Lecture 4 [Perceptron & Generalized Linear Model]

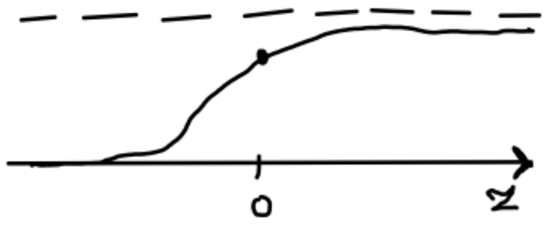
Perceptron:

The perceptron algorithm is not something widely used in practice, we study it mostly for historical reasons and it's easy to analyze

Logistic Regression uses the Sigmoid function

Logistic Regression

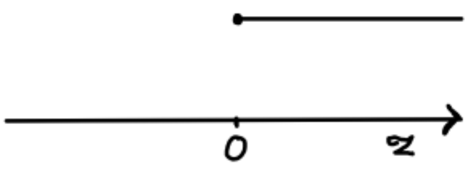
Sigmoid



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

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

Perceptron

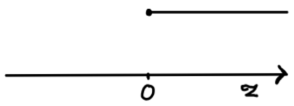


$$g(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$$h_\theta(x) = g(\theta^T x)$$

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

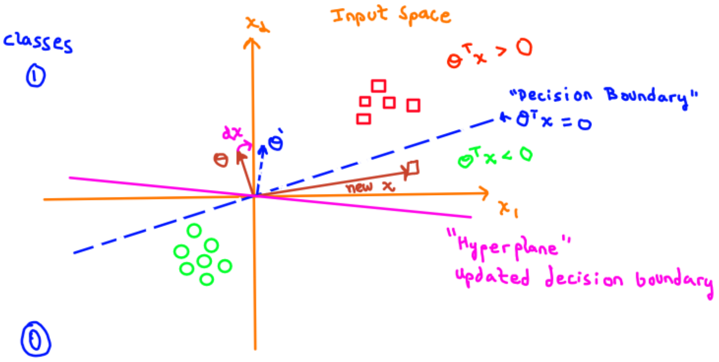
Perceptron



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

$$h_\theta(x) = g(\theta^T x)$$

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



We want $\theta \approx x \mid y = 1$

$\theta \not\approx x \mid y = 0$

$0 \rightarrow$ algorithm got it right

$\pm 1 \rightarrow$ 1 if wrong $y^{(i)} = 1$

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

As Θ is rotating, the decision boundary is going to be perpendicular to Θ

You go example by example in an on-line manner and if the example is already classified, you do nothing. If it is mis-classified, you either add or subtract the vector/example itself to Θ

Exponential Family:

It's essentially a class of probability distributions which are somewhat nice mathematically. They are also very close to GLMs (Generalized Linear Models)

An **Exponential Family** is one whose **PDF** (Probability Density Function, for discrete distribution it would be the Probability Mass Function, **PMF**) can be written in the form:

Exponential Family

PDF

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

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

y - Data

"eta" η - natural parameter

$T(y)$ - sufficient statistic

$\left.\begin{matrix}\eta \\ T(y)\end{matrix}\right\}$ dimensions must match

$b(y)$ - base measure

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

You can think of this as a normalizing constant of the distribution such that the whole thing integrates to 1

The partition-function is a technical term to indicate the normalizing constant of probability distributions

You can plug in any definition of b, a, and T as long as the expression integrates to 1

To show that a distribution is in the exponential family, the most straightforward way to do it is to write out the PDF of the distribution in a form that you know and just do some algebraic massaging to bring it into this form, and then you do a pattern match to conclude that it's a member of the exponential family

Bernoulli distribution (used to model binary data)

parameter ϕ = probability of event
phi

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

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

$$= \underbrace{1}_{b(y)} \exp\left[\underbrace{\log\left(\frac{\phi}{1-\phi}\right)}_{\eta} \underbrace{y}_{T(y)} + \underbrace{\log(1-\phi)}_{a(\eta)}\right]$$

$b(y) = 1$

$T(y) = y$

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

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

A Gaussian distribution has 2 parameters, the mean and the variance {you can have guassians where the variance is a variable}

Gaussian distribution (with fixed variance)

Assume $\overset{\text{"sigma"}}{\sigma}^2 = 1$

$$P(y; \overset{\text{"Mu"}}{\mu}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y-\mu)^2}{2}\right)$$

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

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

$$T(y) = y$$

$$\eta = \mu$$

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

If the variance is unknown, you can write it as an **Exponential Family**, in which case η will now be a vector instead of a scalar and you will also have a mapping between each of the canonical parameters and each of the natural parameters

Properties

ⓐ MLE w.r.t. $\eta \Rightarrow$ concave

"negative log likelihood" NLL is convex

ⓑ "expectation"

$$E[y;\eta] = \frac{\partial}{\partial \eta} a(\eta)$$

ⓒ "variance"

$$Var[y;\eta] = \frac{\partial^2}{\partial \eta^2} a(\eta)$$

you only need to differentiate; no integrals

If we perform **Maximum Likelihood** on the **Exponential Family**, when the exponential family is parameterized in the natural parameters, then the optimization problem is concave.

