Aashish Adhikari
Homework Assignment 1 - CS 534
Machine Learning

Q.1 Consider 2 coins, one is fair and the other one has a probability for head as 1/10. Now you randomly pick one of the coins. Answer the following.

(2) What is the probability that you picked the fair coin? What is the probability of the first toss being head?

Solution:

Let the two coins be represented as C_1 and C_2 where C_1 is the fair coin.

In choosing the fair coin C_1 from two coins C_1 and C_2,

$$\text{probability} = \frac{\text{No. of favorable case}}{\text{Total Sample Space}}$$

$$= \frac{1}{2} \quad \therefore P(C_1) = 1/2$$

There are two possible paths by which the first toss can be a head; either choosing C_1 and then making the first toss to get a head or choosing C_2 and then making the first toss head.

Assuming event E represents the first toss being Head, the possible sample space is $\{C_1H, C_2H\}$

$\therefore P(E)$ = Getting the first head after choosing C_1 or Getting the first head after choosing C_2

$$= P(C_1) \times P(H|C_1) + P(C_2) \times P(H|C_2)$$

$$= \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{10}$$

$$= \frac{1}{4} + \frac{1}{20} \quad = \frac{6}{20} = \frac{3}{10}$$

New Link
Date
Page

(b.) If both tosses are heads, what is the probability that you have chosen the fair coin?

Solution:

Let P(f) = probability of choosing the fair coin
P(HH) = probability of both the tosses being head

∴ P(f|HH) = probability that fair coin was chosen given that both the tosses are heads

$$= P(HH|f) \times P(f) / P(HH) \quad \text{---------- eq. I}$$

Here, P(HH|f) = probability that both the tosses were head given that the fair coin was chosen

$$= \frac{1}{2} \times \frac{1}{2}$$

$$= \frac{1}{4}$$

Also, P(HH) can happen in 2 ways, either select C_1 and get both heads or select C_2 and get both heads

$$= \left(\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}\right) + \left(\frac{1}{2} \times \frac{1}{10} \times \frac{1}{10}\right)$$

$$= \frac{1}{2}\left(\frac{1}{4} + \frac{1}{100}\right)$$

$$= \frac{1}{2}\left(\frac{26}{100}\right)$$

$$= \frac{13}{100}$$

∴ from equation I,

$$P(f|HH) = \frac{1}{4} \times \frac{1}{2} \times \frac{100}{13}$$

$$= \frac{25}{26}$$, the probability that fair coin was chosen given that both the tosses are heads. #

Q.2. Given a set of i.id samples $x_1, x_2, \cdots x_n \sim \text{uniform}(0, \theta)$.

(a) Write down the likelihood function of θ.

Solution;

for a uniform distribution with interval $0 \le x_i \le \theta$, probability density function, pdf is given as

$$\text{pdf } f(x) = \begin{cases} \frac{1}{\theta - 0} = \frac{1}{\theta} & \text{for } 0 \le x_i \le \theta \\ 0 & \text{otherwise} \end{cases}$$

∴ the likelihood function for θ is given as

$$\mathcal{L}(\theta) = ~~f(x|\theta)~~ \; f(x_1, x_2, \cdots x_n | \theta)$$

$$= \prod_{i=1}^{n} \frac{1}{\theta}$$ [since iid samples, we can multiply]

$$= \frac{1}{\theta^n}$$

Hence $\mathcal{L}(\theta) = \begin{cases} 1/\theta^n & \text{for } 0 \le x_i \le \theta \text{ where } i = 1, 2, \cdots n \\ 0 & \text{otherwise} \end{cases}$

(b) Find the maximum likelihood estimator for θ.

Solutions

To make calculation easier, we take the log of likelihood first and then maximize. Here, $\log(L(\theta))$ gives the log of the likelihood ~~and~~

$$\log\left(\frac{1}{\theta^n}\right)$$

$$= \log(\theta^{-n})$$

$$= -n \log \theta$$

Taking its derivative, we get $\frac{d}{d\theta}(-n \log \theta)$

$$= \frac{-n}{\theta}$$

Since both n and θ are positive, this derivative value is always negative $\Rightarrow$ the function is decreasing in θ. Thus, to maximize $L(\theta)$, we should estimate the smallest value of θ ~~for the given~~. However, choosing a value x_n means that the smallest possible value of θ that we can use is going to be that x_n which is going to be greater than $x_{n-1}, x_{n-2}, \ldots, x_1$. Thus the maximum likelihood estimator MLE of θ is given as the maximum value among all x_i where $i = 1, 2, \ldots n$

$$\Rightarrow \theta_{MLE} = \max\{x_1, x_2, \ldots x_n\} \quad \#$$

Q3. In class, when discussing linear regression, we assume that the Gaussian noise is independently identically distributed. Now, we assume the noises $\epsilon_1, \epsilon_2, \cdots \epsilon_n$ are independent but each $\epsilon_m \sim N(0, \sigma_m^2)$, i.e., it has its own distinct variance.

