $Var[y;\eta] = \frac{\partial^2}{\partial \eta^2} a(\eta)$

Generalized Linear Models

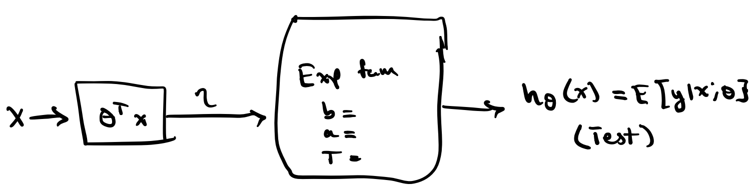
Assumptions / Design choices

1) $y \mid x;\theta \sim$ Exponential family

   Real – Gaussian
   Binary – Bernoulli
   Count – Poisson
   $\mathbb{R}^+$ – Gamma, Exponential
   Dist^n – Beta, Dirichlet

2) $\eta = \theta^T x \qquad x \in \mathbb{R}^d$
   $\qquad\qquad\qquad \theta \in \mathbb{R}^d$

3) Test time: output $h_\theta(x) = E[y \mid x;\theta]$



Goal: train θ to predict parameter of exp fam
whose mean is hypothesis of parameter to be output

At train time:

$$\max_\theta \log p(y^{(i)}; \theta^T x^{(i)})$$ via gradient descent

Learning update rule:

$$\theta_j := \theta_j + \alpha\left(y^{(i)} - \underbrace{h_\theta}_{\text{plug in appropriate } h_\theta(x)}(x^{(i)})\right) x_j^{(i)}$$

Terminology: η natural parameter

$$\mu = E[y;\eta] = g(\eta) \rightarrow \text{Canonical response fn}$$

$$\eta = g^{-1}(\mu)$$

$$g(\eta) = \frac{\partial}{\partial \eta} a(\eta)$$

3 parametrizations

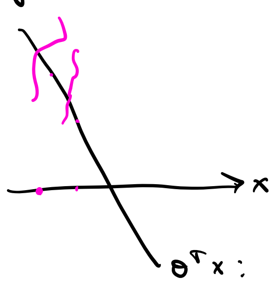
Model parameters		Natural param		Canonical param
θ	$\xleftrightarrow[\theta^T x]{}$	η	$\xrightarrow{g}$ / $\xleftarrow[g^{-1}]{}$	ϕ : Bernoulli
$\uparrow$				μ, σ^2: Gaussian
learning	Design choice			λ : poisson

Logistic regression in GLM framework

$$h_\theta(x) = E[y|x;\theta] = \phi = \frac{1}{1+e^{-\eta}} = \frac{1}{1+e^{-\theta^T x}}$$

Assumptions

Regression



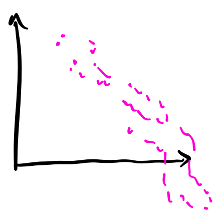
$\theta^T x : \eta = \mu$

Data generation

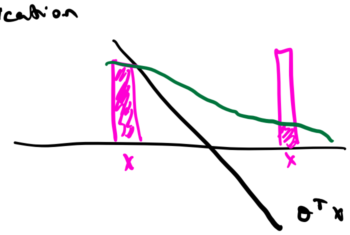
$\Longrightarrow$

find θ

$\Longleftarrow$



Classification



$$\theta^T x = \eta \xrightarrow{\frac{1}{1+e^{-\eta}}} \phi$$

Softmax Regression

Member of GLM family

Cross Entropy Minimization

Multiclass Classification

$x^{(i)} \in \mathbb{R}^d$

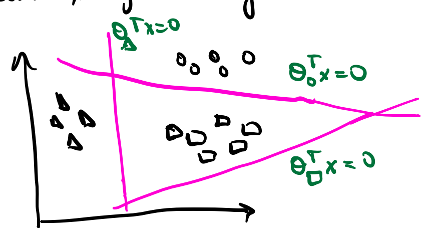
Label $y \in \{0,1\}^k$ i.e. $[0, 0, 1, 0]$ "one-hot vector"

$\theta_{class} \in \mathbb{R}^d$

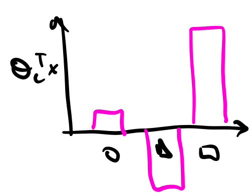
class $\in \{\triangle, \circ, \square, \dots\}$

$$\begin{array}{cc} & \leftarrow d \rightarrow \\ \begin{array}{c}\uparrow \\ k \\ \downarrow\end{array} & \begin{bmatrix} \leftarrow \theta_1 \rightarrow \\ \leftarrow \theta_2 \rightarrow \\ \vdots \end{bmatrix} \end{array}$$

Softmax regression: generalization of logistic regression

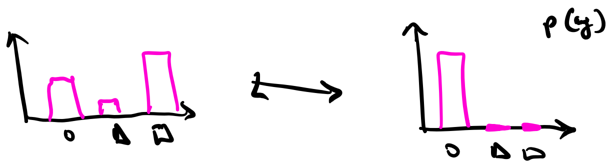


Given x



logit space

once exp'd, normalized



Goal: minimize dist. between distributions by minimizing cross entropy

Cross Entropy

$$\text{CrossEntropy}(p, \hat{p}) = -\sum_{y \in \{\triangle, \bigcirc, \square\}} p(y) \log(\hat{p}(y))$$

$$= -\log \hat{p}(y_{\bigcirc})$$

$$= -\log \frac{e^{\theta_{\bigcirc}^T x}}{\sum_{i \in \{\triangle, \bigcirc, \square\}} e^{\theta_i^T x}}$$

Minimize via gradient descent.