Similarly if you flip the sign and use what's called the **negative log likelihood** {you take the log of the expression, negate it, and in this case the **NLL** is like the **cost function** equivalent of doing **maximum likelihood**}. So you're just flipping the sign, instead of maximizing you minimize the **NLL**

Generalized Linear Models:

The **GLM** (Generalized Linear Model) is somewhat like a natural extension of the exponential families to include co-variates or include your input features in some way

We can build many powerful models by choosing an appropriate family in the exponential family and plugging it onto a linear model

Generalized Linear Models

"Assumptions/Design choices"

(i) $y \mid x ; \theta \overset{\text{"a member of"}}{\sim} \text{Exponential Family}(\eta)$

In a particular scenario, Exponential Family( ) could take on other distributions part of the exponential family

###
If you have:

DATA |	Distributions	
Real	Gaussian	(real means: any value between 0 and infinity)
Binary	Bernoulli	
Count	Poisson	(count means: non negative integers)
R^+	Gamma, Exponential	(positive real valued integers)

You can also have probability distributions over probability distributions:

Distn	Beta, Dirichlet	{mostly show up in Bayesian machine learning or Bayesian statistics}

###

At test time when we want an output for a new x [given a new x we want to make an output], the output will be

Generalized Linear Models

"Assumptions/Design choices"

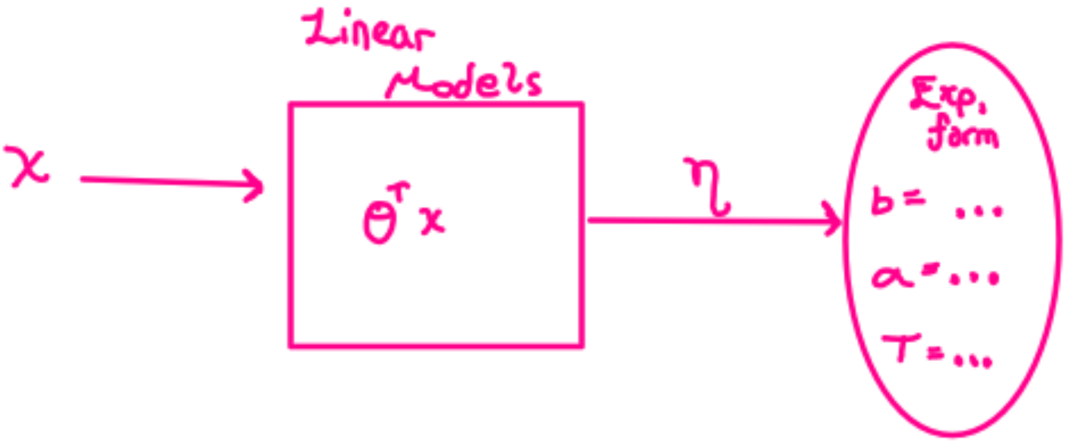
(i) $y | x ; \theta \overset{\text{"a member of"}}{\sim}$ Exponential Family (η)

(ii) $\eta = \theta^T x \qquad \theta \in \mathbb{R}^n, \; x \in \mathbb{R}^n$

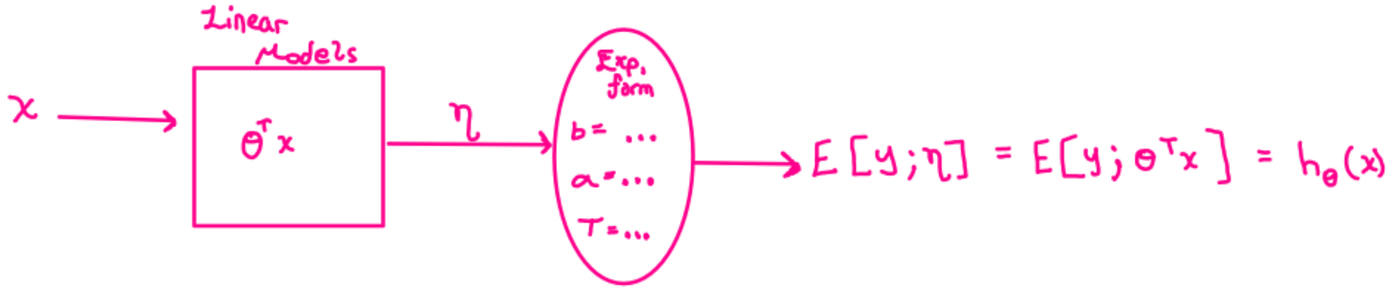
(iii) Test time : Output $E[y | x ; \theta]$