(a) Write down the log likelihood function of w.

Solution:

Here, w is the weight vector of our model.

Likelihood function of w

$$= \mathcal{L}(w)$$
$$= \mathcal{L}(w; y_1, y_2, \cdots y_n | x_1, x_2, \cdots x_n)$$
$$= f(y_1, y_2, \cdots y_n | x_1, x_2, \cdots x_n; w)$$
$$= \prod_{i=1}^{n} f(y_i | x_i; w)$$

Since we see an error for each query point x_i while generating y_i, $f(y_i|x_i;w)$ is equivalent to $f(\epsilon_i|w)$ where ϵ_i is the error at this particular query point x_i.

$\therefore \mathcal{L}(w)$ becomes $\prod_{i=1}^{n} f(\epsilon_i|w)$

$$= \prod_{i=1}^{n} \frac{1}{\sigma_i \sqrt{2\pi}} e^{\left(\frac{-\epsilon_i^2}{2\sigma_i^2}\right)}$$

[plugging in the value for Normal distribution with distinct variances in errors]

Taking log-likelihood allows us to sum the values instead of multiplication.

$$\log(L(w)) \qquad \left(\frac{-\epsilon_i^2}{2\sigma_i^2}\right)$$

$$= \log \prod_{i=1}^{n} \frac{1}{\sigma_i \sqrt{2\pi}} e$$

$$= \sum_{i=1}^{n} \log\left(\frac{1}{\sigma_i \sqrt{2\pi}} \times e^{\frac{-\epsilon_i^2}{2\sigma_i^2}}\right)$$

$$= \sum_{i=1}^{n} \log \frac{1}{\sigma_i \sqrt{2\pi}} + \sum_{i=1}^{n} \log e^{\frac{-\epsilon_i^2}{2\sigma_i^2}}$$

$$= C + \sum_{i=1}^{n} \frac{-\epsilon_i^2}{2\sigma_i^2}.$$

Now, since ϵ_i is an error at query point x_i,

ϵ_i can be replaced with $y_i - w^T x_i$

$$\log(L(w)) \quad C + \sum_{i=1}^{n} \frac{-(y_i - w^T x_i)^2}{2\sigma_i^2} \qquad \text{Eq. II}$$

(b) Show that maximizing the log likelihood is equivalent to minimizing a weighted least square loss function $J(w) = \frac{1}{2} \sum_{m=1}^{n} a_m (w^T x_m - y_m)^2$, and express each ~~terms~~ ~~an~~ a_m in terms of σ_m.

Solution:

Equation II above gives us the log Likelihood for w. Maximizing log Likelihood means minimizing the value to be subtracted in equation II i.e., $\frac{(y_i - w^T x_i)^2}{2\sigma_i^2}$ which is exactly what we want to minimize in a weighted least square loss function.

Comparing eq. II with $J(w)$

$m = i$

$a_m = \frac{1}{\sigma_p^2}$ ← weight associated with each query point.

$y_m = y_i$

$x_m = x_i$

Hence, $a_m = \frac{1}{\sigma_p^2}$

(c) Derive a batch gradient descent algorithm for optimizing this objective.

Solution:

Before deriving, let's see in steps how batch gradient descent works.

- First we take a random weight vector $w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}$
- Then we calculate the cost of using this weight vector w on our training examples.
- Then we take the partial derivatives of the cost function wrt to each parameter in w to find out the direction for descent.
- We determine the step of the descent using some constant called α, the learning rate.
- Then we simultaneously update the values of every parameter w_i in w and calculate the new cost of the cost function corresponding to this new w.
- We repeat until we ~~find out~~ reach the convergence criterion.

Converting above steps into an algorithm:

Steps:

1) Start

2) Initialize a random weight vector w
   a local weight vector a
   the input vector x
   the output vector y
   error criterion, p

3) Define Error, $\epsilon := \frac{1}{2} \times \sum_m (w^T x_m - y_m)^2$

4) While $\epsilon \geq p$, do for each m simultaneously
   $w := w - \alpha * a_m * (w^T x_m - y_m) * x_m$
   $\epsilon := \frac{1}{2} \times \sum_m (w^T x_m - y_m)^2$

5) Return w

6) Stop #

(d) Derive a closed form solution to this optimization problem.

Solution:

Instead of iterating through gradient descent, we can get an analytical solution to this problem. Since our solution converges when the slope of the cost function becomes zero, we can directly equate these two to find out the optimal values for the weights.

