Bhishan Poudel

106+6 / 120

HW Assignment 3 (Due by 10:30am on Oct 12)

1 Theory (100 points)

1. [**Maximum Likelihood**, 20 points]
   The Poisson distribution specifies the probability of observing k events in an interval, as follows:
   $$P(k \text{ events in interval}) = e^{-\lambda}\frac{\lambda^k}{k!} \tag{1}$$
   For example, k can be the number of meteors greater than 1 meter diameter that strike Earth in a year, or the number of patients arriving in an emergency room between 10 and 11 pm[1].

   Suppose we observe N samples k_1, k_2, ..., k_N from this distribution (i.e. numbers of meteors that strike Earth over a period of N years). Derive the maximum likelihood estimate of the event rate λ.

2. [**Logistic Regression**, 20 points]
   Consider a dataset that contains the 4 examples below i.e., the truth table of the logical XOR function. Prove that no logistic regression model can perfectly classify this dataset. Do not forget the bias feature $x_0 = 1$.

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	0

   Hint: Prove that there cannot be a vector of parameters $\mathbf{w}$ such that $P(t = 1|\mathbf{x}, \mathbf{w}) \geq 0.5$ for all examples $\mathbf{x}$ that are positive, and $P(t = 1|\mathbf{x}, \mathbf{w}) < 0.5$ for all examples $\mathbf{x}$ that are negative.

3. [**Logistic Regression**, 20 points]
   Prove that the gradient (with respect to $\mathbf{w}$) of the negative log-likelihood error function for logistic regression corresponds to the formula shown in lecture 4:
   $$\nabla_{\mathbf{w}} E(\mathbf{w}) = \sum_{n=1}^{N}(h_n - t_n)\mathbf{x}_n \tag{2}$$

4. [**Logistic Regression**, 20 points]
   In `scikit`, the objective function for logistic regression expresses the trade-off between training error and model complexity through a parameter C that is multiplied with the error term, as shown below. See the `scikit` documentation at `http://scikit-learn.org/stable/modules/linear_model.html#logistic-regression`.
   $$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C * \sum_{n=1}^{N}\ln(e^{-t_n(\mathbf{w}^T\mathbf{x}_n)} + 1) \tag{3}$$

[1] https://en.wikipedia.org/wiki/Poisson_distribution

- Show that the sum in the second term is equal with the negative log-likelihood, where $t_n = +1$ stands for positive labels and $t_n = -1$ stands for negative labels.
- Compute the C parameter such that the objective is equivalent with the standard formulation shown on the slides in which the regularization parameter λ is multiplied with the L2 norm term.

5. [**Softmax Regression**, 20 points]
Show that Logistic Regression is a special case of Softmax Regression. That is to say, if $\mathbf{w_1}$ and $\mathbf{w_2}$ are the parameter vectors of a Softmax Regression model for the case of two classes, then there exists a parameter vector $\mathbf{w}$ for Logistic Regression that results in the same classification as the Softmax Regression model.

6. [**Softmax Regression (*)**, 20 points]
Prove that the gradient (with respect to $\mathbf{w}_k$) of the negative log-likelihood error function for regularized softmax regression corresponds to the formula shown in lecture 4, for any class $k \in [1..K]$:

$$\nabla_{\mathbf{w}_k} E(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}(\delta_k(t_n) - p(C_k|\mathbf{x}_n))\mathbf{x}_n + \alpha \mathbf{w}_k \tag{4}$$

2 Submission

Turn in a hard copy of your homework report at the beginning of class on the due date. On this theory assignment, **clear and complete explanations and proofs of your results are as important as getting the right answer.**

HW03

Bhishan Poudel

QN1

QN 1

maximum Likelihood

(Derive MLE of event rate parameter λ)

Let $x_1, x_2, \ldots x_n$ be iid poisson random variables with prob mass function,

pmf of poisson distn

$$\boxed{p(x_i, \lambda) = \frac{e^{-\lambda} \lambda^{x_i}}{x_i!} = p(x_i)}$$

Now, the joint prob of all variables x_i, called likelihood function, is given by

Likelihood

$$\boxed{L(\lambda) = \prod_{i=1}^{n} p(x_i, \lambda)}$$

$$= \prod_{i=1}^{n} \frac{e^{-\lambda} \lambda^{x_i}}{x_i!}$$

The log likelihood of pmf is,

Log Likelihood

$$\boxed{\ln L(\lambda) = \ln \prod_{i=1}^{n} \frac{e^{-\lambda} \lambda^{x_i}}{(x_i)!}}$$

$$= \sum_i \ln \frac{e^{-\lambda} \lambda^{x_i}}{x_i!}$$

$$= \sum_i [-\lambda + x_i \ln \lambda - \ln x_i!]$$

$$\boxed{\ln L(\lambda) = -n\lambda + \ln\lambda \sum_i x_i - \sum_i \ln (x_i)!}$$

