CSE 4/574
Introduction to Machine Learning
Homework 3
Total Marks 50

Your Name: Karan Bali
Your email ID: kbali@buffalo.edu
Your UB Person ID: 50381691

---

1. [6] Show that for a linearly separable data set, the maximum likelihood solution for the logistic regression model is obtained by finding a vector w whose decision boundary $w^T \varphi(x) = 0$ separates the classes and then taking the magnitude of w to infinity.

Ans.1.

As we know that for a linearly separable data into Class 'A' & 'B', we should have

$$\mathbf{w}^T \phi(\mathbf{x}) > 0 For Class' A'$$

$$\mathbf{w}^T \phi(\mathbf{x}) < 0 For Class' B'$$

(or vice versa)

Thus to get maximum likelihood solution for the logistic regression model, we have to find the solution where:

$\mathbf{w}^T \phi(\mathbf{x}) \to \infty or -\infty$ For class A & B respectively

Putting these values in equations of logistic regression gives us the following:

$$p\left(A \mid \phi\left(\mathbf{x}\right)\right) = \sigma\left(\mathbf{w}^T \phi\left(\mathbf{x}\right)\right) \rightarrow 1$$

$$p\left(A \mid \phi\left(\mathbf{x}\right)\right) = 1 - p\left(B \mid \phi\left(\mathbf{x}\right)\right) = 1 - \sigma\left(\mathbf{w}^T \phi\left(\mathbf{x}\right)\right) \rightarrow 1$$

So while learning the model, the model tries to increase 'w' to infinity to make wT * φ(x) goes to infinity.

Hence, the maximum likelihood solution for the logistic regression model is obtained by finding a vector w whose decision boundary wTφ(x) = 0 separates the classes and then taking the magnitude of w to infinity.

---

2. [8] Define Logistic Sigmoid function $\sigma(.)$. Show that the logistic sigmoid function σ satisfies the property $\sigma(-a) = 1-\sigma(a)$ and that its inverse is given by $\sigma^{-1}(y) = ln(y/(1-y))$.

Ans.2.

From 4.59 on Page number 197 of Bishop's book, we have:

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

Thus,

$$\sigma(-a) = \frac{1}{1 + \exp(a)}$$

Thua adding both give us:

= $\sigma(a) + +\sigma(-a) = \frac{1}{1 + \exp(-a)} + \frac{1}{1 + \exp(a)}$ = 1

Thus, $\frac{1}{1 + \exp(a)} = 1 - \frac{1}{1 + \exp(-a)}$

= $\sigma(-a) = 1 - \sigma(a)$

Also, To find inverse of sigmoid we have to find a function that gives back input 'a':

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

$$\Rightarrow \quad \frac{1}{y} - 1 = e^{-a}$$

$$\Rightarrow \quad \ln\left\{\frac{1-y}{y}\right\} = -a$$

$$\Rightarrow \quad \ln\left\{\frac{y}{1-y}\right\} = a = \sigma^{-1}(y)$$

---

3. [10] Given a set of data points $\{x_n\}$, we can define the convex hull to be the set of all points x given by $x = \sum_n \alpha_n x_n$. where $\alpha_n \geq 0$ and $\sum_n \alpha_n = 1$. Consider a second set of points y_n together with their corresponding convex hull. By definition, the two sets of points will be linearly separable if there exists a vector w and a scalar w_0 such that $w^T x_n + w_0 > 0$ for all x_n and $w^T y_n + w_0 < 0$ for all y_n. Show that if their convex hulls intersect, the two sets of points cannot be linearly separable, and conversely that if they are linearly separable, their convex hulls do not intersect.

Ans.3.

First infuse the given form of 'x' convex hull points in the equation of discriminant:

$$\widehat{\mathbf{w}}^T \mathbf{X_O} + w_0$$
$$= \widehat{\mathbf{w}}^T \left(\sum_n \alpha_n \mathbf{x_n} \right) + w_0$$

$$= \left(\sum_n \alpha_n \widehat{\mathbf{w}}^T \mathbf{x_n} \right) + \left(\sum_n a_n \right) w_0$$

We can convert the above equation into this (considering bias w0 as constant under the manipulation):

$$= \sum_n \alpha_n \left(\widehat{\mathbf{w}}^T \mathbf{x_n} + w_0\right) \quad (Equation1)$$

Also, we know that $\alpha_n \geq 0$ and Sum(α_n) =1

Now we can similarly consider 'y' set of points and repeating same exercise will give us

$$= \sum_n \beta_n \left(\widehat{\mathbf{w}}^T \mathbf{y_n} + w_0\right) \quad (Equation2)$$

Now for to be linearly separable, Equation 1 > 0 & Equation 2 < 0 (or vice versa)

But to intersect we can calculate Equation 1 = Equation 2,

But this is not possible as both equations are of different signs and thus, they can't be (>0 and < 0) AND (=0) at the same time.