$$\Rightarrow h_\theta(x) = E[y | x ; \theta]$$

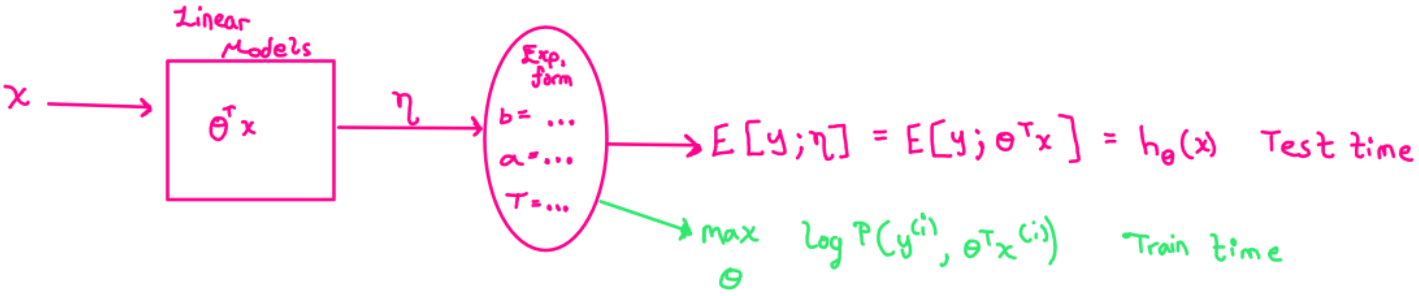
Given an x, we get an exponential family distribution and the mean of that distribution will be the prediction that we make for a given x



Given x, there is a learnable parameter **Θ** and **Θ^{T}x** will give you a parameter **η**. The distribution is a member of the exponential family and the parameter for this distribution is the output of the linear model. Depending on the data we have, you would choose an appropriate **b**, **a**, and **T** based on the distribution of your choice



We are training Θ to predict the paramter of th exponential family distribution whose mean is the prediction we're going to make for y


Linear Models
x
$\theta^T x$
η
Exp. form
b = ...
a = ...
T = ...
$E[y;\eta] = E[y;\theta^T x] = h_\theta(x)$
Test time
$\max_\theta \log P(y^{(i)}, \theta^T x^{(i)})$
Train time

During training/learning, we do Maxmimum Likelihood with respect to Θ

GLM Training

Learning Update Rule is the same for any chosen distribution

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

This is the same update rule for any specific type of GLM based on the choice of distribution that you have, whether you're doing classification, regression, poisson regression, the update rule is the same

η is the natural parameter and the function that links the natural parameter to the mean of the distribution (**Canonical Response function**)

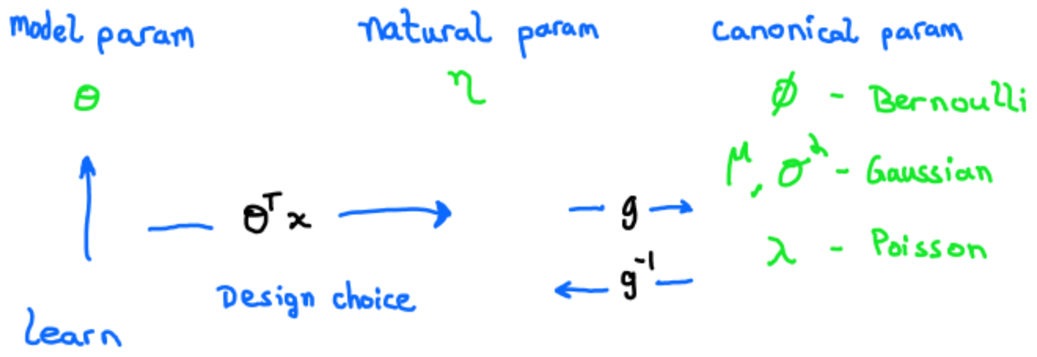
Terminology

η - natural parameter

$\mu \quad E[y;\eta] = g(\eta)$ → canonical response function $\quad g(\eta) = \frac{\partial}{\partial\eta} a(\eta)$