To get the maximum likelihood estimate of the parameter λ, we maximize the Log likelihood $(\ln L(\lambda))$ w.r.t. λ. [maximizing $(+\ln L)$ is same as minimizing $(-\ln L)$]

$$0 = \frac{\partial}{\partial \lambda} \ln L(\lambda)$$

$$= \frac{\partial}{\partial \lambda} \left[-n\lambda + \ln \lambda \sum_i x_i - \sum_i \ln x_i! \right]$$

$$0 = -n + \frac{\sum_i x_i}{\lambda} - 0$$

$$n = \frac{\sum_i x_i}{\lambda}$$

MLE of λ

$$\boxed{\hat{\lambda} = \frac{\sum_{i=1}^{n} x_i}{n}}$$

$n \rightarrow$ number of samples

Ans

20

Here, the maximum likelihood estimate of the poisson distribution parameter λ is just the mean (or expectation) of the distribution.

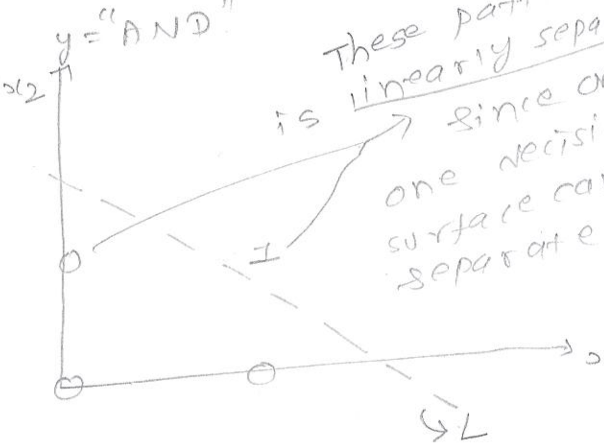
QN 2

QN 2

XOR problem in Logistic Regression

To describe XOR problem in L/R, I shall start with "AND" logic.

x_1	x_2	$y = x_1 \& x_2$
0	0	0
0	1	0
1	0	0
1	1	1



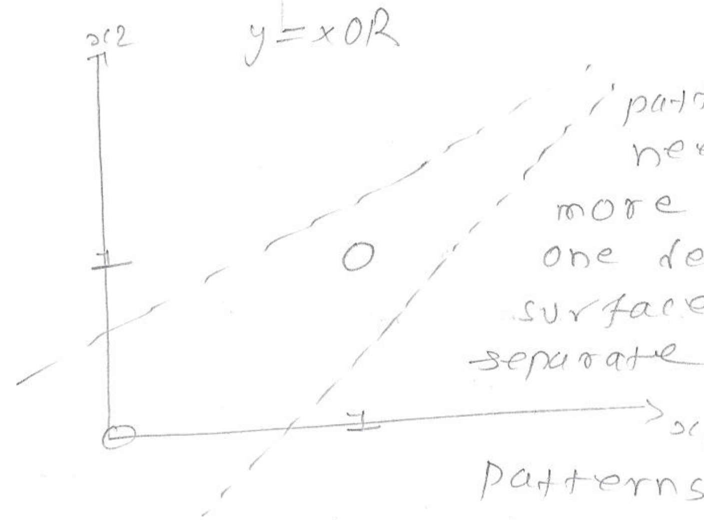
These pattern is linearly separable since only one decision surface can separate them

↳ L decision surface.

Now, look at XOR,

Truth table

x_1	x_2	$y = XOR$
0	0	0
0	1	1
1	0	1
1	1	0



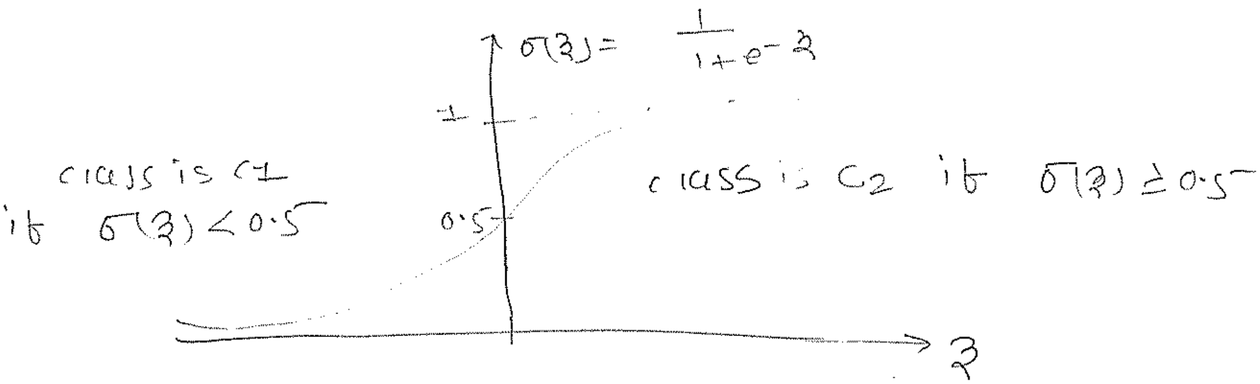
pattern needs more than one decision surface to separate patterns of 0's and 1's.