Hence,
if their convex hulls intersect, the two sets of points cannot be linearly separable, and conversely that if they are linearly separable, their convex hulls do not intersect.

---

4. [8+6] Define Least Square Classifier and discuss its drawbacks. How does Fisher Linear Discriminant aim to address these shortcomings.

Ans.4.

Consider a general classification problem with K classes, with a 1-of-K binary coding scheme for the target vector t. One justification for using least squares in such a context is that it approximates the conditional expectation E[t|x] of the target values given the input vector. For the binary coding scheme, this conditional expectation is given by the vector of posterior class probabilities. Unfortunately, however, these probabilities are typically approximated rather poorly, indeed the approximations can have values

outside the range (0, 1), due to the limited flexibility of a linear model as we shall see shortly.

Each class Ck is described by its own linear model so that:

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

where k = 1, . . . , K. We can conveniently group these together using vector notation so that:

$$Y(\mathbf{x}) = \widehat{\mathbf{W}}^{\mathrm{T}}\mathbf{x}$$

where ,W_hat is a matrix whose kth column comprises the D + 1-dimensional vector w_hat_k = (wk0,w_k^T)^T and x is the corresponding augmented input vector (1, x^T)^T with a dummy input x0 = 1. We now determine the parameter matrix ,W by minimizing a sum-of-squares error function. Consider a training data set {xn, tn} where n = 1, . . . , N, and define a matrix T whose nth row is the vector t_n^T , together with a matrix X whose nth row is x_hat_n^T. The sum-of-squares error function can then be written as:

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\,\mathrm{Tr}\left\{(\tilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\tilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}$$

Setting the derivative with respect to,W_hat to zero, and rearranging, we then obtain the solution for ,W_hat in the form:

$$\widetilde{\mathbf{W}} = \left(\widetilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}}\right)^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T} = \widetilde{\mathbf{X}}^{\dagger}\mathbf{T}$$

where X† is the pseudo-inverse of the matrix X. We then obtain the discriminant function in the form:

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}} = \mathbf{T}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}^{\dagger}\right)^{\mathrm{T}}\tilde{\mathbf{x}}$$

However, problems with least squares can be more severe than simply lack of Robustness. In cases where a data set is linearly separable, but highly variant with classes having clusters of data points deep inside. This causes aberration in the least square method, as it tries to take into account the huge distance of deep seated points that skews the calculation. There's an example in the book in Pg. 186 that defines the problem with images.

The failure of least squares should not surprise us when we recall that it corresponds to maximum likelihood under the assumption of a Gaussian conditional distribution, whereas highly variant data have a distribution that is far from Gaussian. On the other hand, Fisher algorithm tries to solve this problem by projecting the data points to a lesser dimension & then tries to maximize the ratio “between-class variance” to “within-class variance” with the goal of reducing data variation in the same class and increasing the separation between classes.

Where ratio is given by following formula,

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$$

Thus, fisher calculation is not affected by the deep seated variant data as in the case of least squares. The least-squares approach to the determination of a linear discriminant was based on the goal of making the model predictions as close as possible to a set of target values. By contrast, the Fisher criterion was derived by requiring maximum class separation in the output space. AS the projection is on a different dimension, which considers the mean & variance of the data set, the problem of least squares is hugely reduced.

5.[6] Describe the Probabilistic Generative Model and highlight its difference from the Probabilistic Discriminative Model.

Ans.5.

The main difference between a Generative model and discriminative model is whether you use your data from the perspective of probability distribution (Generative) OR Quantitative perspective (perspective). We then try to minimize(or maximize) the optimization problem obtained using different perspectives of data.

For the two-class classification problem, we have seen that the posterior probability of class C1 can be written as a logistic sigmoid acting on a linear function of x, for a wide choice of class-conditional distributions p(x|Ck). Similarly, for the multiclass case, the posterior probability of class Ck is given by a softmax transformation of a linear function of x. For specific choices of the class-conditional densities p(x|Ck), we have used maximum likelihood to determine the parameters of the densities as well as the class priors p(Ck) and then used Bayes’ theorem to find the posterior class probabilities.

However, an alternative approach is to use the functional form of the generalized linear

model explicitly and to determine its parameters directly by using maximum likelihood. Least squares is one of the common discriminative algorithms. The indirect approach to finding the parameters of a generalized linear model, by fitting class-conditional densities and class priors separately and then applying Bayes' theorem, represents an example of generative modelling, because we could take such a model and generate synthetic data by drawing values of x from the marginal distribution p(x). In the direct approach, we are maximizing a likelihood function defined through the conditional distribution p(Ck|x), which represents a form of discriminative training. One advantage of the discriminative approach is that there will typically be fewer adaptive parameters to be determined, as we shall see shortly. It may also lead to improved predictive performance, particularly when the class-conditional density assumptions give a poor approximation to the true distributions.