$\eta = g^{-1}(\mu)$ → canonical link function

3 - parameterizations



Logistic Regression

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

The choice of what distribution you're going to choose is really dependent on the task that you have

So if your task is regression, where you want to output real valued numbers, then you choose a distribution over the real numbers like a Gaussian

If your task is classification, where your output is binary 0 or 1, you choose a distribution that models binary data

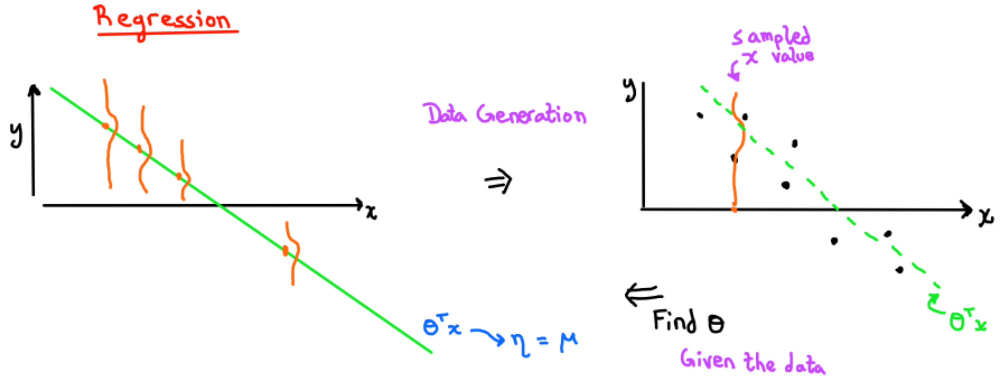
If you want to model the number of visitors to website, which is like a count, you'd want to use a poisson distribution because a poisson distribution is a distribution over integers

“Are GLMs used for classification, or are they used for regression, or are they used for something else?”

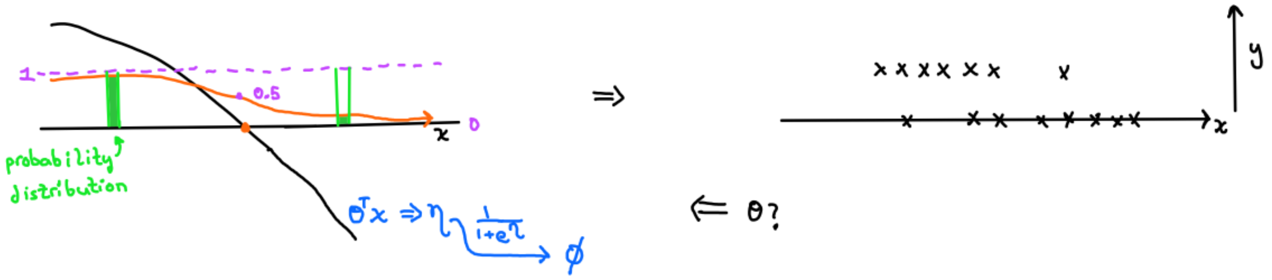
The answer really depends on what the choice of distribution is. GLM are just a general way to model data

Assumptions

Regression



Classification

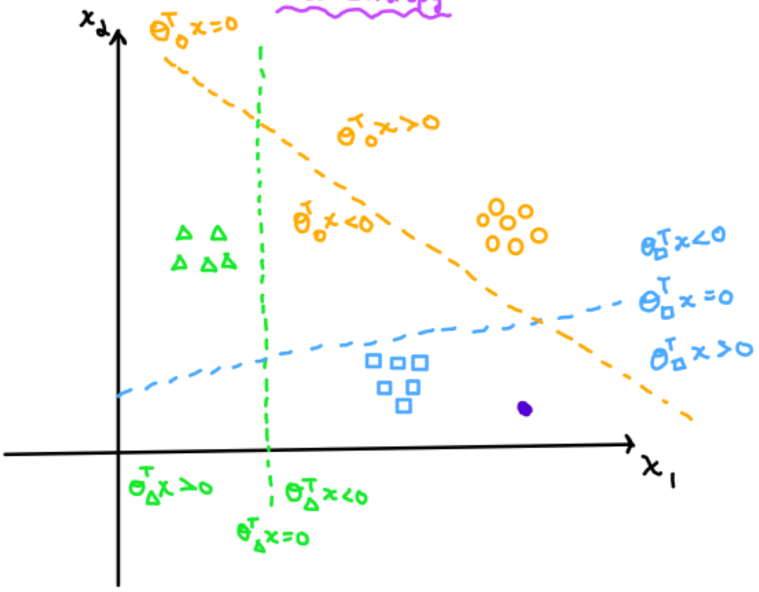


Softmax Regression (Multiclass Classification):