∴ XOR logic is not linearly separable.

P.T.O.

Now, we shall show that Logistic Regression is linear classifier :

The classifier used in LR is sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$



In Logistic Regression,

pattern is 1 if $\frac{1}{1+e^{-w^T x}} \geq \frac{1}{2}$

0 if $\frac{1}{1+e^{-w^T x}} < \frac{1}{2}$

let's set the ~~classifier~~ equal to separator value:

$$\frac{1}{1+e^{-w^T x}} = \frac{1}{2}$$

$$2 = 1+e^{-w^T x}$$

$$\frac{1}{1} = e^{-w^T x} = \frac{1}{e^{w^T x}}$$

$$e^{w^T x} = 1$$

$$w^T x = \ln 1 = 0$$

$$\Rightarrow \boxed{\sum_i w_i x_i = 0}$$

this means LR is linear classifer.

conclusion: 1) LR is linear classifer

2) XOR problem is non-linear and linearly inseparable

$\therefore$ LR can NOT perfectly classify dataset that follows XOR logic.

Ans

12

use:

$W^T x_n \geq 0$ if $x_n \in +1$

$W^T x_n < 0$ if $x_n \in -1$

then put 4 samples into the inequations.

Better method

x_0	x_1	x_2	t
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

for linear separability, we want

$$w_0 + \sum_i w_i x_i < 0 \text{ if } t_i == 0$$

$$w_0 + \sum_i w_i x_i \geq 0 \text{ if } t_i == 1$$

Now,

$$w_0 < 0 \quad \text{— (1)}$$

$$w_0 + w_2 \geq 0 \quad \text{— (2)}$$

$$w_0 + w_1 \geq 0 \quad \text{— (3)}$$

$$w_0 + w_1 + w_2 < 0 \quad \text{— (4)}$$

(2), (3): $w_0 + (w_0 + w_1 + w_2) \geq 0$

but $w_0 < 0$

and $w_0 + w_1 + w_2 < 0$

$\therefore$ it contradicts our assumptions of linear separability.

QN3

QN3

Gradient of Logistic Regression cost f^n

(find $\nabla_w E$)

for linear regression, hypothesis $h = w^T x$

for logistic regression, hypothesis $h = \sigma = \frac{1}{1+e^{-w^T x}}$

$$\boxed{h = \sigma = \frac{1}{1+e^{-wx}}} \quad (1)$$

(I write $w^T x$ as wx since they are just matrix dot product of two matrices)

Aside:
Binomial distn
$p(x; p, n) = {}^nC_x\, p^x (1-p)^{n-x}$
for $x \in (0 \ldots n)$
${}^nC_x = \frac{n!}{x!\,(n-x)!}$

Binomial (Bernouli) Likelihood Function

pmf for Logistic Regression
$p(t|w) = h_n^{t_n} \cdot (1-h_n)^{1-t_n}$

likelihood function $\boxed{L(w) = \prod_{n=1}^{N} h_n^{t_n} (1-h_n)^{1-t_n}}$

(Bishop eq 4.89 p. 206/223)

-ve log likelihood, E or $J = -\ln L(w)$

$$E = -\ln \prod_{n=1}^{N} h_n^{t_n} (1-h_n)^{1-t_n}$$

$$E = \sum_n E_n = \sum_{n=1}^{N} (-t_n \ln h_n - (1-h_n) \ln (1-h_n))$$

Loss, $E(w) = -\frac{1}{N} \sum_n [t_n \ln h_n + (1-t_n) \ln (1-h_n)]$

for any nth sample the cost can be written as,

LOSS $\boxed{E = -t \ln h - (1-h) \ln (1-h)} \quad (2)$

-ve log likelihood error fn for LR

here $h = \sigma = \sigma(w^T x) = \sigma(wx)$

Before proceeding further, I would derive derivative of sigmoid function.