For this,

let our input vector is $X = \begin{bmatrix} x_{10} & x_{11} & \cdots & x_{1d} \\ x_{20} & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & & \vdots \\ x_{N0} & x_{N1} & \cdots & x_{Nd} \end{bmatrix}$

Output vector is $Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$

, weight vector is $W = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix}$

Since every point in the example has 1 particular local weight, there is 1 weight associated with each i in $1 \ldots N$.

So, $A = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & \frac{1}{\sigma_N^2} \end{bmatrix}$

Calculating the gradient of the cost function, $\nabla_w J(w)$

$$= \nabla_w \frac{1}{2} \times A \times (XW - Y^2)$$

$$= \nabla_w \frac{1}{2} \times (XW - Y)^T \times A \times (XW - Y)$$

$$= \nabla_w \frac{1}{2} \times (XW - Y)^T \times (AXW - AY)$$

$$= \nabla_w \frac{1}{2} \times (W^T X^T - Y^T) \times (AXW - AY)$$

$$= \nabla_w \frac{1}{2} \times (W^T X^T AXW - W^T X^T AY - Y^T AXW + Y^T AY)$$

Trace of a real number is a real number, so;

$$= \nabla_w \, tr\left(\frac{1}{2} \times (W^T X^T AXW - W^T X^T AY - Y^T AXW + Y^T AY)\right)$$

$$= \frac{1}{2} \times \nabla_w \left(tr(W^T X^T AXW - W^T X^T AY - Y^T AXW + Y^T AY)\right)$$

Using the distributive property

$$= \frac{1}{2} \times \nabla_w \left(tr(W^T X^T AXW) - tr(W^T X^T AY) - tr(Y^T AXW) + tr(Y^T AY)\right)$$

= using $tr(X) = tr(X^T)$ we get

$$= \frac{1}{2} \times \nabla_w \left(tr(W^T X^T AXW) - tr(W^T X^T AY) - tr(W^T X^T A^T Y) + tr(Y^T AY)\right)$$

New Link

Date
Page

we get the second and the third term to be equal since A is a diagonal matrix

$$= \frac{1}{2} \times \nabla_W \left(tr(W^T X^T A X W) - 2\, tr(W^T X^T A^T Y) + tr(Y^T A X) \right)$$

$$= \frac{1}{2} \times \nabla_W \left(tr(W^T X^T A X W) - 2\, tr(W^T X^T A Y) + tr(Y^T A X) \right)$$

$$[\because A = A^T]$$

$$= \frac{1}{2} \times \left[\nabla_W \left(tr(W^T X^T A X W) - 2 \nabla_W\, tr(W^T X^T A Y) + 0 \right) \right]$$

$$= \frac{1}{2} \left[\nabla_W \left(tr(W^T X^T A X W) - 2 X^T A Y \right] \quad \because \nabla_{W^T} tr\, WX = X$$

= We have $\nabla_{P^T} tr\, PQP^T R = Q^T P^T R^T + Q P^T R$ ——— (III)

Let us assume that $W^T = P$, $X^T A X = Q$, $P^T = W$, $R = I$

Now, using equation (III), we get

$$\frac{1}{2} (X^T A^T X W I + X^T A X W I - 2 X^T A Y)$$

$$= \frac{1}{2} (X^T A X W I + X^T A X W I - 2 X^T A Y)$$

$$= X^T A X W - X^T A Y$$

We can equate this to zero to find the critical value

$$X^T A X W - X^T A Y = 0$$

or, $X^T A X W = X^T A Y$

$\Rightarrow W = (X^T A X)^{-1} X^T A Y$

which gives us the solution for our weight vector W analytically. #

Q4. Consider a binary classification task with the following loss matrix:

predicted label $\hat{y}$	true label y	
	0	1
0	0	10
1	5	0

We have to build a probabilistic model that for each example x gives us an estimated $P(y=1|x)$. It can be shown that, to minimize the expected loss for our decision, we should set a probability threshold θ and predict $\hat{y}=1$ if $P(y=1|x) > \theta$ and $\hat{y}=0$ otherwise.

a) Compute θ for the above loss matrix.

Solution:

If the loss matrix looked like below,

predicted label $\hat{y}$	true label y	
	0	1
0	0	A
1	A	0

the loss for predicting $\hat{y}=1$ when $y=0$ would equal the loss for predicting $\hat{y}=0$ when $y=1$. In such case, the probability threshold would not be of much care. However, in our question the loss for predicting $\hat{y}=0$ when true label is 1 is 10, which is greater than

the loss of predicting $\hat{y}=1$ when $y=0$.
Thus, we need to find out a threshold for probability that will act as a boundary for when to predict $\hat{y}=1$ and when to predict $\hat{y}=0$. Since the loss of predicting $\hat{y}=0$ when $y=1$ is more, by intuition, the probability that our model will predict $\hat{y}=1$ will be more.