In the Generative model we can synthesize the data, so it's commonly used in scenarios where data is scarce like related to rare events, rare disease, etc. We try to generate the data using prior knowledge. This prior also helps us to infuse our human knowledge into the model. However, this infused knowledge might be right or wrong. On other hand, discriminative learning uses less parameters & many times gets accurate results. It's a good choice if you have plenty of labeled data. Anyways, both have their respective advantages & disadvantages.

---

6. [6] Define Logistic Regression for Binary and multi-class classification tasks.

Ans.6.

the posterior probability of class C1, C2 can be written as a logistic sigmoid acting on a linear function of the feature vector φ so that:

$$p\left(C_2 \mid \phi\left(\mathbf{x_m}\right)\right) = 1 - p\left(C_1 \mid \phi\left(\mathbf{x_m}\right)\right) = 1 - \sigma\left(\mathbf{w}^T \phi\left(\mathbf{x_m}\right)\right)$$

Here σ(.) is the logistic sigmoid function. In the terminology of statistics, this model is known as logistic regression. We now use maximum likelihood to determine the parameters of the logistic
regression model. To do this, we shall make use of the derivative of the logistic sigmoid function, which can conveniently be expressed in terms of the sigmoid function itself.

$$\frac{d\sigma}{da} = \sigma(1-\sigma)$$

For a data set {φn, tn}, where tn ∈ {0, 1} and φn = φ(xn), with n =1, . . . , N, the likelihood function can be written:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} \{1 - y_n\}^{1-t_n}$$

where t = (t1, . . . , tN)^T and y_n = p(C1|φn). As usual, we can define an error function by taking the negative logarithm of the likelihood, which gives the cross entropy error function in the form:

$$E(\mathbf{w}) = -\ln p(\mathbf{t} \mid \mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln (1 - y_n)\}$$

where yn = σ(a_n) and a_n = w^T*φ_n. Taking the gradient of the error function with respect to 'w', we obtain:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n) \phi_n$$

If desired, we could make use of the above result to give a sequential algorithm in which patterns are presented one at a time, in which each of the weight vectors is updated using gradient descent algorithm.

Similarly for Multiclass Logistic regression, the posterior probabilities are given by a softmax transformation of linear functions of the feature variables:

$$p\left(\mathcal{C}_k \mid \phi\right) = y_k(\phi) = \frac{\exp\left(a_k\right)}{\sum_j \exp\left(a_j\right)}$$

where the 'activations' a_k are given by: a_k = w_k^T * φ

There we used maximum likelihood to determine separately the class-conditional densities and the class priors and then found the corresponding posterior probabilities using Bayes' theorem, thereby implicitly determining the parameters {wk}. Here we consider the use of maximum likelihood to determine the parameters {wk} of this

model directly. To do this, we will require the derivatives of yk with respect to all of the activations aj:

$$\frac{\partial y_k}{\partial a_j} = y_k \left(I_{kj} - y_j \right)$$

where I_kj are the elements of the identity matrix. Next we write down the likelihood function. This is most easily done using the 1-of-K coding scheme in which the target vector t_n for a feature vector φn belonging to class C_k is a binary vector with all elements zero except for element k, which equals one. The likelihood function is then given by:

$$p\left(\mathbf{T} \mid \mathbf{w}_1, \ldots, \mathbf{w}_K\right) = \prod_{n=1}^{N} \prod_{k=1}^{K} p\left(\mathcal{C}_k \mid \phi_n\right)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

where y_nk = y_k*(φn), and T is an N X K matrix of target variables with elements tnk. Taking the negative logarithm then gives:

$$E\left(\mathbf{w}_1, \ldots, \mathbf{w}_K\right) = -\ln p\left(\mathbf{T} \mid \mathbf{w}_1, \ldots, \mathbf{w}_K\right) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk}$$

This is known as the cross-entropy error function for the multiclass classification problem.

We now take the gradient of the error function with respect to one of the parameter vectors w_j . Making use of the result calculated in previous steps for the derivatives of the softmax function, we obtain:

$$\nabla_{\mathbf{w}_j} E\left(\mathbf{w}_1, \ldots, \mathbf{w}_K\right) = \sum_{n=1}^{N} \left(y_{nj} - t_{nj}\right) \phi_n$$

Again, we could use this to formulate a sequential algorithm in which patterns are presented one at a time, in which each of the weight vectors is updated using gradient descent algorithm.