$$\sigma(z) = \frac{1}{1+e^{-z}} \Rightarrow \frac{d\sigma}{dz} = \frac{-1 \cdot e^{-z} \cdot (-1)}{(1+e^{-z})^2}$$

$$\frac{d\sigma}{dz} = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} \cdot \frac{(e^{-z}+1)-1}{1+e^{-z}}$$

$$= \frac{1}{1+e^{-z}} \cdot \left(\frac{e^{-z}+1}{1+e^{-z}} - \frac{1}{1+e^{-z}} \right)$$

$$= \frac{1}{1+e^{-z}} \cdot \left(1 - \frac{1}{1+e^{-z}} \right)$$

$$\boxed{\frac{d\sigma(z)}{dz} = \sigma(1-\sigma)}$$

similarly $\boxed{\frac{d\sigma(az)}{dz} = a\sigma(1-\sigma)}$ (here $a \neq a(z)$)

$\therefore$ derivative of sigmoid function $\boxed{\frac{d\sigma(az)}{dz} = a\sigma(1-\sigma)}$

Now, going back to the problem, let's calculate the gradient of loss function.

$$\frac{\partial E}{\partial w} = \frac{\partial}{\partial w}\left[-t\ln h - (1-t)\ln(1-h)\right]$$

$$= -\frac{t}{h}\frac{\partial h}{\partial w} - \frac{(1-t)(-1)}{1-h}\cdot\frac{\partial h}{\partial w}$$

$$= -\frac{t}{h}\frac{\partial \sigma(wx)}{\partial w} + \frac{1-t}{1-h}\frac{\partial}{\partial w}\sigma(wx)$$

Note $h = \sigma(w$

σ and h are same thing here

property of sigmoid function

Note: $\boxed{\frac{\partial h}{\partial w} = \frac{\partial h(xw)}{\partial w} = xh(1-h)}$

$$= -\frac{t}{\cancel{h}} \, x \, \cancel{h}(1-h) + \frac{1-t}{\cancel{(1-h)}} \, x \, h\cancel{(1-h)}$$

$$\because \frac{d\sigma(az)}{dz} = a\sigma(1-\sigma)$$

$$= -tx - \cancel{txh} + xh - \cancel{txh}$$

$$= xh - tx$$

$$\boxed{\frac{\partial E}{\partial w} = (h-t)\,x}$$

(here this is for a single sample, but for total loss we add up all the losses)

$$\Rightarrow \boxed{\nabla_w E = \sum_{n=1}^{N} (h_n - t_n)\, x_n}$$

Q.E.D.

QN4

QN 4

Logistic Regression in sklearn

part a

The cost function for L2 penalized logistic regression is given by,

L2 Regularized cost fn for LR (in sklearn)

$$\boxed{E = \underbrace{\frac{1}{2} w^T w}_{\text{L2 regularizer}} + \underbrace{C \sum_{n=1}^{N} \ln \left(1 + e^{-t_n w^T x_n}\right)}_{\text{cost for LR}}}$$

where $t_n \in \{-1, 1\}$ — (1)

we have to show that the summation term is equal to the -ve log likelihood,

-ve log likelihood

$$\boxed{E_D = -\ln p = \sum_{n=1}^{N} \ln \left(1 + e^{-t_n w^T x_n}\right)} \quad (2)$$

$t_n \in \{0, 1\}$

Now, the posterior probabilities for class 0 and 1 are,

posterior prob for LR

$$p(C_1 | x) = \sigma(w^T x) = \frac{1}{1 + e^{w^T x}}$$

$$p(C_2 | x) = 1 - \sigma(w^T x) = \frac{e^{-w^T x}}{1 + e^{-w^T x}} \quad (3)$$

the likelihood function is,

Likelihood for LR

$$\boxed{p(t | w) = \prod_{n=1}^{N} h_n^{t_n} (1 - h_n)^{(1 - t_n)}} \quad (4)$$

The -ve log likelihood is,

$$-\ln p(t|w) = -\ln \prod_{n=1}^{N} h_n^{t_n} \cdot (1-h_n)^{1-t_n} = -\sum_{n=1}^{N} \ln (h_n^{t_n} \cdot (1-h_n)^{1-t_n})$$

$$= -\sum_{n=1}^{N} \left[\ln (h_n^{t_n}) + \ln (1-h_n)^{1-t_n} \right]$$

cost for LR

$$\boxed{E_D = -\ln p = -\sum_{n=1}^{N} [\underbrace{t_n \ln h_n}_{\text{for class 0}} + \underbrace{(1-t_n) \ln (1-h_n)}_{\text{for class 1}}]} \quad \text{--- (5)}$$

where $t_n \in \{0, 1\}$

$$\boxed{h_n = \frac{1}{1+e^{-w^T x_n}}}$$

$$1-h_n = 1 - \frac{1}{1+e^{-w^T x_n}}$$

$$= \frac{1+e^{-w^T x_n} - 1}{1+e^{-w^T x_n}}$$

$$= \frac{e^{-w^T x_n}}{1+e^{-w^T x_n}}$$

$$= \frac{1}{e^{w^T x_n} + 1}$$

$$\boxed{1-h_n = \frac{1}{1+e^{w^T x_n}}}$$

$$E_D = -\ln p$$

$$= -\sum_{n=1}^{N}\left[t_n \ln\left(\frac{1}{1+e^{-w^T x_n}}\right) + (1-t_n)\ln\left(\frac{1}{1+e^{w^T x_n}}\right)\right]$$