At threshold, the losses should be equal. Let θ be the threshold probability.

∴ Loss when predicted $\hat{y}=0$ while $y=1$
$= 0 \times (1-\theta) + 10 \times \theta$
$= 10\theta$

Loss when predicted $\hat{y}=1$ while $y=0$
$= 5 \times (1-\theta) + 0 \times \theta$
$= 5 - 5\theta$

Since the losses should equal at threshold, equating them

$10\theta = 5(1-\theta)$
or, $2\theta = 1-\theta$
$\Rightarrow \theta = 1/3$

Hence, we have a model that will predict $\hat{y}=1$ for any probability greater than (or equal to) $(1/3)$.
ie., $P(y=1|x) > 1/3 \Rightarrow$ predict $\hat{y}=1$. #

b. Show a loss matrix where the threshold is 0.1

Solution:

Given $\theta = 0.1$

$\therefore$ predict $\hat{y} = 1$ if $P(y=1|x) > 1/10$

$\hat{y} = 0$ if $P(y=1|x) \leq \frac{1}{10}$

Loss Matrix

		true label y	
		0	1
predicted label $\hat{y}$	0	0	A
	1	B	0

The losses should equal at $\theta = 0.1$

$\Rightarrow 0 \times 0.1 + A \times 0.9 = B \times 0.1 + 0 \times 0.9$

or, $9A = 1B$

$\Rightarrow A : B = 1 : 9$

Hence, our loss matrix

		true label y	
		0	1
predicted label $\hat{y}$	0	0	1
	1	9	0

Q.5 Consider the maximum likelihood estimation problem for multi-class logistic regression using the soft-max function defined below:

$$p(y=k|x) = \frac{\exp(w_k^T x)}{\sum_{j=1}^{k} \exp(w_j^T x)}$$

We can write out the likelihood function as:

$$L(w) = \prod_{i=1}^{N} \prod_{k=1}^{k} p(y=k|x_i)^{I(y_i=k)}$$

where $I(y_i=k)$ is the indicator function, taking value 1 if y_i is k.

a) What are i and k in this likelihood function?

Solution:

Here i is the iterator for the total number of training data that we have.

k is the iterator for the total number of classes in the logistic regression.

b) Compute the log-likelihood function

Solution:

$$\mathcal{L}(w) = \prod_{i=1}^{N} \prod_{k=1}^{K} p(y=k|x_i)^{I(y_i=k)}$$

Take log on both sides

$$\log(\mathcal{L}(w))$$

$$= \log\left(\prod_{i=1}^{N} \prod_{k=1}^{K} p(y=k|x_i)^{I(y_i=k)} \right)$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \log\left(p(y=k|x_i)^{I(y_i=k)} \right)$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} I(y_i=k) \log\left(\frac{e^{w_k^T x_i}}{\sum_{j=1}^{k} e^{w_j^T x_i}} \right)$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} I(y_i=k) \left(\log\left(e^{w_k^T x_i} \right) - \log\left(\sum_{j=1}^{k} e^{w_j^T x_i} \right) \right)$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} I(y_i=k) \left(w_k^T x_i - \log\left(\sum_{j=1}^{k} e^{w_j^T x_i} \right) \right) \quad \text{---(IV)}$$

which is the required log-likelihood function.

c. What is the gradient of the log-likelihood function wrt the weight vector w_c of class $\mathcal{C}$.

Solution:

Let the weight ~~fa~~ vector for class c be w_c.
From question no. 5(b), we have, $\log(\lambda(w))$, so

$$\nabla w_c \log(\lambda(w))$$

$$= \nabla w_c \sum_{i=1}^{N} \sum_{k=1}^{K} I(y_i = k)(w_k^T x_i) - \nabla w_c \sum_{i=1}^{N} \sum_{k=1}^{K} I(y_i = k) \cdot \log\left(\sum_{j=1}^{K} e^{w_j^T x_i}\right)$$

$$= \sum_{i=1}^{N} I(y_i = c)\left(\nabla w_c^T (w_c^T x_i)\right)^T - \sum_{i=1}^{N} \frac{e^{(w_c^T x_i)(x_i)}}{\sum_{j=1}^{K} e^{w_j^T x_i}}$$

$$= \sum_{i=1}^{N} I(y_i = c)(x_i) - \sum_{i=1}^{N} \frac{e^{(w_c^T x_i)(x_i)^T}}{\sum_{j=1}^{K} e^{(w_j^T x_i)}}$$

$$= \sum_{i=1}^{N} (x_i)\left(I(y_i = c) - \frac{e^{w_c^T x_i}}{\sum_{j=1}^{K} e^{w_j^K x_i}}\right) //$$