Cross entropy

Softmax Regression

Cross Entropy



k - # of classes

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

Label $y = [\{0,1\}^k]$ e.g. $[0,0,1,0]$ one-hot vector

$\theta_{class} \in \mathbb{R}^n$, k classes

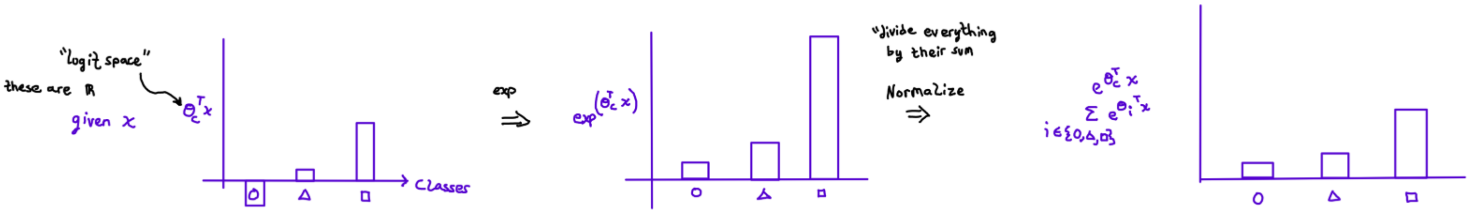
class $\in \{△, ○, □, \ldots\}$

- Can be represented as an nxk matrix

$$\begin{array}{c|c} & n \\ k & \begin{bmatrix} — \theta_c — \\ — \theta_c — \end{bmatrix} \end{array}$$

In softmax regression, it's a generalization of logistic regression where you have a set of parameters per class

The goal is to get a probability distribution over the classes

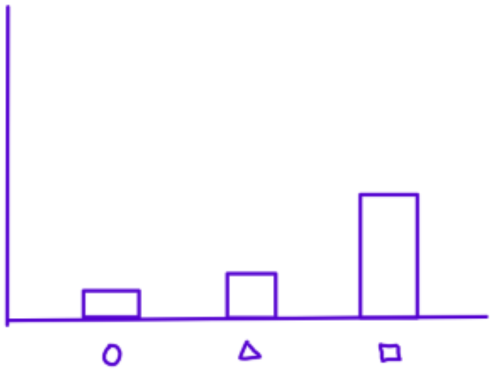

"logit space"
these are ℝ
given x
$\theta_c^T x$
○
△
□
Classes
exp
⇒
$exp(\theta_c^T x)$
○
△
□
"divide everything by their sum"
Normalize
⇒
$\frac{e^{\theta_c^T x}}{\sum_{i \in \{○,△,□\}} e^{\theta_i^T x}}$
○
△
□

After normalizing, you will get a probability distribution where the sum of the heights will add up to 1

Given a new point x and we run through this pipeline, we get a probability output over the classes for which class that example is most likely to belong to "hypothesis function"

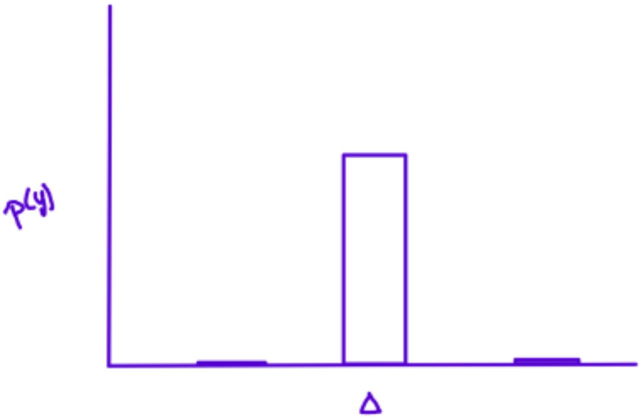
$\hat{p}(y)$ for the given x

$$\frac{e^{\theta_c^T x}}{\sum_{i \in \{○,△,□\}} e^{\theta_i^T x}}$$



$\updownarrow$ minimize the cross entropy between the two distributions

$p(y)$ "Label"



$$\text{Cross Entropy } \overset{\text{"between"}}{(p, \hat{p})} = \sum_{y \in \{\circ, \triangle, \square\}} p(y) \log \hat{p}(y)$$

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

$$= -\log \frac{e^{\theta_{\triangle}^{T} x}}{\sum_{c \in \{\circ, \triangle, \square\}} e^{\theta_{c}^{T} x}}$$

↑ do gradient descent w.r.t. the parameters