$t_n \in \{0,1\}$

$$= -\sum_{n=1}^{N}\begin{cases}\ln \dfrac{1}{1+e^{-w^T x_n}} & \text{if } t_n = 1\\ \ln \dfrac{1}{1+e^{w^T x_n}} & \text{if } t_n = 0\end{cases}$$

$$= -\sum_{n=1}^{N} \ln \frac{1}{1+e^{-t_n w^T x_n}} \quad \text{if } t_n \in \{-1,1\}$$

14

$$\boxed{E_D = +\sum_{n=1}^{N} \ln\left(1+e^{-t_n w^T x_n}\right)} \quad \text{Q.E.D.}$$

$t_n \in \{-1,1\}$

QN 4b   Find 'C' parameter of LR cost f^n in sklearn.

Now, the L2 regularized cost function for LR

$$E = E_W + E_D$$

$$= \frac{\lambda}{2} w^T w + (-\ln p)$$

~~Regularized cost~~

when $t_n \in \{0,1\}$

$$\boxed{E = \frac{\lambda}{2} w^T w - \sum_{n=1}^{N} \left[t_n \ln h_n + (1-t_n) \ln (1-h_n) \right]} \quad (0)$$

$t_n \in \{0,1\}$

Regularized cost

when $t_n \in \{-1,1\}$

$$\boxed{E = \frac{\lambda}{2} w^T w + \sum_{n=1}^{N} \ln (1 + e^{-t_n w^T x_n})} \quad (1)$$

$t_n \in \{-1,1\}$

when $t_n \in \{-1,1\}$ in sklearn

$$\boxed{E(w) = \frac{1}{2} w^T w + C \sum_{n=1}^{N} \ln (1 + e^{-t_n (w^T x_n)})} \quad (2)$$

To get maximum Likelihood Estimate of parameter w,

$$\vec{w} = \underset{w}{\operatorname{argmin}} \left[\frac{\lambda}{2} w^T w + \sum_{n=1}^{N} \ln (1 + e^{-t_n w^T x_n}) \right]$$

$$\Rightarrow \vec{w} = \underset{w}{\operatorname{argmin}} \left[\frac{1}{2} w^T w + \frac{1}{\lambda} \sum_{n=1}^{N} \ln (1 + e^{-t_n w^T x_n}) \right] \quad \text{(general form)}$$

$$\text{but, } \vec{w} = \underset{w}{\operatorname{argmin}} \left[\frac{1}{2} w^T w + C \sum_{n=1}^{N} \ln (1 + e^{-t_n w^T x_n}) \right] \quad \text{(scikit form)}$$

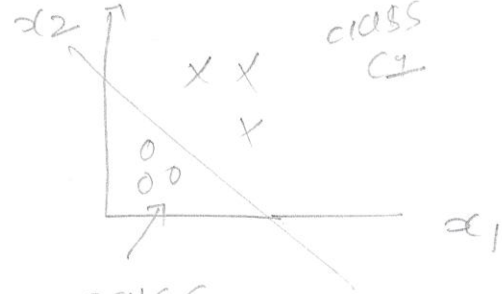
(b) $\therefore$ $\boxed{C = \frac{1}{\lambda}}$ ✓ Ans

Extra note: To add bias term sklearn uses $w^T \to w^T - c$ in the data term E_D

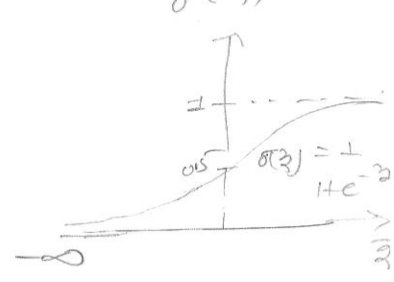
QN 5 ANS

softmax Regression as special case of Logistic Regression

FOR LR



(binary classification)



(sigmoid fn)

the posterior prob of class C1 is

posterior prob

$$\boxed{p(c_1 \mid x; w) = p(c_1 \mid x) = \sigma(w^T x) = \frac{1}{1+e^{-w^T x}}}$$

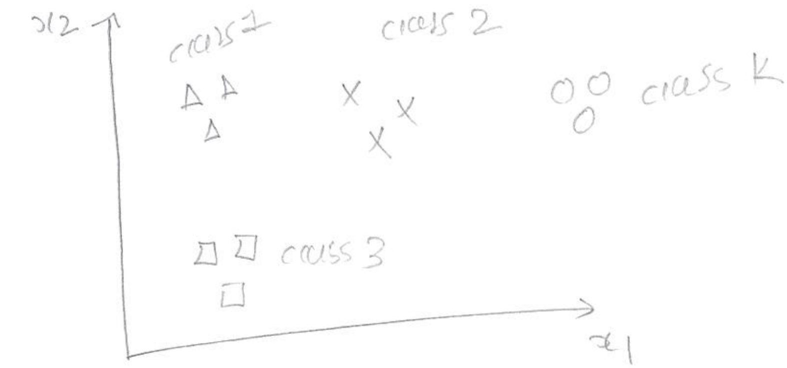
the posterior prob of class C2 is,

$$p(c_2 \mid x) = 1 - \sigma(w^T x) = \frac{e^{-w^T x}}{1+e^{-w^T x}}$$

hypothesis — the hypothesis for LR,

$$h(w) = \begin{bmatrix} \frac{1}{1+e^{-w^T x}} \\ \frac{e^{-w^T x}}{1+e^{-w^T x}} \end{bmatrix} \begin{matrix} \rightarrow \text{for class 1} \\ \rightarrow \text{for class 2} \end{matrix}$$

Again, <u>For SR</u> (softmax Regression or multiclass Logistic Regression)



total number of samples = N

input data X   target

$$\begin{bmatrix} x_1 & \text{some values} \\ x_2 & \\ \vdots & \\ x_N & \end{bmatrix} \quad \begin{bmatrix} \text{triangle } t_1 \\ \text{cross } t_2 \\ \text{rectangle} \\ \cdots \\ \text{circle } t_N \end{bmatrix}$$

In softmax Regression, the posterior probability of any class C_K is given by,

$$\boxed{p(C_K|x, w) = \frac{e^{w_K^T x}}{\sum_{K=1}^{K} e^{w_K^T x}}}$$

posterior prob.

[Bishop, 4.3.4 eq 4.104 p. 209 / pdf 227]

<u>note</u> $a_K = w_K^T x$ is called activation f^n for the parameter vector w_K

hyp:

$$\boxed{h = \frac{1}{\sum_K e^{w_K^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \\ \vdots \\ e^{w_K^T x} \end{bmatrix}}$$

plus sign $+w_K^T x$

sigmoid is −
softmax is +

Show $h(w) = h(w - \psi)$

[SR has overparameterization property]

ψ is any fixed vector

$$h(w - \psi) = \frac{e^{(w_K - \psi)^T x}}{\sum_{K=1}^{K} e^{(w_K - \psi)^T x}}$$

$$= \frac{e^{w_K^T x} \cdot e^{-\psi^T x}}{\sum_K e^{w_K^T x} \cdot e^{-\psi^T x}}$$

$$= \frac{e^{w_K^T x} \cdot \cancel{e^{-\psi^T x}}}{\underbrace{\cancel{e^{-\psi^T x}}}_{\text{does not depend on } K} \sum_K e^{w_K^T x}}$$

$$= \frac{e^{w_K^T x}}{\sum_K e^{w_K^T x}}$$

$$\boxed{h(w - \psi) = h(w)}$$

$\therefore$ If we change any parameter vector $w_K \to w_K - \psi$ we get <u>same</u> hypothesis for softmax Regression.

for two-class SR,

$$h = \frac{1}{\sum_{K} e^{w_K^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \end{bmatrix}$$

$$= \frac{1}{e^{w_1^T x} + e^{w_2^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \end{bmatrix}$$

use overparametrization property of SR
$h(w - \psi) = h(w)$ where $\psi = w2$

$$= \frac{1}{e^{(w_1 - w_2)^T x} + 1} \begin{bmatrix} e^{(w_1 - w_2)^T x} \\ 1 \end{bmatrix}$$

write $\vec{w_2} - \vec{w_1} = \vec{w}$

$$= \frac{1}{1 + e^{-w^T x}} \begin{bmatrix} e^{-w^T x} \\ 1 \end{bmatrix}$$

$$(h)_{SR} = \begin{bmatrix} \frac{e^{-w^T x}}{1 + e^{-w^T x}} \\ \frac{1}{1 + e^{-w^T x}} \end{bmatrix} = (h)_{LR} \; \#$$

proved

This is same as h for LR

Direct method to prove softmax (K=2) = LR

the hypothesis for softmax Regression is

$$h(w,x) = \frac{1}{\sum_{k=1}^{K} e^{w_k^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \\ \vdots \\ e^{w_K^T x} \end{bmatrix}$$

when K=2,

$$h = \frac{1}{e^{w_1^T x} + e^{w_2^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \end{bmatrix}$$

$$(h)_{softmax} = \begin{bmatrix} \frac{e^{w_1^T x}}{e^{w_1^T x} + e^{w_2^T x}} \times \frac{e^{-w_2^T x}}{e^{-w_2^T x}} \\ \frac{e^{w_2^T x}}{e^{w_1^T x} + e^{w_2^T x}} \times \frac{e^{-w_2^T x}}{e^{-w_2^T x}} \end{bmatrix}$$

20

$$= \begin{bmatrix} \frac{e^{-(w_2^T - w_1^T)x}}{1 + e^{-(w_2^T - w_1^T)x}} \\ \frac{1}{1 + e^{-(w_2^T - w_1^T)x}} \end{bmatrix} = \begin{bmatrix} \frac{e^{-w^T x}}{1+e^{-w^T x}} \\ \frac{1}{1+e^{-w^T x}} \end{bmatrix}$$

where $w^T = w_2^T - w_1^T$

or,

$$(h)_{softmax} = \begin{bmatrix} \dfrac{e^{-w^T x}}{1+e^{-w^T x}} \\ \\ \dfrac{1}{1+e^{-w^T x}} \end{bmatrix}$$

this is (h) Logistic Regression

$$\therefore \quad \boxed{(h)_{SR} = (h)_{LR}}$$ proved!

QN 6

QN6 Derivative of cost f^n for softmax Regr

For softmax Regression,

prob of data x belonging to class C_K i.e.

posterior prob of class C_K given data x is,

posterior prob of class C_K

$$\boxed{p(C_K|x,w) = \frac{e^{w_K^T x}}{\sum_{K=1}^{K} e^{w_K^T x}}} \quad \text{--- (0)}$$

Bishop 4.3.4
p. 209/227

rename class C_K by target t_n, then the probability that the nth example x_n belong to target t_n is given by,

$$\boxed{p(t_n|x,w) = \frac{e^{w_{t_n}^T x_n}}{\sum_{n=1}^{N} e^{w_{t_n}^T x_n}}} \quad \text{--- (1)}$$

Then, likelihood

$$\boxed{l(w) = \prod_{n=1}^{N} p(t_n|x_n,w)} \quad \text{--- (2)}$$

The -ve log likelihood is,

$$-\ln l(w) = -\ln \prod_{n=1}^{N} \left(\frac{e^{w_{t_n}^T x_n}}{\sum_{n=1}^{N} e^{w_{t_n}^T x_n}} \right)$$

cost for data

$$\boxed{E_D = \frac{1}{N} \cdot \left(-\ln l(w)\right) = \frac{-1}{N} \sum_{n=1}^{N} \ln\left(\frac{e^{w_{t_n}^T x_n}}{\sum_{n=1}^{N} e^{w_{t_n}^T x_n}} \right)} \quad \text{--- (3)}$$

using w_{t_n}

(weight vector for each N examples)

here, t_n is $t_1, t_2, t_3, \ldots, t_N$ (single value)

x_n is $x_1, x_2, x_3, \ldots, x_N$ (x_1 is row vector)

w is $w_1, w_2, \ldots \ldots, w_N$ ($w\,x_n$

(if x_1 is row w_1 is column) should be single value

Here, instead of $\vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_N \end{bmatrix}$ we can group the data so that there are only K weight vectors $w = [w_1\ w_2, \cdots w_K]$, then,

using w_K

$$\boxed{p(C_k|x) = \frac{e^{w_k^T x}}{\sum_{k=1}^{K} e^{w_k^T x}}} \quad (1)$$

(Regrouping weight vectors for each K classes, we can rewrite cost fns)

and,

using w_K (weight vector for each K-classes)

$$\boxed{E_D(w) = -\frac{1}{N} \sum_{n=1}^{N} \underbrace{\sum_{k=1}^{K}}_{\text{insert this}} \delta_k(t_n)\ \ln \left(\frac{e^{w_k^T x_n}}{\sum_{k=1}^{K} e^{w_k^T x_n}} \right)}$$

$$= -\frac{1}{N} \sum_{n=1}^{N} \sum_{k} \delta_k(t_n) \left[w_k^T x_n - \ln \sum_{k=1}^{K} e^{w_k^T x_n} \right]$$

$$E_D = -\frac{1}{N} \sum_{n=1}^{N} \left[\sum_{k=1}^{K} \delta_k(t_n)\, w_k^T x_n - \sum_{k=1}^{K} \delta_k(t_n) \ln \left(\sum_{k=1}^{K} e^{w_k^T x_n} \right) \right]$$

$$\frac{\partial E_D}{\partial w_j} = -\frac{1}{N} \sum_{n=1}^{N} \left[\frac{\partial}{\partial w_j} \sum_{k=1}^{K} \delta_k(t_n)\, w_k^T x_n - \frac{\partial}{\partial w_j} \sum_{k=1}^{K} \delta_k(t_n) \ln \left(\sum_{k=1}^{K} e^{w_k^T x_n} \right) \right]$$

here w_j operates only when $w_j = w_K$ and summation vanishes

$$\frac{\partial E_D}{\partial w_j} = \frac{-1}{N} \sum_{n=1}^{N} \left[\delta_j(t_n)\, x_n - \delta_j(t_n) \cdot \frac{1 \cdot e^{w_j^T x_n}}{\sum_{k=1}^{k} e^{w_j^T x_n}} \cdot \delta_j(t_n)\, x_n \right]$$

$$\frac{\partial E_D}{\partial w_K} = \frac{-1}{N} \sum_{n=1}^{N} \left[\delta_K(t_n) \cdot x_n - x_n \cdot \underbrace{\delta_K(t_n) \frac{e^{w_K^T x_n}}{\sum_{k=1}^{K} e^{w_K^T x_n}}}_{= p(C_K|x_n)} \right]$$

LOOK (eq 1)

we change dummy index from $j \to K$

$\delta_K(t_n) \cdot \delta_K(t_n) = \delta_K(t_n)$

$\delta_K(t_n) \cdot p(C_K)$ is prob only when label is K so basically $\delta_K(t_n)$ is 1 for $p(C_K)$

20 ✓

$$= \frac{-1}{N} \sum_{n=1}^{N} \left[\delta_K(t_n)\, x_n - x_n\, p(C_K|x_n) \right]$$

gradient of E_D

$$\boxed{\nabla_w E_D = \frac{-1}{N} \sum_{n=1}^{N} \left(\delta_K(t_n) - p(C_K|x_n) \right) x_n} \quad \text{A}$$

Again $\nabla_w E_w(w) = \frac{\partial}{\partial w} \frac{\alpha}{2} w_K^T w_K = \frac{\alpha}{2} \cdot 2 w_K^T = \alpha w_K^T$

$$\boxed{\nabla_w E_w = \alpha w_K^T} \quad \text{B}$$

$\therefore$ Ans

$$\boxed{\nabla_w E = \nabla_w E_D + \nabla_w E_w = -\frac{1}{N} \sum_{n=1}^{N} \left(\delta_K(t_n) - p(C_K|x_n) \right) x_n + \alpha w_K^T}$$

method 2 $E_D(w) = -\frac{1}{N} \sum_n \ln\left(\frac{e^{w_{t_n}^T x_n}}{\sum_n e^{w_{t_n}^T x_n}} \right)$ (grouping n-weights)

$$E_D(w) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \delta_k(t_n) \ln\left(\frac{e^{w_k^T x_n}}{\sum_{k=1}^{K} e^{w_k^T x_n}} \right)$$ (grouping k-weights)

let, $$E_n = \sum_{k=1}^{K} \delta_k(t_n) \ln\left(\frac{e^{w_k^T x_n}}{\sum_k e^{w_k^T x_n}} \right)$$

$$E_n = \sum_k \delta_k(t_n)\, w_k^T x_n - \sum_k \delta_k(t_n) \ln\left(\sum_k e^{w_k^T x_n} \right)$$

$$\frac{\partial E_n}{\partial w_j} = \delta_j(t_n)\, x_n - \delta_j(t_n) \cdot \frac{1}{\sum_j e^{w_j^T x_n}} \cdot e^{w_j^T x_n} \cdot \delta_j(t_n)\, x_n$$

$$= \delta_j(t_n)\, x_n - \frac{e^{w_j^T x_n}}{\sum_j e^{w_j^T x_n}}\, x_n$$

$$\frac{\partial E_n}{\partial w_k} = \delta_k(t_n)\, x_n - \frac{e^{w_k^T x_n}}{\sum_k e^{w_k^T x_n}}\, x_n$$

$$\boxed{\frac{\partial E_n}{\partial w_k} = \left(\delta_k(t_n) - p(C_k|x_n) \right) x_n}$$

$$\therefore \nabla_{w_k} E_D(w) = -\frac{1}{N} \sum_n \left[\delta_k(t_n) - p(C_k|x_n) \right] x_n \quad \text{—— ⓐ}$$

also, $$\boxed{E_w(w) = \frac{\alpha}{2} w_k^T w_k}$$

$$\nabla_{w_k} E_w(w) = \frac{\partial}{\partial w_k} \frac{\alpha}{2} w_k^T w_k = \alpha w_k$$

$$\therefore \nabla E = -\frac{1}{N} \sum_{n=1}^{N} \left(\delta_k(t_n) - p(C_k|x_n) \right) x_n + \alpha w_k$$

$$\boxed{\nabla_w E = \alpha \vec{w}_k - \frac{1}{N} \sum_n \left(\delta_k(t_n) - p(C_k|x_n) \right) \vec{x}_n}$$

Ans